Clickstream Predictive Analytics Project Report

1. Project Overview and Objectives

This project utilized machine learning techniques to analyze e-commerce clickstream data, aiming to understand and predict user behavior within a session. The primary goal was to deploy a predictive tool that provides three core insights for every user session:

1. **Classification:** Predict the likelihood of a user viewing a high-priced item (`price_2`).
2. **Regression:** Estimate the potential revenue generated by the session (`session_revenue`).
3. **Clustering:** Segment the customer base for targeted marketing efforts.

The outcome is a fully deployed, interactive Streamlit application (`app.py`) capable of processing real-time manual inputs or large batch files.

2. Methodology (Model Development - ClickStream.ipynb)

2.1 Data Preprocessing and Feature Engineering

The initial phase focused on transforming raw clickstream data into features suitable for modeling.

- **Handling Categorical Data:** Features such as `country`, `main_category_mode`, and `colour` were treated as categorical variables.
    - `model_photography` (Yes/No) was encoded using Label Encoding (Yes=1, No=0).
    - Other nominal categorical features were handled using One-Hot Encoding (OHE) to prevent the models from assuming any false ordinal relationship.
- **Feature Scaling:** Numerical features, primarily `total_clicks`, were scaled using a `StandardScaler` to ensure all features contributed equally to model training, particularly for the K-Means clustering algorithm.

2.2 Model Training and Selection

Three distinct machine learning tasks were executed:

Model Task	Target Variable	Algorithm Used	Outcome
Classification	`price_2` (Binary: 0 or 1)	Optimal Classification Model (e.g., Random Forest or Gradient Boosting) via Grid Search	`best_clf_pipeline_final.pkl`
Regression	`session_revenue` (Continuous)	Optimal Regression Model (e.g., Ridge or Lasso) via Grid Search	`best_reg_pipeline_final.pkl`
Clustering	N/A (Unsupervised)	K-Means Clustering	`customer_segmenter_kmeans.pkl`

Key Technique: Pipelines All final models were saved as `Pipeline` objects. This ensures that the exact steps of scaling, encoding, and final model prediction are executed in the correct sequence on new, unseen data, preventing data leakage and misalignment issues.

3. Deployment (Streamlit Application - app.py)

The final, trained models and the fitted `StandardScaler` were exported as `.pkl` files and integrated into the deployment application.

3.1 Architecture

The `app.py` script follows a standard deployment architecture:

1. **Artifact Loading:** Load all four `.pkl` files (`best_clf_pipeline_final.pkl`, `best_reg_pipeline_final.pkl`, `customer_segmenter_kmeans.pkl`, `fitted_scaler.pkl`) on startup.
2. **Feature Contract:** Utilizes a strict `FEATURE_COLS` list containing 67 OHE-aligned features to ensure data integrity.
3. **Data Transformation Function:** The core `transform_input` function ensures any user-provided data (manual or batch) is converted back into the 67-feature structure required by the models.

3.2 User Interface and Functionality

The Streamlit interface is divided into three tabs:

- **Manual Session Prediction:** Allows real-time testing of a single user session by adjusting sliders and drop-down menus (e.g., Total Clicks, Country, Colour). Results are displayed instantly via metric cards.
- **Batch File Prediction:** Enables users to upload a production CSV file containing thousands of raw sessions. The application processes the entire batch, applies all three models, and provides a downloadable CSV with added prediction columns (Probability, Revenue, Segment ID).
- **Analysis & Visualization:** Provides visual summaries of the batch results, including charts for Conversion Prediction counts, Revenue Distribution (Histogram), and Customer Segment distribution (Pie Chart).

4. Key Results and Business Impact

The deployed application provides immediate, actionable insights for e-commerce strategy:

- **Targeting Efficiency:** The Classification model allows marketing teams to focus high-value advertising resources only on sessions predicted to have a high likelihood of viewing high-priced items.
- **Resource Allocation:** The Regression model helps forecast potential sales volume based on current traffic and session metrics, aiding inventory and sales team planning.
- **Personalization:** The Clustering model segments users (e.g., "High-Engagement Shopper," "Price-Sensitive Explorer"). This segmentation is used to personalize content, offers, and site navigation immediately upon segment identification.

5. Conclusion

This project successfully transitioned from raw data analysis in a notebook to a robust, tri-model predictive deployment. The application is ready to be integrated into an operational environment to provide real-time, data-driven decisions based on user clickstream behavior.

Appendix: Core Files and Artifacts

File Name	Role	Description
`ClickStream.ipynb`	Development	Contains all data loading, cleaning, training, grid search, and model selection code.

`app.py`	Deployment	The Streamlit script that loads the `.pkl` models, runs predictions, and manages the interactive web interface.
`best_clf_pipeline_final.pkl`	Artifact	Trained classification model pipeline for predicting `price_2`.
`best_reg_pipeline_final.pkl`	Artifact	Trained regression model pipeline for estimating `session_revenue`.
`customer_segmenter_kmeans.pkl`	Artifact	Trained K-Means model for customer segmentation.