AIM 511: Course Project - Mexican Tourist Profiles

Team: BYTE ME

Introduction

This project aims to predict the spending category—low, medium, or high—of tourists visiting Mexico, based on demographic and trip-related data. By analyzing travelers' trip details, demographics, and package choices, the model will provide insights for tourism strategists to tailor services and optimize resources. Evaluation is based on the F1-score, focusing on accurate spending tier classification.

Contributors

- Member 1: **Bhavya Kapadia (IMT2022095)**
- Member 2: **Shreyas Biradar (IMT2022529)**
- Member 3: **Siddeshwar Kagatikar (IMT2022026)**

0.1 Approaches Explored

Shreyas Biradar's Approaches and Findings

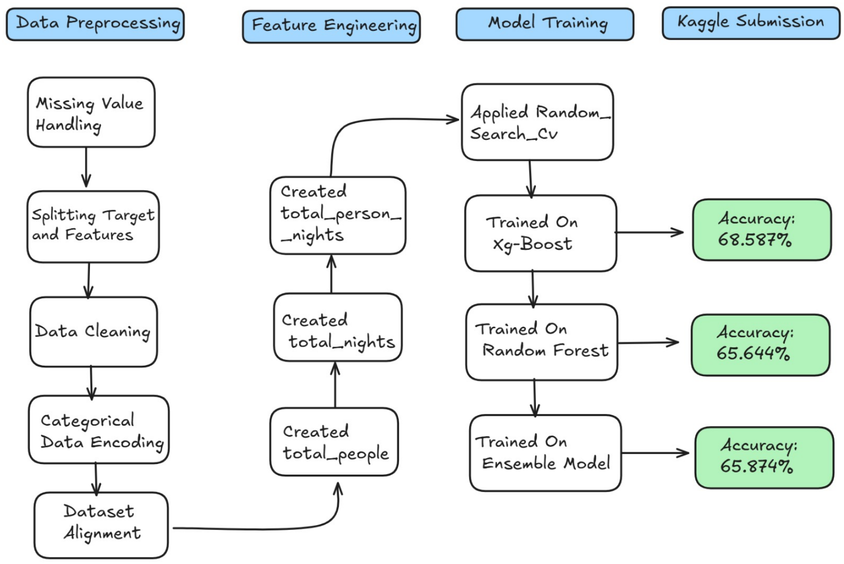

Figure 1: Workflow by Shreyas Biradar

Data Preprocessing

The **dataset was cleaned** by removing rows with missing values in the **target column (category)** using `dropna()`. The **target variable (category)** was separated, and **unused columns (female count, male count)** were dropped after creating combined features. **Column names were cleaned** to remove special characters for **XGBoost compatibility**. Categorical columns with low cardinality were **one-hot encoded** to convert them into numerical format. Finally, the train, validation, and test datasets were aligned, ensuring consistency by **filling missing columns with zeros** where necessary.

Feature Engineering

New features were created by summing the **'female count'** and **'male count'** to generate **'total people'**, and adding **'mainland nights'** and **'island nights'** to create **'total nights'**. Additionally, **'total person nights'** was calculated as the **product** of **'total people'** and **'total nights'**.

Model Performance and Analysis

The dataset was split using **stratified sampling**, and **XGBoost** and **Random Forest** models were trained. **Hyperparameter tuning** for XGBoost was done with **RandomizedSearchCV**. The best XGBoost model and a default Random Forest were used for predictions, which were combined via averaging. Models were evaluated with **MAE**, and predictions were categorized and saved into **CSV files** for submission.

Bhavya Kapadia's Approach and Findings

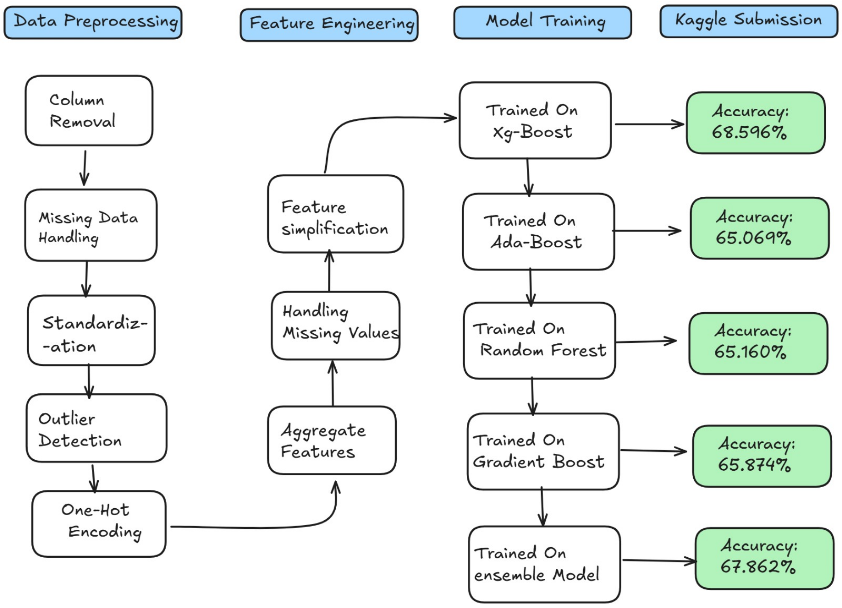

Figure 2: Workflow by Bhavya Kapadia

Data Preprocessing

Several columns were dropped for **irrelevance** or **redundancy**, including `trip_ID`, `travelling_with`, `trip_purpose`, `first_time_visitor`, `source_of_info`, `weather_at_arrival`, and others. Rows with **missing values** in critical columns (`transport_package_international`, `package_accomodation`, `food_package`, `insurance_package`) were removed to ensure **data integrity**. Missing values in `days_before_booked` were imputed with random **dominant values** (`"61-90"` or `"90+"`), and for `tour_length`, values `"7-14"` or `"30+"` were used. Finally, **categorical variables** were transformed using **one-hot encoding** for **machine learning compatibility**.

Feature Engineering

New features were created to capture meaningful patterns, including **total count** (sum of `female_count` and `male_count`) and **total nights** (sum of `mainland_nights` and `island_nights`). Categorical columns with **low cardinality** were simplified or grouped to reduce sparsity and enhance model generalization. Additionally, `widlife` in `key_activity` was renamed to `wildlife`, and countries with less participation was converted to `others`.

Model Performance and Analysis

Various models were trained to compare performance: **Random Forest** (accuracy **0.65160**), **Adaboost** (**0.65069**), and **Gradient Boosting** (initial **0.65305**, improved to **0.65925** after tuning). The **XGBoost model** started with an accuracy of **0.65318**, improved to **0.65361**, and after extensive **hyperparameter optimization**, achieved **0.68409**. An **ensemble model** combining multiple classifiers reached an accuracy of **0.67862**. The optimized XGBoost model outperformed all others with the highest accuracy of **0.68596**, making it the **final choice**.

Siddeshwar Kagatikar's Approaches and Findings

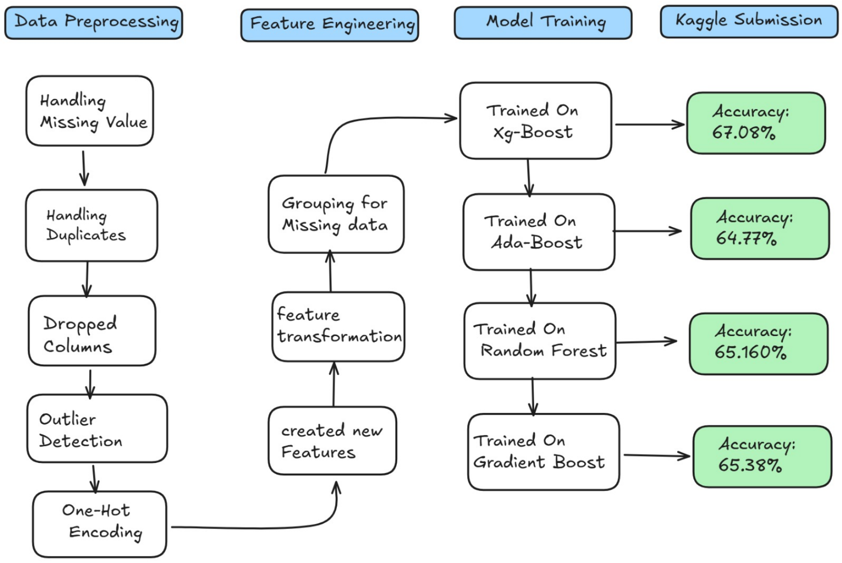

Figure 3: Workflow by Siddeshwar Kagatikar

Data Preprocessing

Missing values in key columns like **special requirements**, **age bracket**, **first-time visitor**, and **key activity** were imputed using default values or the **mode**. Columns such as **travelling with**, **visitor nation**, and **days before booked** were filled with default strings like "Not Specified" and "Unknown". Missing weather data at arrival was imputed with the **mode value** within each **visitor nation group**. Duplicate entries were removed to maintain data consistency, and irrelevant columns like **trip ID**, **female count**, and **male count** were dropped. **Outliers** in numerical columns were handled using the **IQR method** to prevent skewed data.

Feature Engineering

New features were created, including **total group size** (sum of `female count` and `male count`) and **total nights** (sum of `mainland` and `island nights`). **Categorical features** like `special requirements`, `age bracket`, `first-time visitor`, and `key activity` were transformed by filling missing values with **default values** or the most frequent category. Missing weather data at arrival was imputed based on the **mode** within each `visitor nation group` to improve accuracy.

Model Performance and Analysis

Two models were evaluated: **Random Forest Classifier**, which achieved **65.54%** accuracy, and **XGBoost Classifier**, which achieved **67.08%**. The **AdaBoost** and **Gradient Boosting** models achieved similar performance levels, with accuracies of **64.77%** and **65.38%**, respectively. **XGBoost** outperformed **Random Forest**, making it the better choice for the task. The **data preprocessing phase** focused on cleaning and handling missing data, while **feature engineering** involved creating and transforming features. The **experimental phase** compared models to identify the one with the highest accuracy.

Team ByteMe's Approaches and Findings

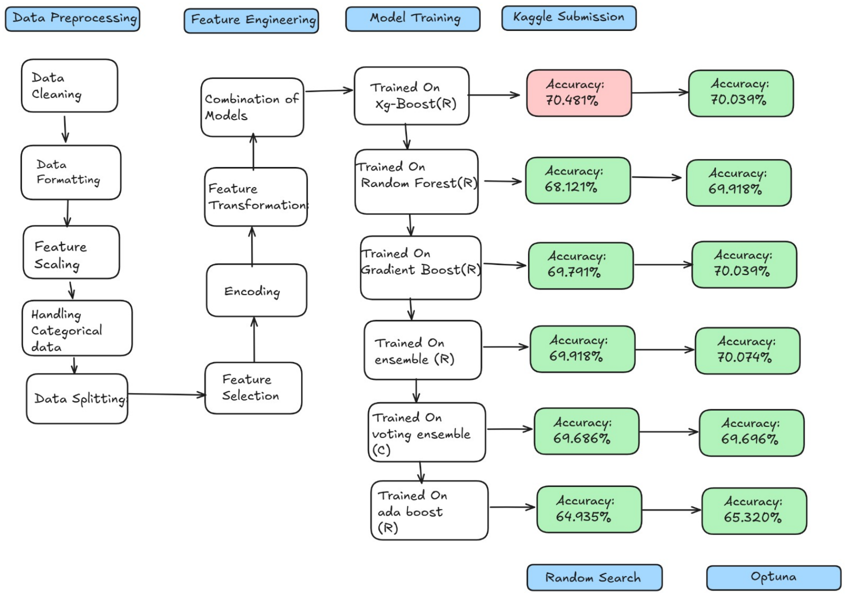

Figure 4: Final WorkFlow by Team Byte Me

Data Preprocessing

During the **data preprocessing phase**, we addressed missing values using techniques such as **imputation** (mean or median), removal, and forward/backward filling for time series data. To ensure consistency, we cleaned the dataset by eliminating unwanted characters. **Feature scaling** was a key step, with **standardization** (mean=0, std=1) applied to normalize feature ranges, which is especially important for models sensitive to distance metrics. **Categorical variables** were handled through **label encoding** for binary classes and **one-hot encoding** for multiclass categories, allowing models to interpret the data effectively. To ensure a fair evaluation, we split the dataset into **training, validation, and test sets**, often using **stratified sampling** to maintain class balance. Using this comprehensive preprocessing strategy, we finalized our work, ensuring our data was ready for robust and accurate model performance.

Feature Engineering

Feature selection helps identify important attributes using domain knowledge or metrics to enhance **model efficiency**. **Categorical variables** are encoded (e.g., `one-hot encoding` or `label encoding`) to prepare them for analysis. **Feature scaling** or **normalization** is applied to improve **model performance** and ensure consistency. Additionally, we created new columns to map **categorical ranges** to numerical values for `days_booked_before` and `tour_length`. We also **converted numerical columns to categorical**, which led to a **boost in accuracy**. Finally, **ensemble techniques**, such as `XGBoost`, `Random Forest`, and `GradientBoosting`, combine model predictions using **weighted averaging** to provide robust and **accurate results**.

Model Performance and Analysis

Models like **XGBoost**, **Random Forest**, and **Gradient Boosting** undergo **hyper-parameter tuning** using techniques like **RandomizedSearchCV** and **Optuna** to enhance generalization and reduce overfitting. The best parameters are used to train these models, and their performance is evaluated using metrics like **Mean Absolute Error (MAE)**. An **ensemble approach** combines predictions from the models using weighted averages for improved accuracy. The final ensemble predicts test set outcomes, categorizes them into classes (0, 1, 2) based on thresholds, and saves the results to a **CSV file**.

0.2 Additional Insights

The regressor models produced results closely matching those of the classifiers, likely due to the nature of the target variable, which may be clustered around discrete points, allowing regression to approximate classification boundaries effectively. This similarity also reflects the strength and consistency of patterns in the dataset, as well as the overlap in evaluation metrics, where both types of models perform similarly on structured data. For the best-performing model, **XGBoost**, an extensive hyperparameter tuning process was conducted using a grid search. The parametric grid included a range of values for key parameters, leading to the selection of optimal settings:

- **Learning Rate** = 0.05
- **N Estimators** = 200
- **Max Depth** = 4
- **Subsample** = 1.0
- **Colsample ByTree** = 0.7
- **Gamma** = 0.1

These settings significantly contributed to the model's predictive performance and its ability to generalize effectively to unseen data.

0.3 Conclusion

The report summarizes collective findings from the three approaches, discussing areas for potential improvement based on the strengths and weaknesses observed. Future work may include exploring hybrid models or incorporating additional features to improve predictive power.

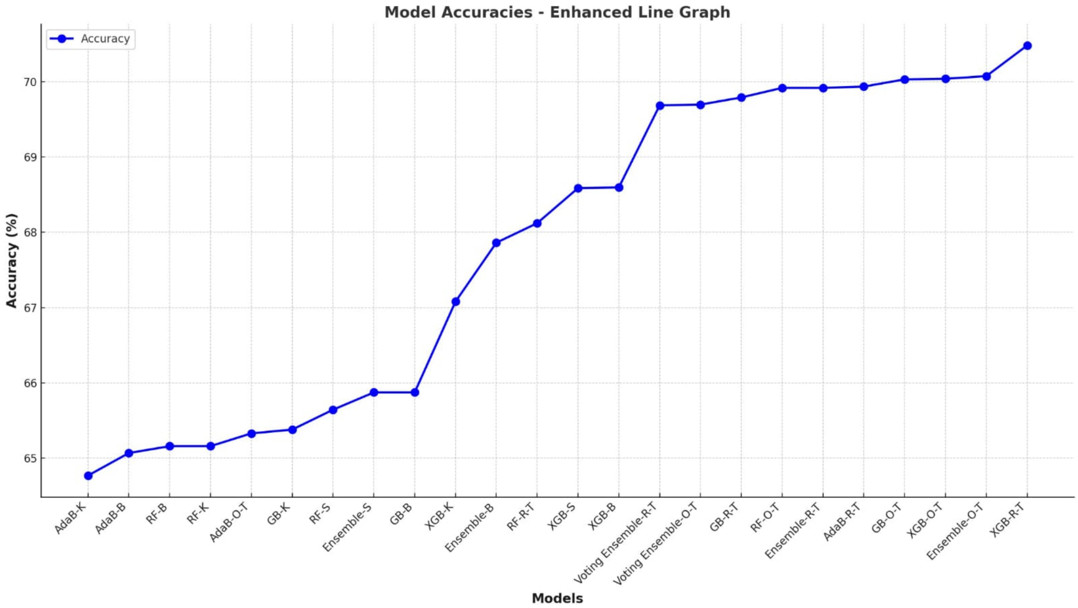

Figure 5: Model accuracies for different algorithms shown in an enhanced line graph, highlighting the incremental improvement in accuracy across models.

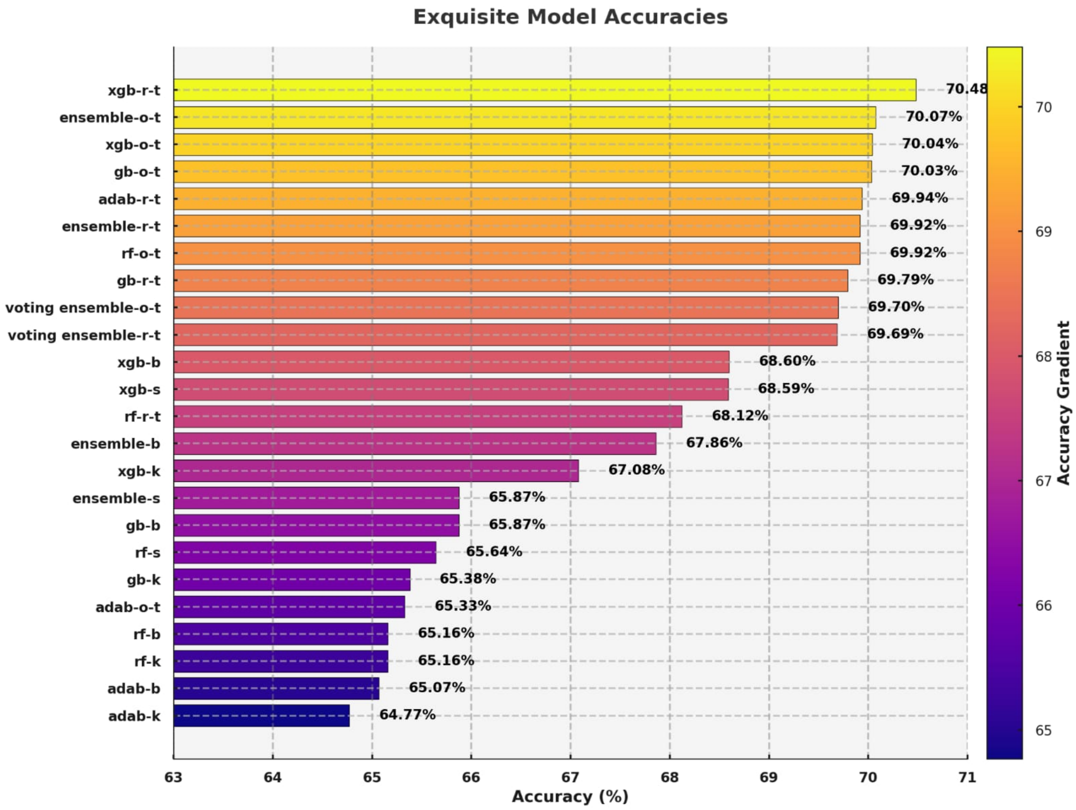

Figure 6: Model accuracies for different algorithms shown in an enhanced line graph, highlighting the incremental improvement in accuracy across models.