User Behavior Analytics Model

Overview

This project aims to predict the dealer(s) most likely to fulfill an order for a specific material based on various factors such as state, season, festival, and price range. The project utilizes machine learning techniques to cluster dealers and predict which dealers are best suited to fulfill new orders. It combines **KMeans Clustering** and feature engineering for effective dealer prediction.

Key Components:

1. **KMeans Clustering:** This technique groups similar dealers based on their transaction patterns, allowing us to predict the dealers most likely to fulfill an order for a specific material, based on order attributes.
2. **Model Training and Prediction:** Using the KMeans clustering model, predictions are made by leveraging relevant features like the state, festival flag, season flag, and material price range.

Data

The project utilizes three primary datasets:

1. **Dealer Data (UBA_F.csv):**
    a. Contains information about dealer transactions, including `dealer_id`, `material_no`, `order_id`, `quantity`, `FINAL_PRICE`, `state_id`, and `ORDER_DATE`.
2. **Festival Data (festival_data.csv):**
    a. Contains festival dates (`Date`) and festival names (`FestivalName`). Festivals are categorized as either national or local.
3. **Off-Season Data (off_season_data.csv):**
    a. Contains dates (`weather_date`) and off-season flags (`off_season`), indicating whether a particular date falls within an off-season period.

Data Preprocessing

1. **Date Conversion:**
    a. The `ORDER_DATE` column in the dealer dataset and the `Date` column in the festival dataset are converted into `datetime` format to enable efficient comparisons.
2. **Festival and Off-Season Flags:**
    a. Two binary columns, `festival_flag` and `season_flag`, are added to the dealer dataset. The `festival_flag` indicates whether an order falls on a festival date, and the `season_flag` indicates if an order falls during an off-season.
3. **Price Categorization:**
    a. The `FINAL_PRICE` is categorized into ranges (1-8) to simplify the model's task and improve pattern recognition.
4. **Feature Engineering:**
    a. New columns `year` and `month` are derived from the `ORDER_DATE` to capture seasonal and temporal patterns.
5. **Data Splitting:**
    a. The dataset is split into training, validation, and test sets for model training and evaluation.

Modeling

KMeans Clustering

- **Purpose:** The KMeans algorithm is used to group dealers based on features such as `material_no`, `state_id`, `festival_flag`, `season_flag`, `year`, `month`, and price range. The number of clusters is determined by the number of unique dealers in the dataset.
- **Steps:**
    - **Prepare the Data:**
        - Remove the `dealer_id` column, as it is not needed for clustering.
        - Select the relevant features for clustering: `material_no`, `state_id`, `festival_flag`, `season_flag`, `year`, `month`, and `price_range`.
    - **Determine the Number of Clusters:** Set the number of clusters (`n_clusters`) equal to the number of unique dealers.

- **Train the KMeans Model:** Initialize the KMeans model with the determined number of clusters. Fit the KMeans model to the selected features in the dataset.
- **Assign Data to Clusters:** Use the trained KMeans model to predict the cluster assignments for each data point (dealer). Assign the cluster labels to the dataset.
- **Evaluate Clustering Quality**
  - Calculate **Inertia**: Use the `inertia_` attribute to compute the sum of squared distances from points to their assigned cluster centers. Lower inertia indicates better clustering.
  - Calculate the **Silhouette Score**: Measure how well each point fits its assigned cluster compared to other clusters. A score closer to 1 indicates better-defined clusters.

Evaluation of Trained Model

Clustering Evaluation:

1. **Inertia:** Inertia measures how compact the clusters are. A lower inertia value indicates better clustering.
2. **Silhouette Score:** This score quantifies how well-separated the clusters are. A score closer to 1 indicates well-defined clustering.

Accuracy Evaluation:

The predicted dealer IDs for a given order are compared with the actual dealer IDs.

Currently, we are working on fine-tuning the model to provide more accurate predictions. Based on our current analysis, when predicting dealers for material `4000071` in **December 2024** for **Tripura (state 19)**, there were 25 actual dealers who purchased during the month. The model correctly predicted 18 out of the 25 dealers, yielding an accuracy of **72%** for this case. Additionally, the **Silhouette Score** for clustering was calculated to be **0.80**, indicating well-separated clusters and strong model performance.

Conclusion

This project offers an efficient method for predicting which dealers are most likely to fulfill a given order based on a variety of factors, such as material type, state, time, and seasonal

or festival conditions. While KMeans clustering serves as the core algorithm for dealer prediction, further improvements could be made by exploring alternative machine learning models and more advanced feature engineering. The methodology can be applied to other industries that require supplier or partner prediction.