Reducing Uncertainty in Hotel Reservations Using AI-Based Cancellation Classification

A Comparative Study of Machine Learning Classifiers

Mohamed Shawky
Misr International University
mohamed2310167@miuegypt.edu.eg

Mohamed Hesham
Misr International University
mohamed2300428@miuegypt.edu.eg

Nourhan Khaled
Misr International University
nourhan2304880@miuegypt.edu.eg

Veronia Sameh
Misr International University
veronia2300592@miuegypt.edu.eg

AbdelKader Adnan
Misr International University
abdulkader2307019@miuegypt.edu.eg

***Abstract*—This research presents an artificial intelligence–based classification approach for predicting hotel booking cancellations, a critical problem that directly impacts revenue management and operational efficiency in the hospitality industry. Using a dataset of customer, reservation, and hotel-related features, a complete machine learning pipeline is implemented, including data preprocessing, exploratory data analysis (EDA), feature engineering, and supervised model training. To enhance predictive performance and reduce feature redundancy, a Genetic Algorithm–based feature selection method identifies the most informative features. Multiple classifiers, including K-Nearest Neighbors, Decision Tree, and Multi-Layer Perceptron, are trained and validated. Model performance is evaluated using accuracy, precision, recall, F1-score, and confusion matrices. The results demonstrate that evolutionary feature selection improves classification effectiveness, providing reliable booking cancellation predictions and supporting data-driven business decisions in hotel management.**

***Index Terms*—Hotel booking cancellation, machine learning, classification, feature selection, genetic algorithm, revenue management, predictive analytics**

I. Introduction

The growing importance of managing hotel reservations and cancellations has made it a critical task for organizations to optimize revenue and operational efficiency. In this research, we investigate the development of a predictive model aimed at forecasting hotel booking cancellations, utilizing a comprehensive dataset containing customer, reservation, and hotel-related features. Our study covers several crucial stages in the model development process, including data preprocessing, exploratory data analysis (EDA), feature engineering, model training, hyperparameter tuning, and performance evaluation using multiple classification metrics.

To enhance clarity regarding our methodology, we provide a detailed, step-by-step outline of the essential phases involved in constructing the booking cancellation prediction model. Each phase is described with in-depth explanations and practical considerations to promote a thorough understanding. Special emphasis is placed on data preprocessing and EDA to extract meaningful insights, followed by feature engineering and Genetic Algorithm–based feature selection strategies to improve predictive performance. The primary objective of this work is to provide readers with a structured and practical framework for applying machine learning techniques to hotel booking cancellation prediction, supporting data-driven business decisions and revenue management in the hospitality industry.

II. DATA PREPARATION FOR MODEL TRAINING

The experimental analysis in this study utilizes the *hotel_bookings.csv* dataset, which contains comprehensive information about hotel reservations, including customer demographics, reservation details, and hotel characteristics. This dataset was loaded and processed using Python's pandas library. Key predictive features include `hotel` (hotel type), `lead_time` (days between booking and arrival), `stays_in_weekend_nights`, `adults`, `children`, `previous_cancellations`, and `adr` (average daily rate). These attributes collectively provide insights into booking patterns, customer behavior, and risk factors associated with cancellations.

A. Data Preprocessing

The dataset underwent thorough preprocessing to ensure effective model training. Missing values in columns such as `required_car_parking_spaces` and `agent` were imputed using statistical measures (mean for numerical, mode for categorical) or domain-specific rules. Categorical variables, including `reservation_status` and `customer_type`, were encoded using one-hot encoding as well as label encoding for nominal features and label encoding for ordinal categories where applicable. Numerical features were standardized using z-score standardization to ensure consistent scaling across different measurement units. Duplicate entries were identified and removed to prevent data leakage, and date features such as `arrival_date` were transformed into usable temporal components to seconds for seasonal pattern analysis.

B. Exploratory Data Analysis (EDA)

Exploratory Data Analysis was conducted to investigate feature relationships and their influence on booking cancellations. EDA is a necessary step prior to preprocessing as it provides a comprehensive understanding of the raw dataset, highlighting anomalies, correlations, and trends that guide appropriate data cleaning and transformation.

Correlation analysis revealed that lead_time and adr exhibit moderate positive correlations with the target variable is_canceled, while features such as required_car_parking_spaces and total_of_special_requests show negative correlations, suggesting that guests with specific requests or parking needs are less likely to cancel.

A **heatmap** was employed to visualize these relationships, highlighting that longer lead times and higher ADRs are associated with increased cancellation likelihood. **Box plots** further confirmed that canceled bookings tend to have significantly higher lead times and ADRs compared to non-canceled ones, indicating that early and expensive bookings are more prone to cancellation.

Bar charts analyzing customer_type and market_segment revealed that transient customers and online travel agency bookings dominate both canceled and non-canceled categories, with transient customers showing the highest cancellation rates. Additionally, cancellation patterns varied across hotel types, with some hotel categories experiencing disproportionately higher cancellations.

Temporal trends were examined using **line plots** across months and week numbers. Booking volumes peaked in January and October, with noticeable dips in May and December, suggesting seasonal influences. Weekly booking trends showed fluctuations throughout the year, with a sharp spike at week 0, possibly reflecting year-end booking behavior.

Geographic distribution analysis indicated a relatively uniform booking count across cities.

Finally, the relationship between stays_in_week_nights and stays_in_weekend_nights was explored, revealing that longer weekday stays are often accompanied by weekend stays, which correlates with lower cancellation rates. This implies that extended stays may reflect more committed travel plans.

Overall, the EDA provided critical insights into customer behavior, booking patterns, and cancellation drivers, forming a robust foundation for predictive modeling and strategic decision-making.

C. Feature Engineering

The dataset was enhanced with derived features to improve model discriminative power. total_stay was computed as the sum of stays_in_weekend_nights and stays_in_week_nights to capture overall stay duration. Additional features including booking frequency indicators and cancellation history ratios were created by analyzing booking_changes and previous_cancellations to quantify customer booking behavior patterns. Seasonal indicators were extracted from date features to capture holiday and peak-season effects on cancellation likelihood.

This comprehensive approach to data preparation, encompassing preprocessing, feature engineering, and exploratory analysis, ensures the dataset is optimally structured for machine learning algorithms. By systematically transforming and enriching the raw booking data, the predictive models can more effectively capture complex relationships between booking attributes and cancellation outcomes, leading to more accurate and reliable predictions.

D. Data Splitting

To evaluate the model fairly, the dataset was divided into training, validation, and test sets. A stratified split was used so that the class distribution remained consistent across all subsets. The split resulted in 70% training, 15% validation, and 15% test data. It is important to split before oversampling to avoid information leakage. If synthetic samples were added before splitting, they could appear in both training and test sets, leading to misleading results. By applying oversampling only to the training set, the model learns from balanced data while the validation and test sets remain representative of real-world conditions.

E. Data Standardization

Prior to model training, numerical features were standardized to ensure that all variables contributed equally to the learning process.

Normalization was not used because tournament selection depends only on the relative fitness ranking, not on absolute fitness values. Since the F1-score is already bounded between 0 and 1, normalization would not affect the selection process or improve performance

To prevent data leakage, the scaler was fit exclusively on the training set. The learned scaling parameters (mean and standard deviation) were then applied to the validation and test sets. This guarantees that information from unseen data does not influence the training process, thereby preserving the integrity of model evaluation.

In total, all numerical features were standardized across the training, validation, and test sets, ensuring consistency and comparability during model development.

III. Genetic Algorithm-Based Feature Selection

Before training the predictive models, a Genetic Algorithm (GA) was implemented to identify the most informative subset of features. The GA is a population-based metaheuristic inspired by natural selection, where candidate solutions evolve over successive generations to maximize a fitness function.

A. Chromosome Representation

Each chromosome was encoded as a binary vector of length equal to the total number of features. A value of 1 indicates that the corresponding feature is included, while 0 indicates exclusion. This representation allows the GA to search across possible feature subsets.

B. Initialization

The initial population consisted of 20 randomly generated chromosomes (`POP_SIZE = 20`). This ensured diversity in the search space at the start of the algorithm.

C. Fitness Function

The fitness of each chromosome was evaluated using the F1-score of a Decision Tree classifier. The Decision Tree was depth-limited (`max_depth = 6`) to reduce computational cost and prevent overfitting during the GA search. Chromosomes that resulted in an empty feature subset were penalized with a fitness score of zero.

D. Selection

Tournament selection was applied with a tournament size of 3. In this process, three chromosomes were randomly sampled from the population, their fitness scores were compared, and the best-performing chromosome was selected as a parent.

E. Crossover

Single-point crossover was used to generate offspring. A random crossover point was chosen, and two parent chromosomes exchanged segments to produce two children.

F. Mutation

Mutation was applied with a probability of 0.2 per gene (`MUTATION_RATE = 0.2`). For each feature position, the bit was flipped (0 to 1 or 1 to 0) with this probability.

G. Generational Loop

The GA was executed for three generations (`N_GENERATIONS = 20`). In each generation, a new population was created by repeatedly selecting parents, applying crossover, and mutating offspring. After reproduction, the fitness of all chromosomes was evaluated, and the best-performing chromosome was tracked.

H. Best Solution

At the end of the evolutionary process, the chromosome with the highest F1-score was selected as the optimal feature subset. This subset was then used in subsequent model training and evaluation.

IV. Data Balancing

Imbalanced class distributions can bias models toward the majority class, reducing their ability to detect cancellations. To address this, we evaluated the target distribution and applied balancing techniques as described below.

A. Checking Data Balance

To ensure the model is not biased toward the majority class, we analyzed the distribution of the target variable. The dataset was found to be imbalanced, which can negatively affect model performance. To address this, we applied SMOTE (Synthetic Minority Oversampling Technique), a method that generates synthetic samples for the minority class to achieve a balanced dataset. Kindly refer to the figure below to observe the distribution of the target variable before applying SMOTE.

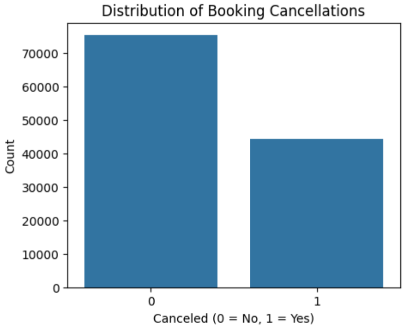

Fig. 1. Distribution of the target variable before and after applying SMOTE for data balancing.

B. Data Balancing with SMOTE

Following feature selection, the training set was balanced using the Synthetic Minority Oversampling Technique (SMOTE). This ordering ensured that oversampling was performed only on the most relevant features, improving the efficiency of the algorithm and reducing the risk of generating synthetic samples in uninformative dimensions.

A sampling strategy of 0.7 was applied, meaning the minority class (canceled bookings) was increased until it reached 70% of the majority class size. Before balancing, the training set contained 52,616 non-canceled bookings (62.96%) and 30,957 canceled bookings (37.04%). After applying SMOTE, the distribution shifted to 52,616 non-canceled (58.82%) and 36,831 canceled (41.18%), resulting in a more balanced dataset for model training.

By applying SMOTE after feature selection, the models were trained on a dataset that was both dimensionally optimized and class-balanced, enhancing their ability to detect cancellations while maintaining generalization.

V. Model Training

A. Model Evaluation Using GA-Selected Features

To assess model performance, three classification algorithms were trained using features selected by a Genetic Algorithm (GA): K-Nearest Neighbors (KNN), Decision Tree, and a Neural Network (MLP). Each model was trained on the resampled training set and evaluated on the validation set.

Training and validation errors were computed using accuracy scores, where error is defined as $1 - \text{accuracy}$. This approach provides a straightforward measure of each model's ability to generalize beyond the training data.

The results showed that KNN achieved the lowest training error (0.1368) but had the highest validation error (0.2125), indicating potential overfitting. The Decision Tree and Neural Network models demonstrated more balanced performance, with validation errors of 0.1703 and 0.1728 respectively. These findings suggest that while KNN fits the training data well, the Decision Tree and MLP models may generalize better to unseen data.

B. Performance Evaluation

To assess the effectiveness of each model, performance metrics were computed on the validation set using GA-selected features. The models evaluated include K-Nearest Neighbors (KNN), Decision Tree, and Neural Network (MLP). For each model, predictions were generated and compared against true labels to calculate standard classification metrics: accuracy (overall correctness of predictions), precision (proportion of predicted cancellations that were correct), recall (proportion of actual cancellations correctly identified), and F1-score (harmonic mean of precision and recall).

Confusion matrices were also constructed to visualize the distribution of true positives, true negatives, false positives, and false negatives. These matrices provide deeper insight into each model's classification behavior, particularly in handling imbalanced data.

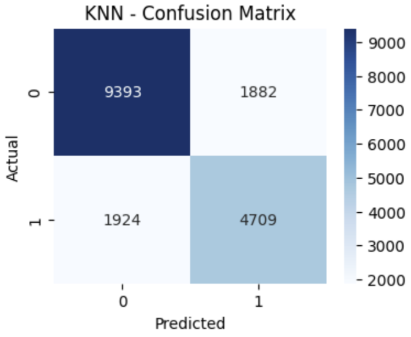

Fig. 2. Confusion Matrix for KNN Model

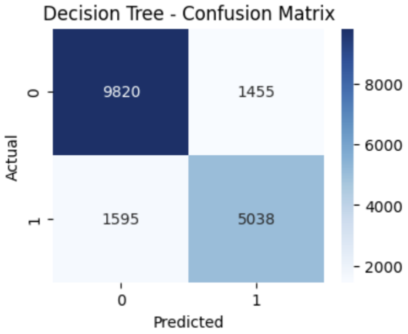

Fig. 3. Confusion Matrix for Decision Tree Model

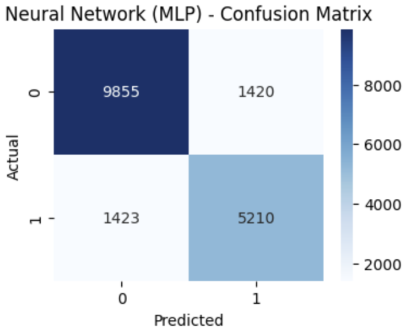

Fig. 4. Confusion Matrix for Decision Tree Model

C. Model Performance Summary

comparison of three classifiers shows that the Neural Network (MLP) achieved the highest test accuracy (84.38%) and F1 score (78.87%), indicating strong overall performance. The Decision Tree obtained the highest precision (82.86%), while KNN recorded the lowest values across all metrics. These results highlight the effectiveness of neural networks in predicting hotel booking cancellations.