Report

Define

The project aims to analyze promotional offers and their influence on purchasing decisions using a dataset simulated to mimic real-world consumer behavior. The dataset comprises three main components:

1. Portfolio: Information about the promotional offers.
2. Profile: User demographic and membership data.
3. Transcript: Event logs capturing user interactions with offers and transactions.

Objectives

The main objective is to determine which types of users are most responsive to different types of offers (Buy One Get One (BOGO), Discount, Informational) and how to best present these offers to maximize user engagement and purchases.

Data Dictionary

1. Profile Data:
   - `gender`: Gender of the user (M, F, O, or null).
   - `age`: Age of the user (with 118 indicating missing values).
   - `id`: Unique identifier for the user.
   - `became_member_on`: Date the user became a member, in YYYYMMDD format.
   - `income`: Annual income of the user.
2. Portfolio Data:
   - `reward`: Monetary reward for completing the offer.

    - `channels`: Channels through which the offer was delivered (web, email, mobile, social).
    - `difficulty`: Spending required to complete the offer.
    - `duration`: Duration the offer is valid, in days.
    - `offer_type`: Type of offer (bogo, discount, informational).
    - `id`: Unique identifier for the offer.
3. Transcript Data:
    - `person`: Unique identifier for the user.
    - `event`: Type of event (offer received, offer viewed, transaction, offer completed).
    - `value`: Dictionary containing either `offer_id` or `amount` depending on the event type.
    - `time`: Time in hours since the start of the dataset.

Analyze

Data Import and Overview

The data was imported using `pandas` from three JSON files representing the portfolio, profile, and transcript datasets. The datasets provide comprehensive details on user demographics, offer specifics, and user interactions.

Merging Datasets

To create a cohesive dataset for analysis, the `profile` dataset was merged with the `transcript` dataset on the user ID. Subsequently, the `portfolio` dataset was merged to incorporate offer details. During this process, several steps were taken:

1. Data Cleaning:
    - Missing values in `gender` and `income` were noted.
    - Special handling for missing values in `age` encoded as 118.
    - Extraction of `offer_id` and `amount` from the `value` column in the `transcript` dataset.
    - Conversion of `became_member_on` from string to datetime format.
2. Feature Engineering:

   - Split the `channels` column into separate boolean columns (`web`, `email`, `mobile`, `social`).
   - Calculated the time of transaction completion after viewing an offer (`com_trans_time`).
   - Identified transactions that followed viewing an offer and the corresponding offer type.
   - Combined information from `offer_for_purchase` and `viewed_offer_type` to create a comprehensive `offer_type` column.
3. Aggregating Data:
   - Grouped data by user (`person`) to analyze cumulative behaviors such as total rewards earned, average spending difficulty, and the number of purchases following an offer.

Exploratory Data Analysis (EDA)

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

1. Distribution of Purchases:
   - Analyzed the distribution of purchases following viewed offers.
   - Found that users often did not purchase after viewing an offer, though some engaged with multiple offers.
2. Offer Type Analysis:
   - Examined the relationship between `offer_type` and subsequent purchases.
   - Observed that `discount` offers, though not the most viewed, led to the highest number of purchases.
3. Categorical Variables:
   - Examined categorical variables such as `gender`, `event`, and `offer_type` to understand their distributions and potential impacts on purchasing behavior.

Data Preprocessing

To prepare the data for machine learning, several preprocessing steps were implemented:

1. Handling Missing Values:
    - Filled missing values in `age` and `income` with median values.
2. Encoding Categorical Variables:
    - Applied one-hot encoding to categorical columns like `gender` and `offer_type`.
3. Standardizing and Scaling:
    - Standardized continuous variables such as `age`, `income`, `reward`, and `difficulty` using `StandardScaler`.

Splitting the Data

The data was split into features (`X`) and the target variable (`y`), which represents the `purchased_after_offer_comb`. This target variable was encoded into numerical categories for modeling purposes. A train-test split was conducted with an 80-20 ratio to ensure the model's performance could be evaluated on unseen data.

By following these steps, the data was cleaned, features were engineered, and a structured dataset was prepared for implementing machine learning models to predict user responsiveness to offers.

Implement

Model Selection and Benchmarking

Two primary machine learning models were selected for implementation: Logistic Regression and Random Forest Classifier. These models were chosen due to their effectiveness in classification tasks and their interpretability.

Logistic Regression

A Logistic Regression model was implemented as a baseline model:

1. Pipeline Construction:
    - Combined preprocessing steps (standardization and one-hot encoding) with the Logistic Regression model into a single pipeline using `make_pipeline`.

2. Model Training:
    - Trained the Logistic Regression model on the training dataset (`X_train`, `y_train`).
3. Model Evaluation:
    - Made predictions on the test dataset (`X_test`).
    - Evaluated model performance using accuracy and weighted F1 score to handle the class imbalance.

Random Forest Classifier

To improve upon the baseline model, a Random Forest Classifier was implemented:

1. Pipeline Construction:
    - Combined the same preprocessing steps with the Random Forest Classifier.
2. Model Training:
    - Trained the Random Forest model on the training dataset.
3. Model Evaluation:
    - Made predictions on the test dataset and evaluated performance using accuracy and weighted F1 score.

Hyperparameter Tuning

Hyperparameter tuning was conducted for the Random Forest model to optimize its performance:

1. Grid Search:
    - Defined a grid of hyperparameters to search, including the number of trees (`n_estimators`) and the maximum depth of the trees (`max_depth`).
    - Used `GridSearchCV` to perform a cross-validated search over the parameter grid to find the best combination of hyperparameters.
2. Evaluation:
    - Evaluated the best model found by grid search using accuracy and F1 score.

Feature Importance

To understand which features contributed most to the model's predictions, feature importance was analyzed:

1. Extract Feature Importances:
   - Retrieved feature importances from the best Random Forest model found by grid search.
2. Visualization:
   - Visualized the feature importances to identify key predictors of user responsiveness to offers.

Results

The results of the model implementations are summarized below:

1. Logistic Regression:
   - Accuracy: Approximately 0.57
   - Weighted F1 Score: Approximately 0.46
2. Random Forest Classifier:
   - Accuracy: Approximately 0.55
   - Weighted F1 Score: Approximately 0.49
3. Hyperparameter Tuning:
   - Best Random Forest parameters: `n_estimators = 100`, `max_depth = None`
   - Best F1 Score from Grid Search: Slight improvement over the baseline Random Forest model
   - Weighted F1 score = 0.50
4. Feature Importance:
   - The most important predictors identified were `difficulty`, followed by `reward`, and various encoded categorical variables (e.g., `offer_type`, `gender`).

The Random Forest Classifier outperformed the Logistic Regression model, demonstrating higher accuracy and a better F1 score. The tuned Random Forest model further improved performance, highlighting the importance of hyperparameter optimization in machine learning tasks.

Conclusion

This project successfully identified key user traits and offer characteristics that influence purchasing decisions. By analyzing user interactions with different promotional offers, we could build predictive models to determine user responsiveness. The Random Forest Classifier, particularly with hyperparameter tuning, proved to be an effective model for this task.

Key Findings

1. Offer Type Impact:
    - Discount offers were found to be the most effective in driving purchases, despite being less frequently viewed than BOGO offers.
2. Feature Importance:
    - `difficulty` and `reward` were the most influential features in predicting offer responsiveness.
    - Demographic factors like `age` and `gender` also played significant roles.
3. Model Performance:
    - The Random Forest model outperformed Logistic Regression, especially after hyperparameter tuning, suggesting the importance of model complexity and optimization.

Future Work

To further enhance the analysis and modeling efforts, the following steps are recommended:

1. Incorporate Additional Features:
    - Explore additional features such as user behavior trends over time or more granular interaction data.
2. Advanced Modeling Techniques:
    - Investigate more complex models like Gradient Boosting Machines or Neural Networks for potentially better performance.
3. Real-World Testing:

- Validate the model's predictions in a real-world setting by applying it to actual promotional campaigns and comparing predicted responsiveness to observed behavior.

By continuously refining the models and incorporating new data, businesses can better tailor their promotional strategies to maximize customer engagement and sales.