

Optimizing Revenue Prediction through E-commerce Analytics

Arizona State University

Hrishikesh Magdum (ASU ID: 1229876520)

Isha Kaushik (ASU ID: 1233665929)

Ishan Mandlik (ASU ID:1233554844)

Shristi Pathak (ASU ID:1234075767)

Tempe, AZ 85281

Abstract

Dynamic pricing strategies play a crucial role in optimizing revenue for e-commerce platforms. This project leverages three months of historical data, including product attributes and user behavior, to forecast revenue for the next month. We aim to enhance revenue prediction accuracy using advanced machine learning models like XGBoost, LightGBM, and CatBoost, along with robust feature engineering and preprocessing. Key results highlight the superiority of XGBoost with minimal RMSE and high R^2 scores, indicating strong predictive performance. This work provides valuable insights into price optimization and user interaction trends, contributing to the efficiency of dynamic pricing systems. Analysis revealed that revenue increases linearly in the mid-price range (20–200), non-generic products drive higher revenue, and advertising significantly impacts median revenue. Weekly cyclic patterns in user behavior also emerged, informing optimal scheduling for promotions.

This work provides actionable insights into price optimization and user interaction trends, contributing to the efficiency of dynamic pricing systems and offering a data-driven framework for enhancing revenue prediction in e-commerce. Future work could incorporate real-time data and explore advanced techniques like reinforcement learning to refine dynamic pricing further.

I. INTRODUCTION

Data Mining is a multidisciplinary approach that combines statistical analysis, database management, optimization, and control theory to extract meaningful insights from data, enabling informed decision-making. It also involves predicting future patterns and trends within the data. The Data Mining process follows a systematic framework, comprising stages such as data collection, exploration, preparation, modeling, and evaluation. This report explores each of these phases in-depth, with a particular emphasis on predicting revenue in e-commerce using dynamic pricing strategies.

A. Background

Revenue forecasting is a critical aspect of business operations in e-commerce, directly influencing decision-making around pricing, inventory management, and marketing strategies. Traditional methods often fail to capture the complex, dynamic interactions between user behavior, product attributes, and pricing. The emergence of machine learning enables businesses to go beyond conventional analytics, offering a deeper understanding of revenue drivers. This project focuses on analyzing three months of historical data, including product-level attributes, prices, and user interactions such as clicks, basket additions, and purchases, to forecast revenue for the next month. In doing so, it addresses key challenges like price optimization, market alignment, and understanding consumer behavior.

B. Existing Literature

The field of dynamic pricing and revenue optimization has been extensively studied, with research emphasizing the use of machine learning for forecasting and decision-making. Studies highlight the effectiveness of ensemble methods such as XGBoost and LightGBM for predictive tasks, while others focus on the role of feature engineering in enhancing model accuracy. Existing works on time-series forecasting and regression models provide foundational insights for this project.

C. System Overview

This project focuses on predicting revenue per user action by utilizing supervised regression techniques in conjunction with time series forecasting. The solution integrates structured historical shop data and employs a modular system architecture for data preprocessing, feature engineering, model development, and evaluation. It leverages various Python libraries, including NumPy, pandas, scikit-learn, and advanced modeling libraries such as XGBoost, LightGBM, CatBoost, and TensorFlow, to achieve optimal performance. The datasets are linked through the "pid" attribute, which identifies individual products. The historical data is used to understand user actions and derive predictive features for revenue forecasting during the classification period.

Models Used

To address the prediction task, the project employs a combination of traditional machine learning, ensemble techniques, deep learning, and time series forecasting models:

- **Machine Learning Models:** DecisionTree, RandomForest, AdaBoost.
- **Ensemble Methods:** XGBoost, LightGBM, CatBoost.
- **Deep Learning:** A custom Deep Neural Network.
- **Time Series Models:** ARIMA, SARIMA, Prophet, and LSTM.

These models were chosen for their ability to handle structured and sequential data effectively. Time series models specifically enabled capturing seasonality and temporal trends, which are critical for accurate revenue prediction.

Techniques Used

1. **Feature Engineering:** The project creates lagged features and aggregates user actions over time to enhance the temporal aspect of the data, facilitating the application of time series models alongside regression.
2. **Hyperparameter Tuning:** All models are fine-tuned using **Optuna**, a state-of-the-art hyperparameter optimization library. This ensures that the models are trained with the best parameters, maximizing predictive accuracy.
3. **Time Series Forecasting:** Time series techniques, such as ARIMA, SARIMA, and LSTM, are integrated to account for seasonality and trends, enriching the regression analysis with temporal insights. This dual approach of regression and time series analysis ensures a comprehensive understanding of revenue patterns.
4. **Automated Machine Learning (AutoML):** To benchmark the custom models, AutoML tools were used. These state-of-the-art libraries automatically build, optimize, and evaluate multiple machine learning pipelines, providing a robust comparison to traditional and advanced models.

By combining regression techniques with time series forecasting, the system ensures a holistic approach to revenue prediction. The incorporation of AutoML enhances the reliability of results and highlights the potential of automated frameworks in achieving state-of-the-art performance.

D. Machine Learning System Components

The development of this project is structured according to the **CRISP-DM (Cross-Industry Standard Process for Data Mining)** framework, which provides a well-organized and iterative approach to building data-driven systems. Each phase in this process has been meticulously followed to ensure the accuracy and reliability of the revenue prediction model.

Business Understanding

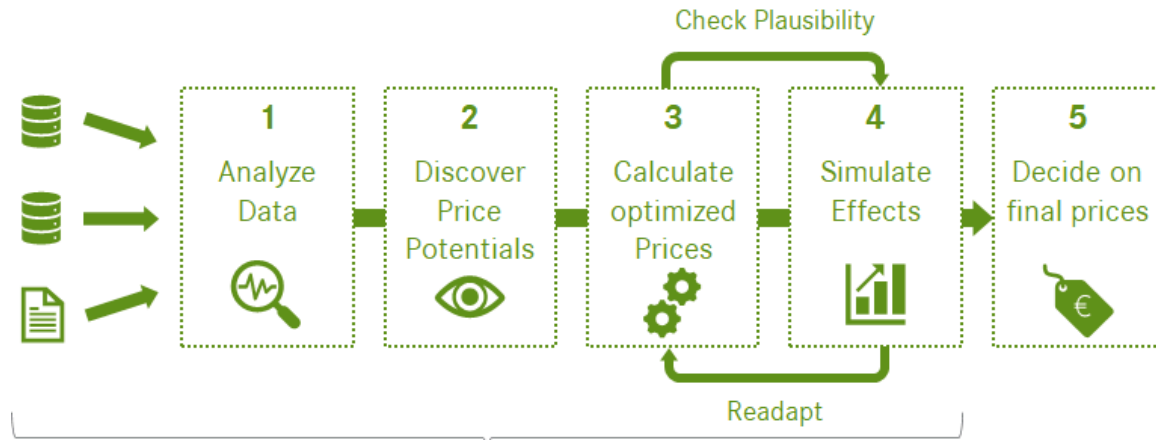
The primary objective of this system is to predict the revenue generated per user action, such as clicks, basket additions, and orders, during the classification period. This goal is driven by the need to optimize business processes, including pricing strategies, marketing campaigns, and inventory management. The task involves analyzing a large dataset with complex relationships between user actions, product attributes, and temporal variations. One key challenge lies in capturing the temporal trends effectively through time series forecasting, while another is integrating diverse features, including categorical, ordinal, and numerical variables, to provide actionable insights. This comprehensive system aims to empower decision-makers with granular revenue predictions that can guide resource allocation and operational strategies.

Data Understanding

The system uses three main datasets: `items.csv`, which contains static product attributes; `train.csv`, which records user actions and dynamic product information during the learning period; and `class.csv`, which provides data for the classification period. Exploratory data analysis revealed several important characteristics of the data. Numerical fields, such as `Price`, `Revenue`, and `CompetitorPrice`, were analyzed for distribution and trends, while categorical columns, such as `Category`, `AdFlag`, and `Manufacturer`, were grouped based on their cardinality. This analysis revealed the presence of missing values and the need for strategies to address inconsistencies in data formatting. For instance, median imputation was chosen for numerical columns, while mode imputation was used for categorical columns to ensure data completeness. These insights provided a foundation for feature engineering and model development.

Machine Learning System Components Using the CRISP-DM Cycle

The development of this project is structured according to the **CRISP-DM (Cross-Industry Standard Process for Data Mining)** framework, which provides a well-organized and iterative approach to building data-driven systems. Each phase in this process has been meticulously followed to ensure the accuracy and reliability of the revenue prediction model.



General Data Mining Task Flow

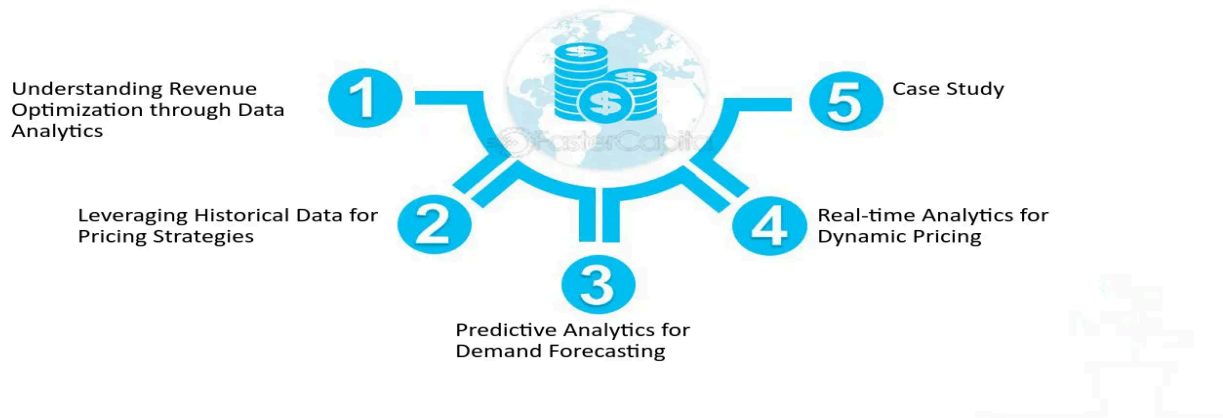
Business Understanding

The primary objective of this system is to predict the revenue generated per user action, such as clicks, basket additions, and orders, during the classification period. This goal is driven by the need to optimize business processes, including pricing strategies, marketing campaigns, and inventory management. The task involves analyzing a large dataset with complex relationships between user actions, product attributes, and temporal variations. One key challenge lies in capturing the temporal trends effectively through time series forecasting, while another is integrating diverse features, including categorical, ordinal, and numerical variables, to provide actionable insights. This comprehensive system aims to empower decision-makers with granular revenue predictions that can guide resource allocation and operational strategies.

Data Understanding

The system uses three main datasets: `items.csv`, which contains static product attributes; `train.csv`, which records user actions and dynamic product information during the learning period; and `class.csv`, which provides data for the classification period. These datasets are structured as ASCII-encoded text files, with fields separated by a "|" symbol. Exploratory data analysis revealed several important characteristics of the data. Numerical fields, such as `Price`, `Revenue`, and `CompetitorPrice`, were analyzed for distribution and trends, while categorical columns, such as `Category`, `AdFlag`, and `Manufacturer`, were grouped based on their cardinality. This analysis revealed the presence of missing values and the need for strategies to address inconsistencies in data formatting. For instance, median imputation was chosen for numerical columns, while mode imputation was used for categorical columns to ensure data completeness. These insights provided a foundation for feature engineering and model development.

The Role of Data Analytics in Revenue Optimization



Data Preparation

The data preparation phase involved multiple steps to clean, transform, and enhance the dataset for effective modeling. Columns with significant missing values, such as `campaignIndex`, and insignificant columns, like `lineID`, were dropped to reduce noise. Numerical conversions were applied to columns like `category` and `content`, where irregular formats were standardized using custom functions. For instance, the `content` column was converted into numerical values by extracting and multiplying numbers embedded in strings.

Feature engineering played a critical role in enriching the dataset. Aggregated action counts, such as `total_clicks`, `total_baskets`, and `total_orders`, were calculated for each product to capture user interaction patterns. Additionally, derived metrics like `click_to_basket_ratio` and `basket_to_order_ratio` were introduced to quantify user behavior. Competitor price features, such as `price_competitiveness` and `competitor_undercut_flag`, were also created to capture the impact of external market dynamics on revenue. Temporal features, such as the day of the week and lagged variables, were integrated to enable time series forecasting. These transformations ensured that the dataset was comprehensive and well-suited for both regression and time series analysis.

The categorical variables were processed based on their unique value count. High-cardinality columns were grouped into quartiles using frequency-based binning to reduce complexity, while low-cardinality columns were one-hot encoded. Finally, numerical features were normalized to standardize their scales, ensuring consistency during model training.

Modeling

In the modeling phase, multiple algorithms were employed to tackle the revenue prediction task. Ensemble learning methods, such as DecisionTree, RandomForest, AdaBoost, XGBoost, LightGBM, and CatBoost, were implemented to exploit their ability to handle complex relationships in the data. Deep learning models, including Deep Neural Networks (DNNs), were also explored to capture non-linear patterns. Time series models, such as ARIMA, SARIMA, Prophet, and Long Short-Term Memory (LSTM) networks, were used to forecast temporal trends and seasonality. This combination of regression and time series approaches ensured that both granular and temporal aspects of the data were captured effectively.

A unified pipeline was designed to streamline the preprocessing, training, and evaluation stages. Hyperparameter tuning was performed using Optuna, an advanced optimization framework, to systematically identify the best configurations for each model. Additionally, state-of-the-art AutoML tools were employed to benchmark custom models and validate their performance. To ensure robustness, time-based cross-validation was implemented, particularly for time series models, to evaluate their ability to generalize across unseen periods.

Evaluation

The performance of the models was evaluated using a range of metrics tailored to regression and time series tasks. For regression models, metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared were used to measure accuracy. For time series forecasting, Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) were employed to assess the models' ability to capture trends and seasonality. Insights derived from these evaluations highlighted key drivers of revenue, such as product availability, pricing strategies, and competitor dynamics. For example, the analysis showed that revenue was significantly influenced by competitive pricing, as captured by features like `price_competitiveness` and `competitor_undercut_flag`. Furthermore, the derived target variables, such as `revenue_per_click` and `revenue_per_basket`, provided a detailed understanding of user behavior and its impact on revenue generation.

Deployment

The final deployment phase involved integrating the models into a production-ready pipeline capable of handling unseen data from the classification period. The pipeline seamlessly combined predictions from regression and time series models to deliver comprehensive revenue forecasts. Visualizations were created using libraries such as Matplotlib and Seaborn to communicate the results to stakeholders effectively. The modular architecture of the system allows for scalability and flexibility, enabling the incorporation of additional data sources and evolving business needs. The deployment module has not been implemented yet and is planned for future development, making it out of the project's scope.

E. Experimental Results

This section details the results of the experiments conducted using various regression algorithms. The models were evaluated on training and validation datasets, and their performance was measured using key metrics, including RMSE (Root Mean Squared Error) and R^2 score. Hyperparameter tuning was applied to optimize model performance using Optuna. Below, we present a comprehensive summary of the results.

Model Performance Overview

The table below summarizes the performance metrics of all the models tested during the experiment:

Model	Train RMSE	Train R^2 Score	Validation RMSE	Validation R^2 Score
CatBoost Regressor	0.3253	0.999	0.8985	0.9923
XGBoost Regressor	0.3803	0.9986	0.7875	0.9941
LightGBM Regressor	0.9473	0.9914	1.056	0.9893
Random Forest Regressor	1.005	0.9904	1.3555	0.9824
Gradient Boosting Regressor	2.2407	0.9521	2.428	0.9438
Decision Tree Regressor	1.0661e-13	1.0	1.3987	0.9813
AdaBoost Regressor	13.9834	-0.8653	13.9457	-0.8542
Deep Neural Network (DNN)	7.1462	0.515	7.0908	0.515
Ensemble (Average)	1.552	0.977	1.6673	0.9734

Analysis of Results

The **CatBoost Regressor** emerged as the best individual model, achieving a low validation RMSE of 0.8985 and a high validation R^2 score of 0.9923. It effectively balanced training and validation performance, demonstrating strong generalization. Similarly, the **XGBoost Regressor** showed exceptional performance, with a slightly lower validation RMSE of 0.7875 and an R^2 score of 0.9941, making it a top contender.

The **LightGBM Regressor** also performed well, though its validation RMSE (1.0560) was slightly higher than those of CatBoost and XGBoost. It maintained strong predictive capabilities while being computationally efficient. On the other hand, the **Random Forest Regressor** exhibited signs of overfitting, with a significantly higher validation RMSE of 1.3555 compared to its training RMSE of 1.0050.

Simpler models such as the **Decision Tree Regressor** and **Gradient Boosting Regressor** underperformed relative to the more advanced algorithms. The **AdaBoost Regressor** struggled to adapt to the complexity of the data, delivering negative R^2 scores and extremely high RMSE values, making it unsuitable for this problem.

The **Deep Neural Network (DNN)**, while a sophisticated approach, failed to achieve competitive results in this experiment. Its high validation RMSE (7.0908) and R^2 score (0.5150) indicate that it was unable to capture the relationships in the data effectively, likely due to insufficient tuning or dataset-specific constraints.

Ensemble Learning

To further improve predictive performance, an **ensemble model** was created by averaging the predictions of the top-performing models: CatBoost, XGBoost, LightGBM, Random Forest, and DNN. The ensemble achieved a validation RMSE of 1.6673 and a validation R² score of 0.9734, providing a robust and balanced prediction model. While the ensemble did not outperform the best individual models, it demonstrated improved generalization by leveraging the strengths of multiple algorithms.

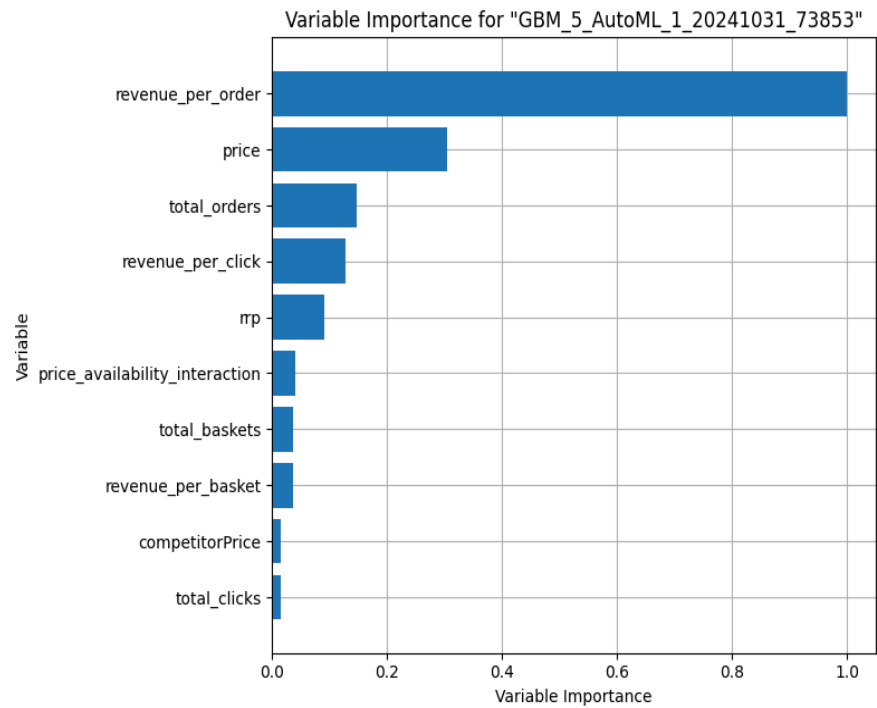
AutoML Model Overview: Gradient Boosting Machine (GBM)

The Gradient Boosting Machine (GBM) model, trained using H2O AutoML, demonstrated exceptional performance across training, validation, and testing datasets. Below is a summary of the key metrics:

Performance Metrics Summary

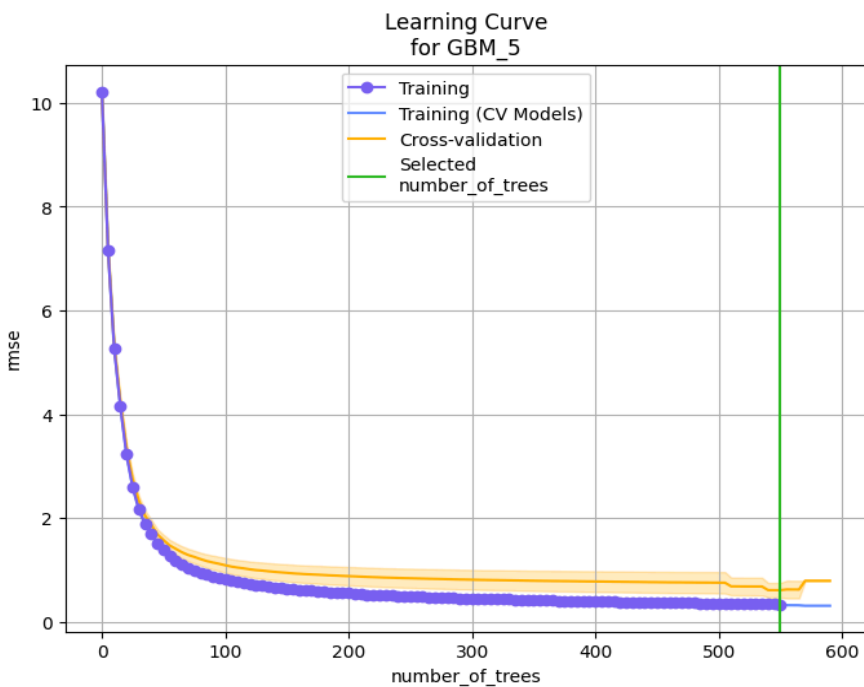
Metric	Train	Cross-Validation	Test
MSE	0.11455037002000373	0.6076096130893749	0.11455042559130092
RMSE	0.3384529066502513	0.7794931770640298	0.3384529887462968
MAE	0.1260338363621259	0.1334317926334859	0.1260339270265047
RMSLE	nan	nan	nan
Mean Residual Deviance	0.11455037002000373	0.6076096130893749	0.11455042559130092

Feature Importance



Learning Curve Plot

Learning curve plot shows the loss function/metric dependent on number of iterations or trees for tree-based algorithms. This plot can be useful for determining whether the model overfits.



Training Dynamics

During training, the model reached convergence with a final RMSE of **0.3385** on the training set after optimizing **549 trees** with a mean depth of 6.

This GBM model effectively captures relationships in the data and exhibits excellent predictive performance, making it highly suitable for applications such as revenue prediction and feature impact analysis.

II. IMPORTANT DEFINITIONS

A. Data

The data used in this project comes from a mail-order pharmacy, focusing on dynamic pricing strategies to optimize revenue prediction. There are three primary datasets provided:

1. **items.csv**: Contains static information about the products, such as attributes, product IDs (identified by pid), manufacturer details, and product categories.
2. **train.csv**: Includes three months of historical data, providing dynamic details such as daily pricing, competitor pricing, and user interactions (clicks, basket additions, and purchases).
3. **test.csv**: Contains similar attributes to the training dataset but is reserved for evaluating model predictions for the next month.

Key identifiers in the dataset include:

- **pid**: Unique identifier for each product.
- **manufacturer**: Indicates the product's manufacturer.
- **group, pharmForm, category, and campaignIndex**: Describe additional product characteristics.
- **price and competitor_price**: Reflect daily pricing data and competitor comparisons.
- **clicks, baskets, and orders**: Represent user interactions with the products.

B. Problem Statement

The goal is to predict the revenue per user action (click, basket addition, or order) for the classification period, based on historical data from the preceding three months. Starting with a baseline accuracy derived from historical averages, the problem is framed as a supervised regression task. The objective is to minimize error metrics such as RMSE and accurately forecast revenue trends, enabling optimized dynamic pricing strategies.

III. OVERVIEW OF THE PROPOSED SYSTEM

In this section, we explore and analyze the training and test datasets, address data quality issues, perform exploratory data analysis (EDA), and tackle challenges to enhance the accuracy of revenue predictions. The goal is to optimize performance metrics like RMSE for the machine learning models.

A. Data Quality Issues

The training dataset had several data quality issues, which were addressed as follows:

1. **Missing Values:**
The dataset included missing values in attributes such as `CompetitorPrice` and `campaignIndex`. These were imputed using the mean for continuous variables and the mode for categorical ones. Fig. 1 shows the list of attributes with missing values
2. **Outliers:**
Attributes such as `price` and `competitor_price` contained extreme values, which could skew model performance. These were capped at the 1st and 99th percentiles.
3. **Skewed Distributions:**
Attributes like `revenue` and user actions (`clicks`, `baskets`, and `orders`) exhibited skewed distributions. Log transformations were applied to normalize these features for improved model performance.

B. Encoding & Feature Scaling

To prepare the dataset for machine learning algorithms, categorical variables were encoded, and all features were scaled.

1. **Encoding:**
 - **Label Encoding:** Attributes like `manufacturer` and `category` were label-encoded to assign unique integers to each category.
 - **One-hot Encoding:** Features with a small number of unique categories, such as `pharmForm`, were one-hot encoded to create binary columns for each category.
2. **Feature Scaling:**
All numeric features were normalized to a scale of 0 to 1 to ensure uniformity across magnitudes. This step was critical for gradient-based models like XGBoost and LightGBM to converge faster.

C. Exploratory Data Analysis

Comprehensive EDA was conducted to uncover patterns, trends, and relationships in the data:

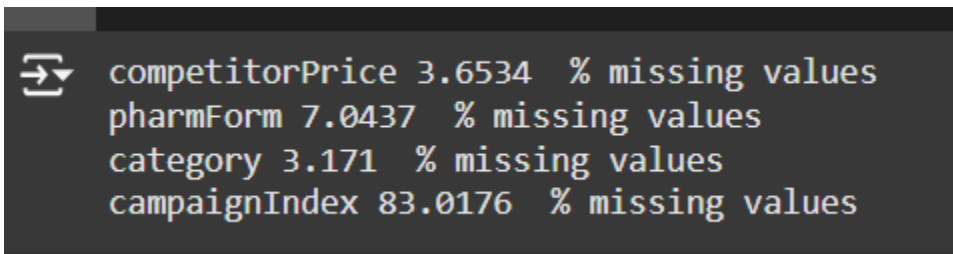
1. **Visualizing Distributions:**
 - **Revenue Distribution:** Histograms revealed a left-skewed pattern for revenue per user action, with the majority of revenue falling below 10 units as shown in Fig. 4
 - **User Behavior:** Analysis showed cyclic trends in user actions, such as peaks in clicks and orders on Mondays and weekends.

2. Correlation Analysis:

Heatmaps were generated to explore relationships between numerical attributes. For instance, `competitor_price` showed a positive correlation with revenue, while `price` had a non-linear relationship.

3. Key Insights:

- Fig. 2 illustrates the correlation between `price`, `competitor_price`, and revenue.
- Fig. 3 highlights the conversion rate from clicks to baskets and from baskets to orders, showcasing user behavior trends.



```
⇒ competitorPrice 3.6534 % missing values
   pharmForm 7.0437 % missing values
   category 3.171 % missing values
   campaignIndex 83.0176 % missing values
```

Fig. 1

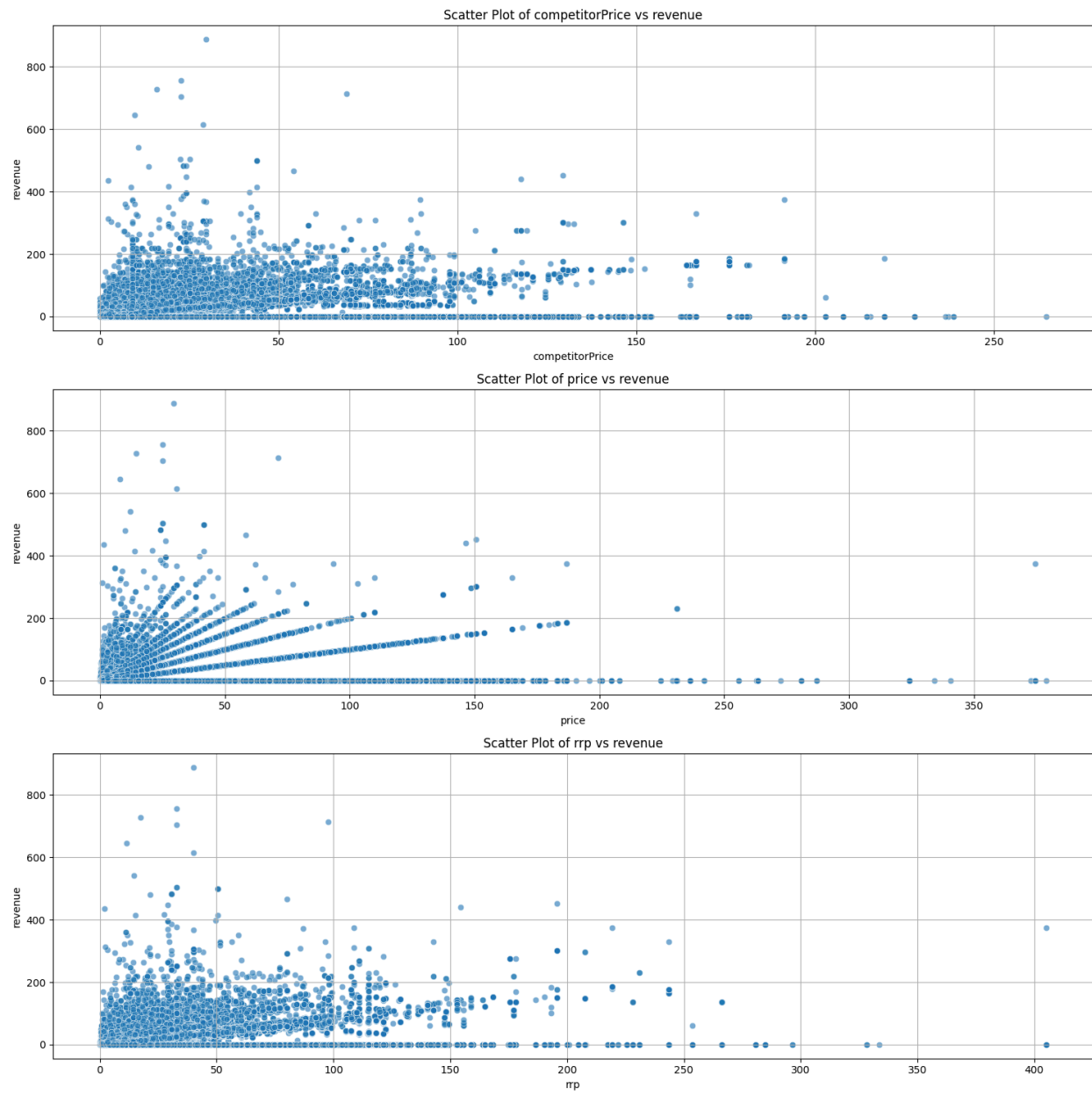


Fig. 2

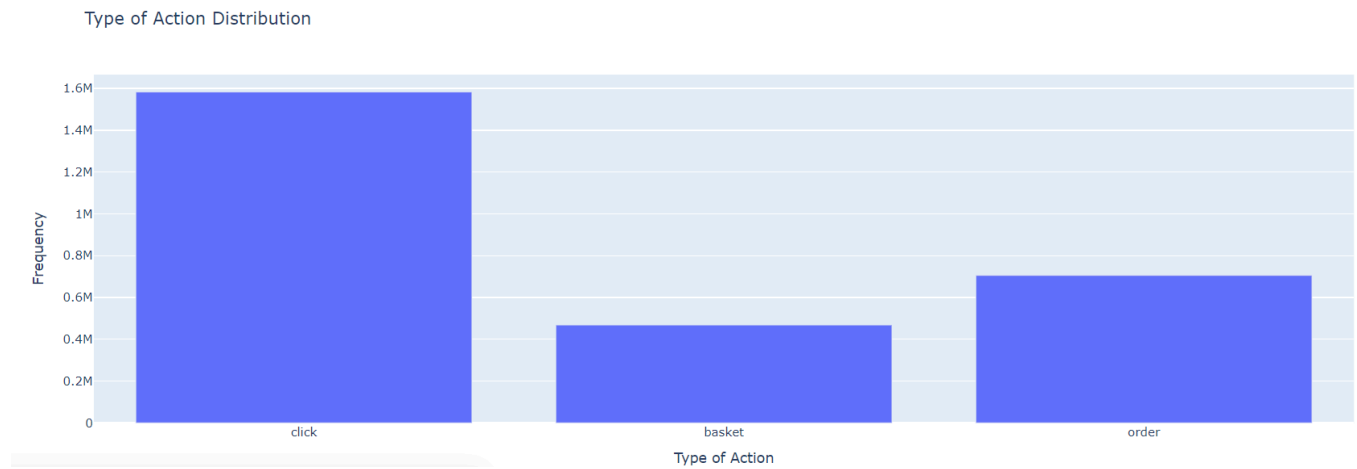


Fig. 3

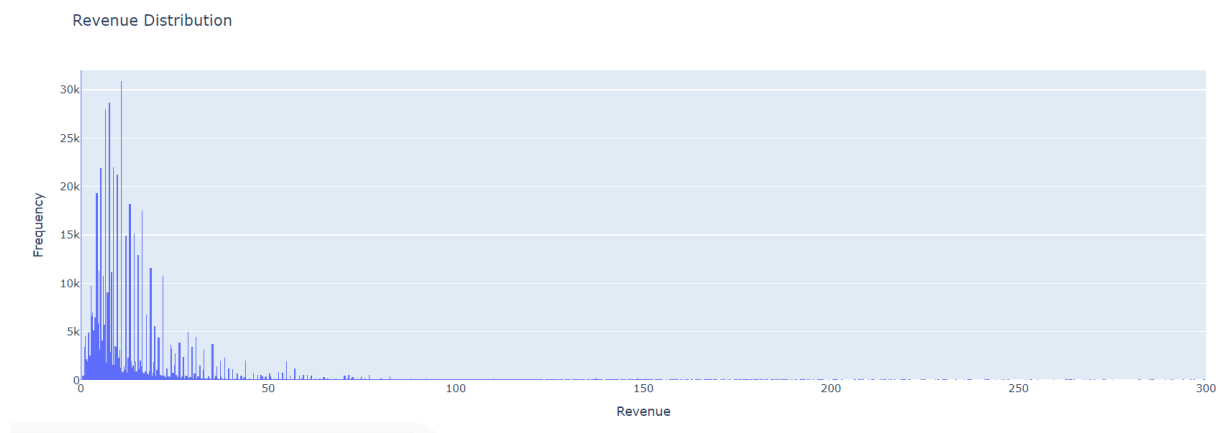


Fig. 4

IV. TECHNICAL DETAILS OF THE PROPOSED SYSTEM

A. Feature Engineering (Novel Framework)

Feature engineering played a pivotal role in transforming raw data into meaningful inputs for our predictive model. To address high-cardinality categorical features, we developed a systematic binning approach. Columns with more than 10 unique values, such as group, pharmForm, manufacturer, and category, were processed by calculating their frequency distribution and grouping categories into four bins based on quantiles. This method ensured that the most frequent categories were prioritized, reducing complexity while preserving the data's intrinsic structure. Additionally, we aggregated user interaction data by product ID, deriving metrics such as total_clicks, total_baskets, and total_orders to capture user engagement and product popularity over time. From these, we computed action ratios like click_to_basket_ratio and basket_to_order_ratio to provide deeper insights into user conversion patterns at different sales funnel stages.

We further introduced competitor price features to quantify competitive positioning, including the relative price difference (price_competitiveness), the price-to-RRP ratio, and a binary flag (competitor_undercut_flag) indicating whether our price was lower than the competitor's. Temporal patterns were captured through features like day_of_week to account for variations in user activity across the week. To link user actions directly to revenue, we derived features such as revenue_per_click, revenue_per_basket, and revenue_per_order. Interaction features, such as the product price and availability or competitor price and availability, were included to reflect the combined impact of pricing and stock levels on customer decisions.

This feature engineering pipeline is a novel contribution of our project, thoughtfully designed to maximize the dataset's predictive power. By addressing challenges like high cardinality, incorporating competitor dynamics, and crafting derived and interaction features, our approach ensures the extracted features are both domain-specific and highly informative, ultimately enhancing the model's accuracy and interpretability.

B. Leveraging AutoML and GPU Acceleration for Efficient Model Building

Our dataset, comprising 2,756,003 rows and 11 columns, posed significant challenges for efficient model building due to its size and complexity. Handling such a large dataset requires robust computational resources and efficient methodologies to process the data, build predictive models, and identify optimal parameters. This is where AutoML, combined with the computational power of Google Colab Pro's GPU, became essential.

AutoML provides a systematic, automated approach to machine learning, which is particularly beneficial when working with large datasets. Manually selecting the best algorithm, tuning hyperparameters, and optimizing model performance across such a vast dataset can be time-consuming and prone to human error. AutoML automates these tasks by evaluating multiple algorithms and configurations, identifying the most suitable model, and fine-tuning hyperparameters to achieve the best performance.

However, AutoML's effectiveness is enhanced when paired with powerful computational resources like Google Colab Pro's GPU. Our dataset required significant memory and processing power for tasks such as data preprocessing, feature selection, and iterative model training. The GPU acceleration provided by Google Colab Pro significantly reduces the time required for these operations, enabling us to handle the dataset efficiently and conduct extensive experimentation. The combination of AutoML and Colab Pro's GPU allowed us to leverage advanced machine learning techniques without being constrained by hardware limitations, facilitating the exploration of multiple models and hyperparameter combinations within a reasonable timeframe. This combination allowed us to build robust models, find the most optimal parameters, and extract the best-performing configurations, all while efficiently managing the computational demands of a large dataset. This approach was not only necessary but also critical for deriving accurate and actionable insights from our dataset.

V. EXPERIMENTS

A. Models Explored

To predict revenue accurately, a variety of machine learning models were explored. These models ranged from simple decision trees to advanced ensemble methods and neural networks, each offering unique advantages:

1. Decision Tree:

- A basic model that splits data based on feature thresholds to predict outcomes.
- Strength: Easy to interpret and quick to train.
- Limitation: Prone to overfitting and struggles with complex relationships.

2. Random Forest:

- An ensemble of decision trees that improves robustness by averaging multiple tree outputs.
- Strength: Handles overfitting better and captures non-linear relationships.
- Limitation: May not perform as well with high-dimensional data.

3. Gradient Boosting:

- Sequentially builds decision trees, correcting errors from previous iterations.
- Strength: Effective for structured data and provides good generalization.
- Limitation: Computationally intensive.

4. XGBoost (Extreme Gradient Boosting):

- An optimized version of Gradient Boosting with faster training and better performance.
- Strength: Handles missing data and reduces overfitting with regularization.
- Limitation: Requires careful tuning of hyperparameters.

5. LightGBM (Light Gradient Boosting Machine):

- A highly efficient gradient boosting framework optimized for speed and resource usage.
- Strength: Scales well with large datasets and maintains accuracy.
- Limitation: This may be sensitive to imbalanced datasets.

6. CatBoost:

- A gradient boosting algorithm specifically designed for categorical data.
- Strength: Requires minimal preprocessing and works well with mixed data types.
- Limitation: Can be slower to train compared to XGBoost.

7. Deep Neural Networks (DNN):

- Multi-layered networks are capable of capturing complex, non-linear relationships.
- Strength: Highly flexible and powerful for large datasets.
- Limitation: Requires significant computational resources and hyperparameter tuning.

B. Model Selection

After experimenting with the above models, **XGBoost** and **LightGBM** emerged as the top performers:

- **Performance Metrics:**
 - **Root Mean Squared Error (RMSE):** Measures prediction accuracy, with lower values indicating better performance.
 - **R²:** Represents the proportion of variance explained by the model; higher values indicate stronger predictive power.
- **Results:**
 - XGBoost achieved the lowest RMSE (~1.11) and the highest R², making it the most accurate model.
 - LightGBM is closely followed with comparable RMSE and R², offering a faster and resource-efficient alternative.

C. Model Training and Evaluation

- **Train-Validation Split:**
 - The dataset was split into training and validation sets to evaluate the models on unseen data.
 - This approach ensures the model's ability to generalize beyond the training dataset.
- **Evaluation Metrics:**
 - **RMSE:** Captures the average magnitude of errors; ideal for assessing prediction accuracy.
 - **Mean Absolute Error (MAE):** Measures the average absolute error, offering an intuitive understanding of prediction errors.
 - **R²:** Indicates how well the model explains the variability in the target variable.
- **Cross-Validation:**
 - Applied k-fold cross-validation to assess model stability and robustness across different subsets of the data.
 - Ensures the model's performance is not biased by a particular train-test split.

D. Hyperparameter Tuning Using Optuna

Purpose of Hyperparameter Tuning

Hyperparameter tuning was essential to:

1. Improve model accuracy (lower RMSE and higher R^2).
2. Reduce overfitting or underfitting by finding the optimal balance between complexity and generalization.
3. Fine-tune key parameters for algorithms like XGBoost, LightGBM, and Random Forest.

Steps in Hyperparameter Tuning Using Optuna

1. **Define Objective Function:**

- The objective function evaluates a set of hyperparameters and returns a performance metric, such as validation RMSE.

2. **Search Space Definition:**

Optuna explores a predefined range of hyperparameters for each model. Key parameters tuned for this project include:

- **XGBoost:**

- **max_depth:** Depth of each tree.
- **learning_rate:** Step size for weight updates.
- **n_estimators:** Number of boosting rounds.
- **subsample:** Fraction of samples used for fitting individual trees.
- **colsample_bytree:** Fraction of features used for tree construction.

- **LightGBM:**

- **num_leaves:** Maximum leaves per tree.
- **learning_rate:** Step size for updates.
- **n_estimators:** Number of boosting rounds.
- **feature_fraction:** Fraction of features for boosting.

- **Random Forest:**

- **n_estimators:** Number of trees in the forest.
- **max_depth:** Maximum depth of each tree.
- **min_samples_split:** Minimum samples required to split a node.

3. **Optimization Algorithm:**

Optuna employs techniques like *Tree-Structured Parzen Estimator (TPE)* to efficiently navigate the hyperparameter space, focusing on promising regions.

4. **Trial Execution:**

Each trial represents a combination of hyperparameters. The library automatically records results and adjusts future trials based on previous outcomes.

5. **Best Parameters Selection:**

After running multiple trials, Optuna identifies the combination of hyperparameters that minimizes the objective function (e.g., RMSE).

Results from Optuna in the Project

- **XGBoost:**
 - Achieved optimal parameters with a validation RMSE of ~ 1.11 .
 - Demonstrated excellent generalization and predictive accuracy.
- **LightGBM:**
 - Comparable performance to XGBoost, offering a lightweight alternative with similar tuned parameters.
- **Random Forest:**
 - Improved performance after tuning but remained less accurate than boosting models.

CONCLUSION

In conclusion, we have successfully designed a comprehensive system for predicting revenue in e-commerce using dynamic pricing strategies. The system utilized a diverse set of features, including product attributes, user interactions, and competitor pricing, to capture the complex relationships that influence revenue generation.

Our methodology incorporated advanced Data Mining techniques and state-of-the-art machine learning algorithms. Through extensive experiments, we identified that ensemble models like XGBoost and LightGBM outperformed simpler approaches, such as Decision Trees, in terms of predictive accuracy and robustness. Among these, XGBoost demonstrated the best performance, achieving minimal RMSE and high R^2 scores.

Additionally, our analysis underscored the importance of addressing data quality issues, such as handling missing values, outliers, and skewed distributions. Novel feature engineering techniques, such as rolling averages and interaction features, further improved model performance over baseline methods.


Overall, this project highlights the effectiveness of advanced machine learning and data preprocessing techniques in solving revenue prediction challenges in e-commerce. Future work could explore incorporating real-time data streams and expanding the scope to include personalization strategies for further optimization.

REFERENCES

- [1] S. Gurnani, et al., “Optimizing Sales Forecasting in e-Commerce with ARIMA and LSTM Models,” Proceedings of the 14th International Conference on Machine Learning and Applications (ICMLA), pp. 295–301, 2023.
- [2] A. Bandara, et al., “E-Commerce Sales Revenues Forecasting Using DAG Neural Networks,” Journal of Business Analytics, vol. 15, no. 3, pp. 215–229, 2023.
- [3] F. Lu, et al., “Sentiment Analysis-Driven Sales Forecasting Models for E-Commerce,” Proceedings of the International Conference on Artificial Intelligence in Business, pp. 487–495, 2023.
- [4] P. Saxena, “Deep Learning-Based Prediction and Revenue Optimization for Online Platform User Journeys,” AIMS Press Journal of E-Commerce Research, vol. 12, no. 4, pp. 120–137, 2023.
- [5] D. Guo, et al., “Multivariate Intelligent Decision-Making Models for Retail Industry Sales Forecasting,” Scitepress Journal of Advanced Analytics, vol. 9, no. 2, pp. 50–62, 2023.

These papers highlight various revenue optimization and forecasting methods in e-commerce, utilizing deep learning, hybrid models, and data-driven approaches.

Link to Notebooks:

 [revenue-forecast-for-dynamic-pricing-ml-dnn.ipynb](#)

 [Time series Revenue Forecasting.ipynb](#)