User Activity Prediction

Objective

The goal of this Exercise is to transform raw datasets into user-level features that capture user behavior and engagement patterns. These features serve as input for machine learning models to predict user activity in July 2017 based on their behavior from April to June 2017. Provided dataset:

- Installs
- App Starts
- Brochure Views
- Brochure Views July
- App Starts July

Workflow Overview

1. Data Loading: Loading multiple datasets from specified file paths.
2. Exploratory Data Analysis (EDA): summary statistics, missing value checks, and basic data insights.
3. Data Preprocessing: Handling missing values and ensuring data consistency.
4. Feature Engineering: Aggregating features from different datasets to create features for training.
5. Data Preparation: Splitting the data into features (X) and target (y), scaling, and splitting into training/testing sets.
6. Model Training and Validation: Training RandomForestClassifier, GradientBoostingClassifier, and XGBoostClassifier models.
7. Evaluation: Evaluating models using accuracy, classification reports, confusion matrices, and feature importance.

Feature Engineering

- Aggregates app starts (count, active days).
- Aggregates brochure views (count, view duration, page turns).
- Adds encoded features for campaignId, model, and productId using LabelEncoder.
- Merges July activity data (is_active_july) as the target variable.
- Handling Missing Values
- Missing values resulting from joins were filled with 0 to indicate no activity.

Training

- Trains a model.
- Performs K-Fold Cross-Validation to calculate mean accuracy.
- Calculate training and testing accuracy.

Model Evaluation

- Accuracy, Classification Report
- Confusion Matrix visualizes true positives, true negatives, false positives, and false negatives. (visualized as a heatmap)
- Feature Importance Plot provides feature importance scores. A horizontal bar chart shows the importance of each feature in predicting is_active_july (for tree-based models).

Outputs

The models are evaluated based on:

- Cross-Validation Accuracy
- Training Accuracy
- Test Accuracy

Conclusion

The code implements a robust user activity prediction pipeline. This pipeline can be extended further for hyperparameter tuning, advanced feature engineering, and deployment in production environments.