Beverage Price Prediction

Instantly estimate beverage prices for any customer

Problem Statement

- Accurately pricing beverages is crucial for maximizing revenue and market penetration, yet it remains a challenge due to varying consumer demographics, preferences, and buying behavior.
- Conventional pricing methods often overlook these nuances, leading to suboptimal pricing decisions and missed opportunities.
- There is a clear need for a data-driven approach that empowers businesses to predict optimal price ranges based on customer profiles—enabling smarter, personalized, and more competitive pricing strategies.

Project Objectives

- Develop a machine learning model to predict optimal price ranges for beverages based on customer demographics and behavioral inputs.
- Integrate the pipeline with MLflow for robust experiment tracking, model management, and reproducibility.
- Develop an intuitive Streamlit-based user interface to facilitate quick and accurate price predictions for stakeholders.
- Deploy the model on the cloud to enable access from any location.

Dataset Overview

1. Dataset Summary

- **Total Records:** 30010 respondents.
- **Target Variable:** price_range (Categorical price buckets).
- **Goal:** Predict price range based on customer profile and preferences.

2. Key Feature Categories

- **Demographics:** respondent_id, age, gender, zone, occupation, income_levels
- **Consumption Behavior:** consume_frequency(weekly), preferable_consumption_size, typical_consumption_situations
- **Brand Awareness & Loyalty**: current_brand, awareness_of_other_brands, reasons_for_choosing_brands
- **Product Preferences:** flavor_preference, packaging_preference, health_concerns
- **Buying Behavior:** purchase_channel
- **Target:** price_range

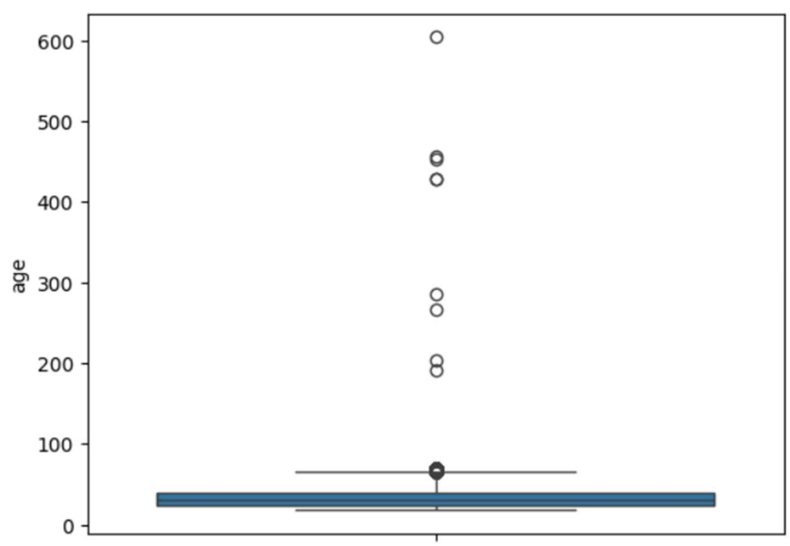
Data cleaning

1. Removing Duplicates

- To ensure data quality and avoid biased learning, duplicate entries were identified using the respondent_id and other key feature combinations.
- All exact duplicates were removed to maintain the integrity of individual responses.

2. Outlier Detection

- Outliers were examined in the **age** column using a box plot visualization.



- Entries with ages above 100 were removed to maintain realistic consumer data.

3. Handling Missing Data

- Initial null value check revealed missing entries across multiple columns.

```
respondent_id                        0
age                                  0
gender                               0
zone                                 0
occupation                           0
income_levels                     8062
consume_frequency(weekly)            8
current_brand                        0
preferable_consumption_size          0
awareness_of_other_brands            0
reasons_for_choosing_brands          0
flavor_preference                    0
purchase_channel                    10
packaging_preference                 0
health_concerns                      0
typical_consumption_situations       0
price_range                          0
dtype: int64
```

- **income_levels:** Missing values were replaced with "Not Reported" to retain entries without introducing bias.
- **consume_frequency(weekly)** and **purchase_channel:** Missing values filled using the mode (most frequent value) after analysis.

4. Correcting spelling mistakes in categorical data

- Checked unique values in each categorical column.
- Cleaned and standardized inconsistent values in **zone** and **current_brand** columns to maintain uniform labeling across records.

Feature engineering

1. Categorizing Age into Groups

- Created a new column **age_group** by binning the existing **age** column into defined age brackets.
- Ensured each entry was mapped to the appropriate group.
- Dropped the original age column post-transformation to eliminate redundancy.

2. Creating cf_ab_score (Consumption & Awareness Score)

- Introduced a new feature **cf_ab_score** to combine **consume_frequency(weekly)** and **awareness_of_other_brands** into a single score.
- Assigned numeric values to both inputs based on predefined categories.
- Calculated a combined score and rounded it to two decimal places for consistency.

3. Creating zas_score (Zone Affluence Score)

- Developed a new metric **zas_score** to reflect consumer affluence by combining geographic and income data.
- Assigned weighted scores to both **zone** and **income_levels** based on their economic indicators.
- Calculated a composite score to represent purchasing power and regional influence.

4. Creating bsi (Brand Switching Indicator)

- Introduced a binary indicator **bsi** to flag respondents likely to switch brands.
- Marked as 1 if the **current_brand** is not Established and **purchase_reasons** include Price or Quality.
- Helps identify price- or quality-sensitive consumers for targeted strategies.

5. Removing Logical Outliers

- Used a pivot table to examine relationships between **occupation** and **age_group.**

occupation age_group	Entrepreneur	Retired	Student	Working Professional
18-25	535	0	7328	2605
26-35	1826	0	697	6570
36-45	1619	0	0	4353
46-55	799	0	0	2167
56-70	221	1130	35	106

- Detected anomalies such as students in the 56–70 age group, which are unlikely in real-world scenarios.
- Removed such records to maintain data quality and analytical accuracy.

Model training

1. Preparing Features and Target Variables

- Defined feature matrix X and target variable y.
- Excluded identifier **respondent_id** and the target **price_range** from the feature set.

2. Data Splitting

- Split the dataset into 75% training and 25% testing using train_test_split to evaluate generalization performance.

3. Feature Encoding

- Applied Label Encoding to selected ordinal features: **age_group**, **income_levels**, **health_concerns**, **consume_frequency(weekly)**, **preferable_consumption_size and awareness_of_other_brands.**
- Used One-Hot Encoding for all other categorical features.
- Label encoded the target column **price_range**.

4. Model Benchmarking

- Tested multiple classification algorithms on the processed dataset to identify the best performer: **Gaussian Naive Bayes**, **Logistic Regression**, **Support Vector Machine (SVM)**, **Random Forest**, **XGBoost** and **Light Gradient Boosting Machine (LightGBM).**

5. Performance Evaluation

- Evaluated models using accuracy and classification report.

6. Final Model Selection

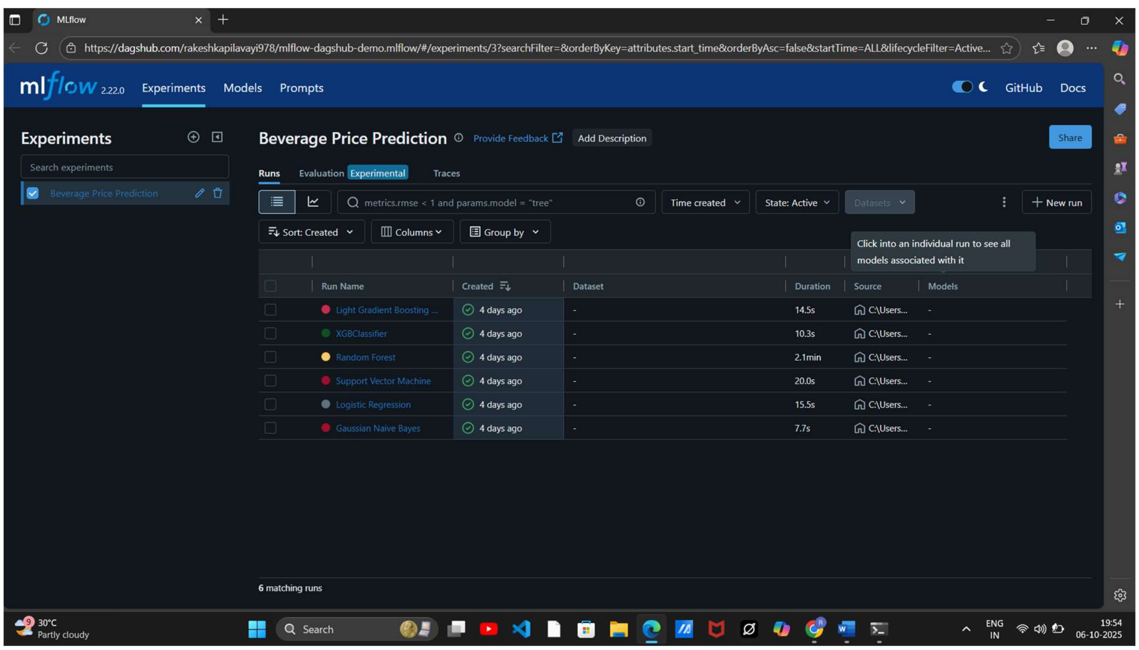
- Based on performance metrics, **XGBoost** was selected as the final model.
- It provided the best balance of accuracy and generalization on the test set.

- Further **hyperparameter tuning using Optuna** improved the model's performance, leading to optimal parameter settings.
- The tuned XGBoost model provided the best balance between **accuracy and generalization** on the test set.

```
Best accuracy: 0.9257577780745093
Best params: {'booster': 'gbtree', 'lambda': 0.34140224118899204, 'alpha': 0.3533784335530234, 'colsample_bytree': 0.7363130649148815, 'subsample': 0.8071117485423065, 'learning_rate': 0.15964322063178146, 'n_estimators': 498, 'max_depth': 6, 'min_child_weight': 4, 'gamma': 0.35515405907448655}
```

MLflow Deployment and Tracking

- To streamline model experimentation and enable reproducible results, **MLflow** was integrated into the pipeline. Each classification model was logged using **MLflow** to capture their parameters, evaluation metrics, and artifacts.
- The tracked models were then published to **DagsHub**, providing a centralized platform for versioning and sharing. This setup allows easy comparison of model performance through a visual **MLflow** dashboard.

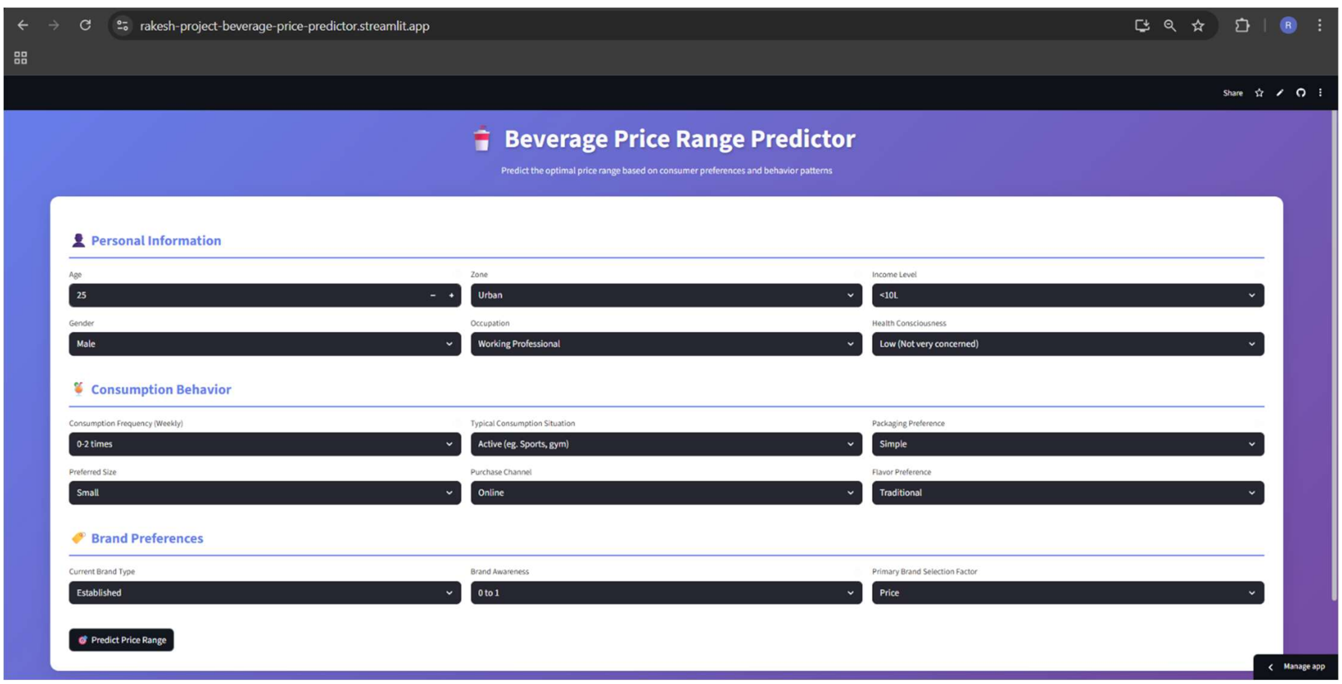


- **MLflow Dashboard:** https://dagshub.com/rakeshkapilavayi978/mlflow-dagshub-demo.mlflow/#/experiments/3?searchFilter=&orderByKey=attributes.start_time&orderByAsc=false&startTime=ALL&lifecycleFilter=Active&modelVersionFilter=All+Runs&datasetsFilter=W10%3D

Streamlit App Integration

- Developed an interactive web application using **Streamlit**.
- Integrated the trained model to enable real-time beverage price prediction based on user inputs.
- Handled data preprocessing within the app to ensure accurate and consistent predictions.
- Implemented logic to dynamically predict beverage price ranges (₹50–100, ₹100–150, ₹150–200, ₹200–250) based on engineered features.
- Deployed the app on **Streamlit** Cloud for easy public access.
- The app allows users to input details such as age group, income level, and consumption habits to instantly receive a personalized price range.

User Interface Preview



- RAKESH KAPILAVAYI