Predictive Modeling for Hotel Booking Cancellations

Ali El Masri

# Abstract

Having worked in the hospitality industry for the past decade, I've developed a keen intuition for identifying a significant number of daily arrivals likely to result in no-shows. However, predicting potential cancellations exceeds my cognitive capacity. This paper offers a thorough analysis and model development for forecasting hotel booking cancellations. Utilizing a dataset of hotel bookings, we applied various machine learning models, including logistic regression, random forests, and neural networks, to understand and predict guest cancellation behavior. This study aims to provide actionable insights to hotel management to reduce cancellations and optimize occupancy rates.

# Introduction

The hospitality industry faces significant challenges with booking cancellations, leading to lost revenue and inefficiencies in resource allocation. Predictive analytics can offer a solution by providing early warnings about potential cancellations, allowing hotels to take proactive measures.

Since hotels are constantly seeking ways to maximize occupancy rates and minimize cancellations predicting booking cancellations can be a game-changer, by boosting a hotel's operational efficiency, revenue management, and customer satisfaction This paper outlines the methodologies, experiments, and findings of our predictive modeling efforts.

# Data Preprocessing

The dataset used for this study contains 119,390 entries with 32 features related to hotel bookings. Key preprocessing steps included:

1. **Handling Missing Values**: Missing values in critical columns such as children, company, and agent were filled with zeroes. Missing values for the country were filled with the mode.
2. **Encoding Categorical Variables**: Label encoding was applied to convert categorical variables into numerical formats suitable for machine learning algorithms.
3. **Feature Selection and Engineering**: Features like “reservation\_status” and “reservation\_status\_date”, which could introduce data leakage, were excluded from the model training process. Additionally, two highly correlated columns were dropped to avoid multicollinearity. A new column was created by summing the number of children and adults to represent the total number of guests.
4. **Normalization**: Numerical columns were scaled using the StandardScaler method from sklearn.
5. **One-Hot Encoding**: The encoded categorical variables were one-hot encoded to avoid introducing numerical bias into the model, as they do not have a numerical relationship

# Experimentation with Class Imbalance

In this project, we aimed to develop our model using a balanced and representative dataset. Initially, the dataset exhibited a class imbalance, with a significantly higher number of non-canceled bookings compared to canceled ones. This imbalance could lead to biased model predictions, favoring the majority class and resulting in poor performance on the minority class.

To address this issue, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to our training and validation sets. SMOTE generates synthetic samples for the minority class by interpolating between existing samples, effectively balancing the class distribution without introducing significant bias. After applying SMOTE, the balanced dataset was split into training, validation, and testing sets, with 20% reserved for incremental learning. This approach ensures that our model is trained on a balanced dataset, promoting fairness and improving its ability to generalize to both majority and minority classes.

We experimented with different methods to handle class imbalance, a weighted class approach. However, this approach slightly worsened the model performance compared to SMOTE.

Figure 1: Class Distribution in Training Set After SMOTE

A graph of a number of blue bars

Description automatically generated

Figure 2: Class Distribution in Test Set without SMOTE

A bar graph with blue rectangles

Description automatically generated

# Exploratory Data Analysis (EDA)

Our EDA highlighted several important patterns:

* High correlation between lead time and cancellation rates.
* Significant differences in cancellation behavior between different market segments and deposit types.
* High occurrence of cancellations in specific months, indicating seasonal trends.

# Model Development and Experimentation

We implemented and compared several machine learning models and one rule-based model based on deposit type, as deposit type was identified as the most important feature by our decision tree model and had a high positive correlation of 0.47. However, the rule-based model had high precision but very low recall, indicating it predicts cancellations accurately but misses many actual cancellations. The machine learning models showed more balanced performance with good precision and recall, leading to higher F1 scores.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| Neural Network | 0.883 | 0.8385 | 0.8474 | **0.8429** |
| Random Forest | **0.8841** | 0.8575 | 0.8224 | 0.8396 |
| Weighted Neural Network | 0.872 | 0.8098 | 0.8553 | 0.8319 |
| Rule Based Model | 0.7495 | **0.9852** | 0.3286 | 0.4928 |
| Logistic Regression | 0.8329 | 0.7473 | 0.8265 | 0.7849 |
| Decision Tree | 0.8626 | 0.81 | 0.82 | 0.81 |
| SVM | 0.8675 | 0.7962 | **0.8612** | 0.8274 |
| KNN | 0.7854 | 0.6613 | 0.8572 | 0.7466 |
| GBM | 0.8529 | 0.7945 | 0.8108 | 0.8026 |

**Hyperparameter Tuning**

Due to computational limitations, some models like SVM, KNN, and GBM were trained using default hyperparameters. Below are the best parameters found for other models using Gridsearch:

* **Decision Tree:** {'criterion': 'entropy', 'max\_depth': 20, 'min\_samples\_leaf': 2, 'min\_samples\_split': 10}
* **Random Forest:** {'max\_depth': 30, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200}
* **Neural Network:**
  + Drop Rate: 0.3
  + Learning Rate: 0.001
  + Batch Size: 64
  + Total number of parameters: 1,692,673
  + Total number of trainable parameters: 1,692,673
  + (layers): ModuleList(

    (0): Linear(in\_features=1011, out\_features=1024, bias=True)

    (1): LeakyReLU(negative\_slope=0.01)

    (2): Dropout(p=0.3, inplace=False)

    (3): Linear(in\_features=1024, out\_features=512, bias=True)

    (4): LeakyReLU(negative\_slope=0.01)

    (5): Dropout(p=0.3, inplace=False)

    (6): Linear(in\_features=512, out\_features=256, bias=True)

    (7): LeakyReLU(negative\_slope=0.01)

    (8): Dropout(p=0.3, inplace=False)

  )

  (output): Linear(in\_features=256, out\_features=1, bias=True)

)

Criterion: BCELossOptimizer: optim.Adam(lr=learning\_rate, weight\_decay=1e-5)Scheduler: CyclicLR(optimizer, base\_lr=1e-5, max\_lr=1e-3, step\_size\_up=2000, mode='triangular2')

# Model Performance

Figure 3: Training Process of Neural Network Model

A graph of a curve

Description automatically generated with medium confidence

The graphs in Figure 3 depict the training process of the neural network model, showcasing the changes in model accuracy and loss over epochs. The accuracy graph illustrates a steady improvement in validation accuracy as the number of training epochs increases, eventually stabilizing around 0.94. Noticeable drops in accuracy at specific points are caused by the use of cyclic learning rates, where the learning rate is periodically adjusted to help the model escape local minima and achieve better convergence.

Similarly, the loss graph demonstrates a consistent decrease in training loss over epochs, further confirming that the model is refining its predictions. The periodic spikes in loss followed by rapid decreases align with the adjustments in learning rates, supporting the model's convergence to an optimal solution. Together, these graphs illustrate the model's effective learning process.

Figure 4: Neural Network ROC Curve

A graph with a line

Description automatically generated

Figure 5: Precision-Recall Curve

A line graph with numbers

Description automatically generated

Figure 6: Confusion Matrices

A blue and white squares

Description automatically generated

The neural network model demonstrates strong performance across multiple evaluation metrics. The ROC curve's AUC of 0.88, the balanced precision-recall curve, and the favorable confusion matrix metrics indicate that the model is highly effective at predicting hotel booking cancellations. The model maintains high precision and recall, ensuring accurate identification of cancellations with minimal false positives and false negatives.

# Conclusion

Predictive modeling offers a valuable tool for the hospitality industry to anticipate and manage booking cancellations. Our study demonstrates that the neural network model provides the best balance between precision and recall, making it one of the most suitable for predicting cancellations in our dataset. Early detection of potential cancellations can be significantly enhanced using machine learning, providing hotels with actionable insights to mitigate revenue losses. The neural network model outperformed other models such as logistic regression and random forests, highlighting its superior performance and potential for real-world applications in hotel management systems.

Future work will focus on further tuning the neural network model to enhance its predictive capabilities and incorporating cyclical feature encoding. Additionally, we plan to experiment with incremental learning techniques, which could provide continuous improvements in model performance over time.

Appendix

* **GitHub Link for Additional Visualizations and Analysis Code:** [GitHub Repository](https://github.com/elmasri-ali)
* **Data Source:** [Kaggle Hotel Booking Demand Dataset](https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand)
* **Main Models Libraries:** IMBlearn, SKlearn, Pytorch (using CUDA for GPU utilization)
* **Additional Visualizations:**

Figure 7: Comparison of Recall Across Models

A graph of a number of rectangular objects

Description automatically generated with medium confidence

Figure 8: Comparison of Precision Across Models

A graph of different sizes of bars

Description automatically generated

Figure 9: Comparison of F1 Score Across Models

A graph of different sizes of objects

Description automatically generated with medium confidence

Figure 10: Comparison of Accuracy Across Models

A graph of a graph of accuracy

Description automatically generated with medium confidence

Figure 11: Feature Importance

A graph with blue and white bars

Description automatically generated

Figure 12: Final Model Diagram

A diagram of a flowchart

Description automatically generated

References

Herrera, A., Arroyo, Á., Jiménez, A., & Herrero, Á. (2024). Forecasting hotel cancellations through machine learning. *Expert Systems*. https://doi.org/10.1111/exsy.13608

Gift, U. (2023). *Predicting hotel cancellations using Machine Learning: A data science journey*. Medium. https://medium.com/@ukpowehonome/predicting-hotel-cancellations-using-machine-learning-a-data-science-journey-43c27766ef07

10 Questions an Audience Might Ask

1. How does the lead time affect the cancellation rate?
   * Lead time has a high correlation with cancellation rates, with longer lead times generally associated with higher cancellation rates.
2. What features had the most significant impact on the prediction model?
   * Deposit type, lead time, and market segment were among the most significant features impacting the prediction model.
3. Why did the weighted class approach perform worse than SMOTE?
   * The weighted class approach slightly worsened performance due to potential overfitting or inadequate balancing compared to SMOTE.
4. How can hotels use these predictions to improve their operational efficiency?
   * Hotels can use these predictions to implement proactive measures, such as overbooking strategies and personalized customer engagement, to reduce cancellations.
5. What challenges did you face in preprocessing the data?
   * Handling missing values, encoding categorical variables, and avoiding data leakage were some of the primary challenges faced during preprocessing.
6. How do seasonal trends affect booking cancellations?
   * Seasonal trends showed higher cancellation rates during specific months, indicating the importance of incorporating seasonality into predictive models. Which we will be experimenting with in our future work.
7. What steps can hotels take to mitigate high cancellation rates?
   * First, it can help us strategies for overbooking, secondly, implementing stricter cancellation policies, offering incentives for confirmed bookings, and using predictive models to identify at-risk bookings can help mitigate high cancellation rates.
8. How can this model be integrated into existing hotel management systems?
   * The model can be integrated through API endpoints or directly into the hotel’s booking management software to provide real-time cancellation risk assessments.
9. What are the limitations of the current model?
   * Limitations include potential overfitting, reliance on historical data, and the need for regular updates to maintain model accuracy.
10. How would you improve the model in future work?
    * Future improvements include further tuning of the neural network model, incorporating additional data sources, cyclical feature engineering and experimenting with incremental learning techniques.