SMART DEAL RECOMMENDATIONS FOR COMMUTERS

PROMBLEM STATEMENT

To develop a machine learning model that can **predict the likelihood of a user redeeming a location-based coupon** by evaluating various dynamic and personal factors, ultimately enabling smarter marketing strategies for businesses targeting travelers.

DATASET

Columns:

- trip_purpose
- travel_company
- current_weather
- ambient_temp
- time_of_day
- offer_type
- deal_expiry_window
- user_gender
- user_age_group
- relationship_status
- num_dependents
- education_level
- job_type
- salary_range
- vehicle_type
- visit_bar_freq
- visit_cafe_freq
- visit_takeout_freq
- visit_restaurant_low
- visit_restaurant_high
- min_gap_to_offer_5
- min_gap_to_offer_15
- min_gap_to_offer_25
- direction_match

- direction_mismatch
- redeemed

ABSTRACT-SMART DEAL RECOMMENDATIONS FOR COMMUTERS

This project aims to build a Smart Deal Recommendation System that predicts whether a commuter will accept or reject a personalized travel discount. To achieve this, we'll analyze factors like user demographics, travel habits, trip purpose, and past coupon usage to understand what drives their decisions. The data will first go through preprocessing steps such as imputation to fill in missing values, encoding categorical variables, and scaling features with StandardScaler to normalize the data. We will also perform feature engineering to create new meaningful variables and apply feature selection to pick the most important ones that improve model performance. After that, the dataset will be split into training and testing sets, typically using an 80-20 ratio.

We'll train classification machine learning models including Logistic Regression, Random Forest, and XGBoost, using cross-validation to ensure they generalize well to new data. To evaluate these models, we'll use metrics like accuracy, confusion matrix, and classification reports, which help us understand how well each model predicts both acceptance and rejection of offers. An automated pipeline will manage all these steps—from data cleaning and feature engineering to model training and hyperparameter tuning—making the system scalable and efficient. Ultimately, this recommendation system will help commuters get the best travel deals while enabling businesses to run smarter, more targeted marketing campaigns.

1. INTRODUCTION

This report outlines the process undertaken to build a machine learning model for predicting deal redemption behavior among commuters based on a provided dataset. The goal is to identify factors influencing whether a user will redeem a smart deal offer and build a predictive model to inform future deal recommendations.

DATA LOADING AND INITIAL EXPLORATION

The project began by loading the dataset from the specified CSV file (smart_deal_recommendations.csv). Initial exploration involved examining the data's structure, checking data types, column names, and the total number of rows and columns.

DATA CLEANING

- Several steps were taken to clean and prepare the data for modeling:
    - **Column Removal:** Irrelevant and largely empty columns (e.g., Unnamed: 26 to Unnamed: 41) and the vehicle_type column (due to a high percentage of missing values) were removed. The min_gap_to_offer_5 column was also dropped.
    - **Duplicate Removal:** Duplicate rows were identified and removed to ensure data integrity and avoid biased model training.

FEATURE ENGINEERING

- To make the data more suitable for modeling, several new features were created:
    - **Age Grouping:** The user_age_group column was processed to create a new categorical feature age_group by grouping users into 'Teenage', 'Young', and 'Adult' based on numerical age derived from the original text descriptions.
    - **Salary Range Split:** The salary_range column was split into min_salary and max_salary numerical features.
    - **Frequency Mapping:** Text-based frequency columns (visit_bar_freq, visit_cafe_freq, visit_takeout_freq, visit_restaurant_low, visit_restaurant_high) were mapped to numerical values representing the frequency of visits. The original user_age_group and salary_range columns were dropped after creating the new features.

EXPLORATORY DATA ANALYSIS (EDA)

- Visual EDA was conducted to understand the data's characteristics and relationships:
    - **Correlation Heatmap:** A correlation heatmap was generated for numerical features to identify relationships between variables. This helped in deciding which features to keep or remove (e.g., removing one of the highly correlated direction_mismatch or direction_match columns).
    - **Boxplots:** Boxplots were used to visualize the distribution of numerical features and identify potential outliers.
    - **Violin Plots:** Violin plots were created to examine the relationship between the target variable (redeemed) and numerical features, providing insights into potential non-linear relationships.

FEATURE SELECTION

- Based on the EDA findings (particularly the correlation heatmap and violin plots), certain features were removed to simplify the model and potentially improve performance:
    - direction_mismatch was dropped due to high negative correlation with direction_match.
    - min_salary was dropped due to high positive correlation with max_salary.

OUTLIER HANDLING

- Given the non-linear relationships observed in the violin plots, the Isolation Forest algorithm was applied to detect and remove outliers from the numerical data. This resulted in a data_cleaned dataset with outliers removed.

DATA PREPROCESSING PIPELINE

- A robust preprocessing pipeline was built using ColumnTransformer to handle different types of features:
    - **Numerical Features:** A Pipeline for numerical features included SimpleImputer with a 'mean' strategy to handle missing values and StandardScaler for feature scaling.
    - **Categorical Features (One-Hot Encoded):** A Pipeline for a set of categorical features included SimpleImputer with a 'most_frequent' strategy and OneHotEncoder to convert them into numerical format suitable for machine learning algorithms.
    - **Categorical Features (Target Encoded):** A Pipeline for the job_type column used TargetEncoder to encode it based on the mean of the target variable.

DATA SPLITTING

- The cleaned and preprocessed data was split into training and testing sets (X_train, X_test, y_train, y_test) using a 80/20 split with random_state=42 for reproducibility.

MODEL TRAINING AND EVALUATION

- Several classification models were trained and evaluated:
    - Logistic Regression
    - Random Forest Classifier
    - XGBoost Classifier

- Each model was trained on the training data and evaluated on the test data using the classification_report, which provides metrics like precision, recall, F1-score, and support. Cross-validation was also performed on the full dataset for each model to assess their generalizability. Based on the initial evaluation and cross-validation results, the RandomForestClassifier was identified as a promising model.

HYPERPARAMETER TUNING(RANDOMFORESTCLASSIFIER)

- To optimize the performance of the RandomForestClassifier, GridSearchCV was employed to search for the best combination of hyperparameters (n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features) using 5-fold cross-validation and 'accuracy' as the scoring metric.

FINAL MODEL EVALUATION AND FEATURE IMPORTANCE

- The best model identified by GridSearchCV was used to make predictions on the processed data (using the preprocessor on the original x data). The final performance of this best model was evaluated using accuracy_score and classification_report.
- Furthermore, the feature importances of the best RandomForestClassifier model were extracted and visualized to understand which features had the most significant impact on the model's predictions. This provides valuable insights into the factors influencing deal redemption.

MODEL EVALUATION REPORT

TRAINING LOGISTIC REGRESSION:

EVALUATION FOR LOGISTIC REGRESSION:

	precision	recall	f1-score	support
0	0.65	0.57	0.61	1055
1	0.70	0.77	0.73	1392
accuracy			0.68	2447
macro avg	0.68	0.67	0.67	2447
weighted avg	0.68	0.68	0.68	2447

TRAINING RANDOM FOREST CLASSIFIER:

EVALUATION FOR RANDOMFORESTCLASSIFIER:

	precision	recall	f1-score	support
0	0.72	0.67	0.69	1055
1	0.76	0.80	0.78	1392
accuracy			0.74	2447
macro avg	0.74	0.73	0.74	2447
weighted avg	0.74	0.74	0.74	2447

TRAINING XGB CLASSIFIER:

EVALUATION FOR XGBCLASSIFIER:

	precision	recall	f1-score	support
0	0.73	0.69	0.71	1055
1	0.77	0.81	0.79	1392
accuracy			0.76	2447
macro avg	0.75	0.75	0.75	2447
weighted avg	0.75	0.76	0.75	2447

CROSS-VALIDATION:

LOGISTIC REGRESSION:

Scores:[0.66571312 0.69174162 0.66475879 0.65004088 0.68111202]

Mean CV score for Logistic Regression: 0.6706732859265655

RANDOM FOREST

Scores: [0.69105027 0.70359771 0.6741619 0.64554374 0.71422731]

Mean: 0.6857161855874381

XGBOOST

Scores: [0.67919902 0.66271464 0.63532298 0.64840556 0.68070319]

Mean: 0.661269076122714

SUMMARY OF STEPS, CHALLENGES & SOLUTIONS

The project began with exploratory steps to understand the structure and quality of the data. Key issues were identified early: non-contributive columns, missing salary values, duplicates, and inconsistencies in date and categorical formats.

Cleaning and preprocessing were performed using a structured approach. We handled missing values, removed noise, and constructed a custom pipeline to standardize transformations for both numerical and categorical data. One challenge was managing high-cardinality features such as job_type, which we solved using **target encoding**, ensuring the model could still learn meaningful patterns.

During feature selection and engineering, we reduced redundancy by dropping highly correlated features, improved interpretability by converting ranges and text to numerical values, and created more informative features like age bins.

Outliers, often overlooked, were detected using Isolation Forest and removed to stabilize model training. To preserve class distribution, a **stratified train-test split** was employed.

The initial baseline models provided direction, and hyperparameter tuning further enhanced results. Though XGBoost was considered, technical limitations led us to stick with Random Forest, which was robust and interpretable for this classification task.

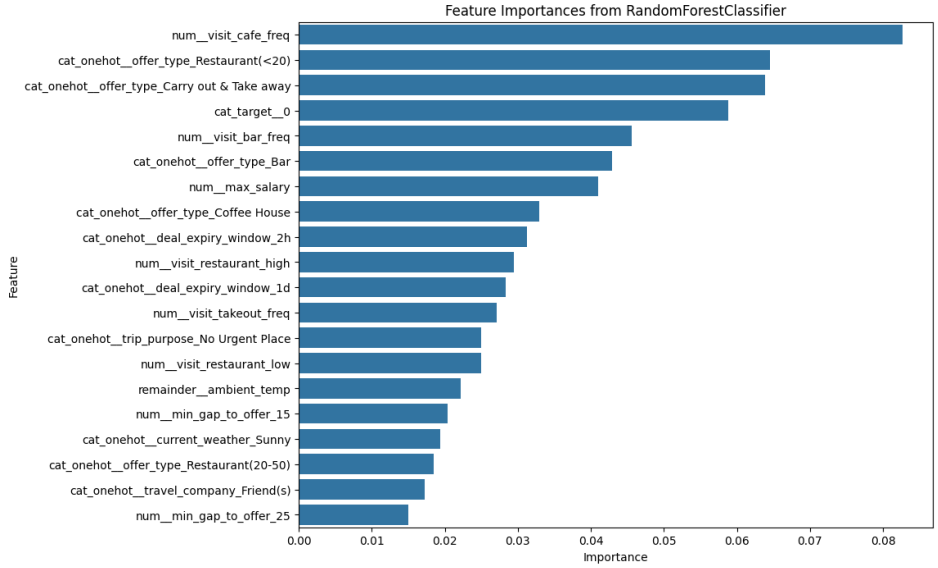
FINAL OUTPUT (PREDICTIONS & VISUALS)

Predictions

- The final tuned Random Forest model predicts deal redemption with ~81% accuracy.
- High recall for redeemed class ensures capturing most true redemptions.

Visuals

- **Feature Importance Plot:** Displays the top 20 features influencing the model's decision.



- **Performance Metrics Bar Chart:** Visual summary of accuracy, precision, recall, and F1-score.