Customer Conversion Analysis Prediction Model

# Abstract

This project focuses on analyzing online customer behavior by leveraging clickstream data. It involves preprocessing and cleaning data, engineering relevant features, selecting significant attributes through statistical analysis, and training multiple machine learning models for classification, regression, and clustering tasks. The developed Streamlit web application enables businesses to predict customer conversions, estimate potential revenue, and perform customer segmentation, aiding in targeted marketing efforts and enhancing customer engagement.

# Introduction

## Project Background

* Clickstream data offers insights into customer interactions online. In competitive e-commerce, understanding such data helps businesses tailor marketing. Effective analysis improves customer retention and satisfaction. This project leverages these insights for strategic decision-making.

## Objective of the Project

* Predict customer conversion accurately.
* Estimate potential revenue per customer.
* Segment customers for targeted marketing.
* Develop an interactive analytical web application.

## Project Scope And Limitations

* The scope covers data preprocessing, modeling, and app development. It integrates classification, regression, and clustering techniques. Limitations include data completeness, size constraints, and computational resources. Findings depend significantly on data quality.

# Methodology

## DataPreprocessing

* Checked for null values; confirmed no null values present in the dataset.
* Checked for duplicate records and removed any identified duplicates.
* Performed encoding for categorical columns using appropriate encoding techniques.
* Checked for outliers and addressed them accordingly.
* Removed unnecessary columns that did not contribute to model performance.
* Applied SMOTE (Synthetic Minority Oversampling Technique) to address class imbalance in the dataset.

## Feature Engineering

In this phase, several new meaningful features were created to enhance model performance:

* **Purchase Completed:** Identified whether browsing behavior indicates a completed purchase based on page visits, price interactions, and click counts.
* **Weekend Indicator:** Created a binary feature identifying if browsing occurred during weekends to capture weekend-specific shopping behaviors.
* **Total Clicks per Session:** Calculated the total number of clicks made by a user in a single session to measure engagement intensity.
* **Maximum Page Reached:** Identified the highest page number a user navigated to, indicating browsing depth and interest level.

These engineered features provided critical insights into user interactions, significantly enhancing the accuracy of predictive models.

## Feature Selection

Feature selection involved statistical testing to identify the most impactful variables for modeling. Specifically:

* **ANOVA Test:** Applied to categorical variables to determine their relationship with continuous outcomes.
* **Chi-square Test:** Evaluated relationships between categorical features and target classification.
* **T-tests:** Analyzed continuous features to assess differences between groups.
* **Correlation Heatmap:** Visualized inter-feature correlations to avoid multicollinearity and redundancy.

This rigorous selection process ensured inclusion of only significant and non-redundant features, improving model efficiency and predictive accuracy.

## Exploratory Data Analysis (EDA)

## Model Selection and Algorithms Used

In the modeling phase, multiple algorithms were tested across classification, regression, and clustering tasks to identify the best-performing models:

### **Classification Algorithms**

* **Logistic Regression:** Chosen for interpretability and baseline performance.
* **Random Forest:** Evaluated due to its robustness and handling of non-linear relationships.
* **XGBoost:** Selected for its high predictive accuracy and efficiency.

### **Regression Algorithms**

* **Linear Regression:** Used for baseline performance evaluation and simplicity.
* **Random Forest Regressor:** Chosen for its ability to capture complex, non-linear patterns.
* **XGBoost Regressor:** Tested for superior accuracy and performance on large datasets.

### **Clustering Algorithms**

* **K-Means:** Applied to clearly segment customers into distinct groups.
* **DBSCAN:** Used to identify outlier segments and handle noise in data.
* **Hierarchical Clustering:** Selected for analyzing nested customer groups and relationships.

Each algorithm was evaluated using specific metrics tailored to the task type, ensuring the selection of optimal models for accurate predictions and insightful customer segmentation.

## Pipeline and Automation Tools

The project utilized advanced pipeline and automation tools to streamline data processing, modeling, and deployment:

* **ZenML:** Employed to automate and orchestrate the machine learning workflows, ensuring seamless data ingestion, preprocessing, modeling, and evaluation.
* **MLflow:** Integrated for efficient experiment tracking, versioning of models, and managing model deployments. This tool provided comprehensive logging of model parameters, metrics, and artifacts.

These automation tools significantly enhanced efficiency, reproducibility, and scalability.

## Deployment Techniques

Deployment involved modern techniques ensuring scalability, consistency, and reliability:

* **Docker Containers:** Used for packaging applications, ensuring consistent environment setups across various deployment stages.
* **AWS Elastic Container Service (ECS):** Leveraged ECS for efficient orchestration, management, and scaling of Docker containers in a production environment.
* **Streamlit:** Provided an interactive web-based interface, facilitating user-friendly access to the predictive models and analytical insights.

This comprehensive deployment strategy ensures robust and efficient operation of the analytical solution.

# Implementation

## Development Environment

The project was developed using Python and key libraries such as Pandas, NumPy, Scikit-learn, XGBoost, and TensorFlow. Streamlit was used for web app development, and Jupyter Notebook facilitated interactive analysis. Version control was managed through Git and GitHub.

## Pipeline Orchestration (ZenML)

ZenML was employed to orchestrate the entire machine learning pipeline, streamlining processes from data ingestion to model deployment. ZenML facilitated efficient automation, reproducibility, and modular management of pipeline components, significantly reducing manual interventions.

## Experiment Tracking and Model Registration (MLflow)

MLflow was integrated for experiment tracking, allowing systematic logging of model parameters, performance metrics, and model artifacts. It ensured transparency, version control, and ease in model comparison, aiding in the selection and registration of the best-performing models.

## Containerization and Docker File

Docker was utilized for containerizing the Streamlit application along with all dependencies. Dockerfiles were configured to package Python applications, libraries, and system requirements, ensuring consistent environments across development and deployment stages.

## Docker Compose Configuration

Docker Compose was leveraged for orchestrating multi-container Docker applications locally, facilitating efficient development and testing. It simplified the management of services such as application servers and databases by defining their dependencies, networks, and configurations within a single file.

## Deployment On Cloud (AWS ECS)

AWS Elastic Container Service (ECS) was chosen for deploying Docker containers in a scalable and reliable cloud environment. ECS efficiently managed Docker containers, automated scaling, load balancing, and provided robust resources to handle user traffic and computational demands reliably.

# Results and Evaluation

## Evaluation Metrics

* **Classification:** Evaluated using Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics to ensure robust prediction capabilities.
* **Regression:** Metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R²) were used to assess prediction accuracy.
* **Clustering:** Evaluated using Silhouette Score and Davies-Bouldin Index to measure cluster quality and cohesiveness.

## Model Performance

* Classification models showed high accuracy, with XGBoost outperforming others.
* XGBoost Regressor provided the highest R² score, indicating superior revenue prediction capabilities.
* K-Means Clustering had the best silhouette scores, clearly segmenting distinct customer groups.

## Application Results

* The deployed Streamlit application successfully enabled users to perform real-time analysis, accurately predicting customer conversion rates, estimating revenue, and displaying distinct customer segments visually.
* Users reported improved marketing campaign effectiveness by targeting segmented user groups based on insights derived from the app.

## Streamlit Application Demonstration

The Streamlit application offered intuitive interactive features:

* **User Input and CSV Upload:** Allowed users to upload data or manually enter inputs.
* **Real-time Prediction:** Provided instant predictions for conversions and estimated revenue.
* **Interactive Visualization:** Enabled users to explore segmentation results visually through plots such as bar charts, pie charts, and histograms.
* **User-friendly Interface:** Simplified navigation and understanding of complex analytical results, enhancing usability for non-technical users.

# Conclusion

## Summary of Achievements

* The project successfully developed an integrated solution providing predictive insights on customer conversion, revenue potential, and segmentation for targeted marketing. The implementation utilized advanced automation, containerization, and deployment practices to ensure scalability, reproducibility, and ease of use.

## Future Scope and Recomendations

* Moving forward, there are several exciting opportunities to enhance this project further. Incorporating real-time data analytics could significantly improve the responsiveness and accuracy of predictions. Implementing deep learning methods could provide deeper insights and enhance predictive capabilities. Additionally, integrating Apache Airflow would help manage complex data workflows more efficiently, while further Dockerizing the project environment will enhance deployment and scalability. Extending the application to mobile platforms and consistently gathering user feedback will also help continuously refine the model's accuracy and overall user experience.

# References