Hotel Reservations

Predictive Analytics for Reservation Cancellation

Student Name
David Wassef
Adam Marshall
Allen Li
Varshini Rechintala

Course ID: Machine Learning 2 - BIA 5402 0LA
Professor: Lubna Mohammad

As part of the hospitality industry, hotels and similar establishments are dependent upon regular bookings to continue healthy operations. One of the main hurdles to this healthy operation is customers canceling established reservations, sometimes leaving little time for the hotel to find new reservations to fill that space and recoup any potential losses in revenue. To assist hotels and similar establishments we have decided to investigate this phenomenon to determine what, if any, reasonable predictors there are that a customer is likely to cancel their reservation. Having access to this information would allow hotels to be better equipped in determining which reservations are most likely to be canceled, in turn giving more time for hotels to fill potentially vacant rooms with new reservations.

In conducting our analysis, we have taken data from 4 years of hotel reservations to determine which features of the dataset have the greatest impact in reservation cancellation (Aboelwafa, 2022). Below are listed the complete set of variables that our dataset keeps track of and an explanation of what that variable means:

- **Booking ID**
  - Unique identifier for each booking
- **Number of adults**
  - Number of adults included in the booking
- **Number of children**
  - Number of children included in the booking
- **Number of Weekend Nights**
  - Number of weekend nights in the booking
- **Number of Weeknights**
  - Number of weekday nights in the booking
- **Type of Meal**
  - Type of meal plan included in the booking
- **Car Parking Space**
  - Indicates whether a parking space was requested in the booking
- **Room Type**
  - Type of room booked
- **Lead Time**
  - Number of days between the booking date and arrival date

- **Market Segment Type**
    - Type of market segment associated with the booking
- **Repeated**
    - Indicates whether the booking is made by a repeat customer
- **P-C**
    - Number of previous bookings that were canceled prior to the current booking
- **P-not-C**
    - Number of bookings that were not canceled prior to the current booking
- **Average Price**
    - Average price of booking on per night basis
- **Special Request**
    - Number of special requests made by the guest
- **Date of Reservation**
    - Date of the reservation
- **Booking Status**
    - Status of the booking (canceled or not canceled)

For our analysis, the important output variable we want to focus on will be Booking Status, as this is the variable we want to predict for our clients. The remaining metrics, excepting Booking ID as it will be irrelevant, will be used as our input variables since we want to use the complex relationships among these variables to determine Booking Status.

Before getting into the predictive analytics of this project, we wanted to point out some interesting aspects of our data that we discovered as part of the pre-analysis process. The below figures show the frequency distribution of Average Price and Lead Time. Of note in these figures we see that Average Price (Fig.1) is normally distributed, and as such we do not expect average price to skew our model. Meanwhile, Lead time (Fig.2) is right skewed, indicating that many of the customers who placed reservations did so with little lead time before arriving for the reservation itself. Fig. 3, also shown below, provides a correlation analysis among all important variables, with further details.

Figure 1. Average price that a customer has paid for each day

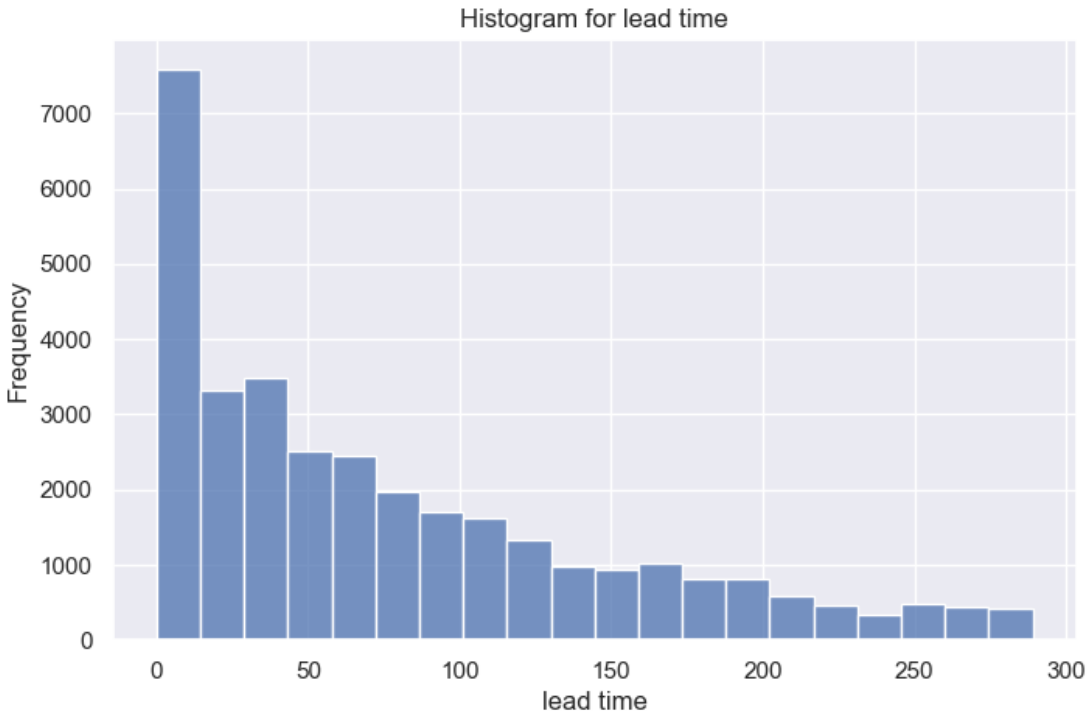

Figure 2. The frequency of lead time for each booking entered

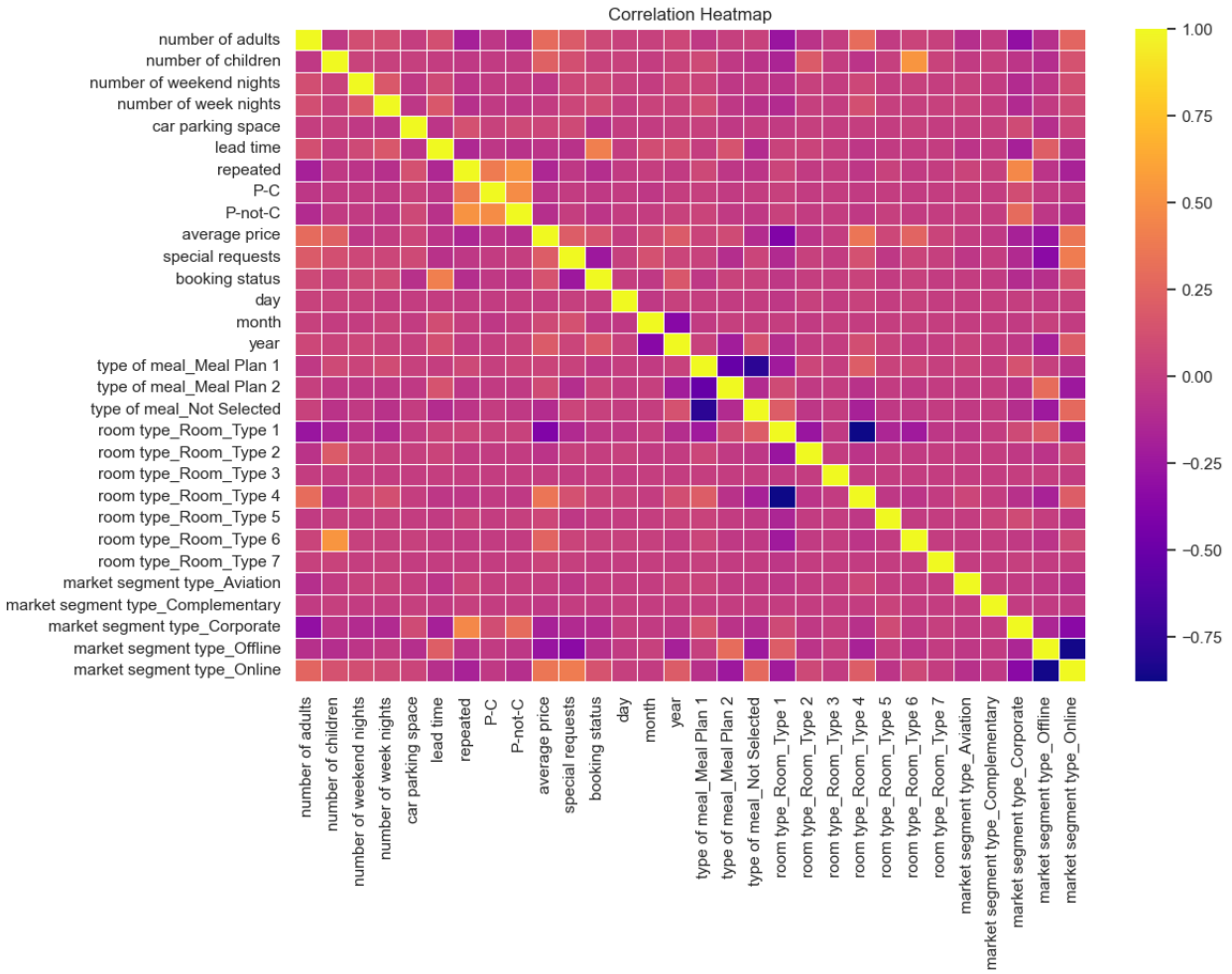

Figure 3. Correlation heatmap showcasing the correlation between variables

In the above figure (Fig.3), we see a strong correlation between P-not-C and repeated, indicating that repeat customers are more likely to not cancel their reservation, based on previous interaction with the hotel. We also see a strong correlation between the number of children included in the reservation and customers booking Room Type 6, providing insight into the size of room and room accommodations our customers require, especially if the reservation booked is for an entire family.

As part of our model development, we incorporated the "k-best" function, which identifies the "k" best or most advantageous solutions from a set of options. The value of "k" is a parameter that can be adjusted based on the specific requirements of the task at hand. In our

implementation, we set "k" to 10 to identify the top ten features influencing Hotel Reservations (scikit, n.d.-b, 2011). These features are highlighted in Figure 4.

While these ten factors are crucial in understanding Reservation Cancellation, their individual impact varies. Most of them have an influence score that does not exceed 1000. For instance, "Car parking space" and "number of weeknights" have the least impact scores of 216.59 and 248.87, respectively. In contrast, "lead time" and "special requests" significantly impact the score, with scores of 6755.25 and 2136.14, respectively. Therefore, most Reservation Cancellations are concentrated around these two factors.

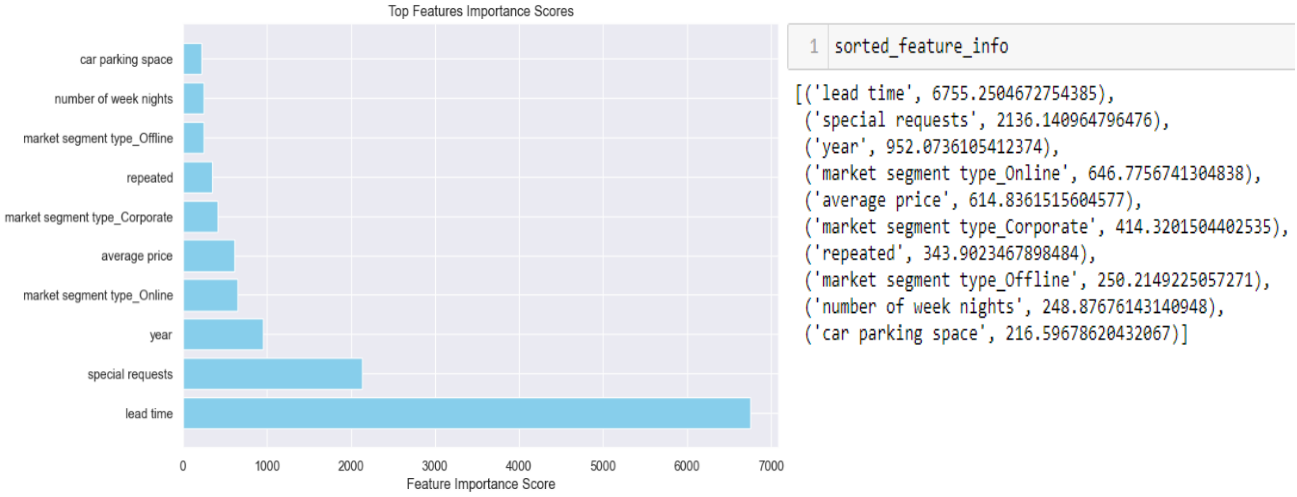

Figure 4. Top 10 features Impacting Booking Status

Subsequently, we used neural network models to capture the intricate relationships among these variables. Neural networks, fundamental components of artificial intelligence inspired by the human brain's structure, consist of interconnected nodes, or artificial neurons, organized in layers. They excel at learning complex patterns and making predictions based on input data. During the learning process, the connections between neurons have weights that adjust, allowing the network to adapt and enhance its performance over time. Neural networks find applications in various domains, such as image and speech recognition, natural language processing, and predictive analytics, showcasing their versatility in solving elaborate problems and contributing to advancements in machine learning (Yiu, 2021).

Hidden nodes in a neural network are the nodes or neurons in the layers between the input and output layers. They play a crucial role in learning complex patterns from the input data. The number of hidden nodes is a critical aspect of designing a neural network, and it can impact the model's ability to generalize well to new, unseen data. The choice of the number of hidden nodes in a neural network involves a combination of established guidelines and adaptability to the specific problem at hand (scikit, n.d.-b, 2011).

In the first neural model, the rule of thumb is to take half of the number of input features and add 1. This calculates the number of nodes in the hidden layer of the Multi-Layer Perceptron (MLP) model. This provides a quick starting point and is commonly used as a general guideline for determining the initial number of nodes in the hidden layer. Since we are using 10 predictors, the expression used would result in 5 neurons for the hidden layer.

The second neural model takes a different approach by balancing model complexity. The number of nodes in each subsequent layer is calculated by halving the nodes, aiming to strike a balance that avoids issues like underfitting or overfitting. This gradual reduction in complexity helps ensure that the network captures essential features without becoming overly complex. The number of nodes is influenced by the number of input features, and by halving the nodes in each subsequent layer, it gradually reduces the complexity of the representation. In this case, the first hidden layer would contain 5 neurons and the second would contain 3 neurons.

The third neural model introduces an element of experimentation and domain knowledge. The number of nodes in each layer is chosen based on trial and error, considering the specific characteristics of the problem. For example, a larger number of nodes in the first layer (50) is selected to capture complex patterns, while a smaller number in the second layer (25) acts as a form of dimensionality reduction, preventing overfitting and focusing on higher-level features. In essence, these three approaches represent a pragmatic combination of general guidelines, adaptability, and informed experimentation to arrive at the most effective configuration for the neural network.

In a machine learning regression model, there are two types of parameters: model parameters and hyperparameters. Model parameters are determined through the training process on the provided data. On the other hand, hyperparameters, responsible for defining the model's structure and training procedures, are set by users prior to the training phase. Users assign these hyperparameters to guide the overall configuration and behavior of the model during the learning process. We have chosen hidden layer size and activation to control the capacity of neural networks and it makes it easy to understand the complex patterns.

Parameters of Neural Network

The neural network parameters play a crucial role in shaping the model's architecture and influencing its ability to learn from data (scikit, n.d.-b, 2011). The "hidden_layer_sizes" parameter determines the number of nodes in each hidden layer, controlling the network's capacity to capture complex patterns. In the example, it is set to (hidden_layer_nodes,) for a single hidden layer or (hidden_layer_nodes_1, hidden_layer_nodes_2) for two hidden layers. The "activation" parameter specifies the activation function used in the hidden layers, with "relu" chosen for its ability to introduce non-linearity and promotes the learning of intricate patterns. The "solver" parameter dictates the optimization algorithm, and "adam" is selected for its adaptive learning rates, fast convergence, and general applicability. The "max_iter" parameter sets the maximum number of training iterations to 1000, allowing the model to converge effectively. The "alpha" parameter, a L2 regularization term, is set to 0.0001 to balance regularization and fitting the training data. Other critical parameters include "batch_size" (32), "learning_rate_init" (0.001), "early_stopping" (True), "validation_fraction" (0.1), and "n_iter_no_change" (10), each chosen to enhance the model's performance, prevent overfitting, and optimize training efficiency. These parameter choices reflect a thoughtful balance between computational efficiency and the model's capacity to generalize effectively to new data.

Results

Model	Accuracy	F1 Score	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)
1st Neural Network Model	0.80	0.66	0.83	0.89	0.86	0.72	0.61	0.66
2nd Neural Network Model	0.81	0.68	0.85	0.87	0.86	0.70	0.66	0.68
3rd Neural Network Model	0.84	0.77	0.87	0.91	0.89	0.77	0.71	0.74
Logistic Regression Model	0.79	0.64	0.82	0.89	0.86	0.70	0.58	0.64
K-Nearest Neighbors Model	0.85	0.75	0.88	0.91	0.89	0.78	0.72	0.75
Decision Tree Model	0.86	0.77	0.89	0.90	0.90	0.78	0.75	0.77
Random Forest Model	0.87	0.79	0.89	0.93	0.91	0.82	0.75	0.79

Table 1. Results of Neural Network and Ensemble Learning Models

Neural Network Performance Trends

The accuracy of the neural network models (Table 1) gradually improves from the 1st to the 3rd model, reaching 0.80, 0.81, and 0.84, respectively. This suggests a positive trend in the model's ability to correctly classify instances. The F1 score, a balance between precision and recall, also shows improvement across the three models (0.66, 0.68, and 0.77). This indicates an enhancement in the models' ability to handle both false positives and false negatives.

Precision, Recall, and F1-Score Analysis

In general, the precision for both Class 0 and Class 1 increases across the neural network models. This signifies an improvement in correctly identifying positive instances (precision). Recall for Class 0 and Class 1 remains relatively high and stable, indicating the models' ability to capture a large proportion of actual positive instances. F1-scores for both Class 0 and Class 1 exhibit an upward trend, showcasing an overall improvement in the balance between precision and recall.

The neural network models consistently outperform the logistic regression model in terms of accuracy, F1 score, precision, recall, and F1-score for both classes. When compared with traditional machine learning models like K-Nearest Neighbors, Decision Tree, and Random Forest, the neural network models generally show competitive or superior performance, especially in terms of F1 scores. The neural network models seem to benefit from increased complexity and capacity, as shown by the improvement across the other models. However, it's essential to consider potential overfitting as the model complexity increases. Further tuning of hyperparameters, adjusting the neural network architecture, or exploring more sophisticated neural network models could potentially enhance performance.

Discussion

When searching for the optimal number of neurons within each hidden layer, another option involves manually selecting the range of neurons to explore. However, it's important to note that this approach can be time-consuming and requires manual intervention. Manually choosing the range involves making informed decisions about the potential values that might yield better model performance. This process may require iterations and adjustments, adding complexity to the model development phase.

In the context of predicting booking status, neural networks are a powerful tool, showcasing consistent improvements in accuracy and F1 scores across different model architectures. The three neural network models exhibited progressively enhanced performance, achieving accuracy values of 80%, 81%, and 84%, respectively. The F1 scores for

both classes also displayed an upward trend, with the third neural network model achieving a notable F1 score of 0.77 for class 1. These results indicate that as the complexity of neural network models increases, their capacity to capture intricate patterns and relationships in the data improves, leading to more accurate predictions of booking status.

The comparison with ensemble learning algorithms, including Random Forest, Decision Tree, and K-Nearest Neighbors, revealed interesting trade-offs. While ensemble algorithms demonstrated competitive accuracy and interpretability, the neural networks consistently outperformed them, especially in terms of capturing complex relationships within the data. The decision to choose between the two approaches depends on the specific requirements of the project.

Conclusion

Neural networks stand out as an effective choice for predicting booking status. Their ability to learn and adapt to complex patterns in the data makes them well-suited for tasks where intricate relationships play a crucial role. However, it's essential to acknowledge the trade-offs, including the increased computational requirements and the challenge of interpretability associated with neural networks. The decision to employ neural networks or ensemble learning algorithms should be made based on the project's goals and constraints. For tasks demanding a high degree of accuracy and the ability to capture complicated relationships, neural networks prove to be invaluable. However, for scenarios where interpretability is paramount, ensemble algorithms can provide transparent insights into the factors influencing booking status. Ultimately, the success of the predictive model relies on consideration of these factors and an alignment with the specific objectives of predicting booking status in each context.

Bibliography

1.11. ensembles: Gradient boosting, random forests, bagging, voting, stacking. scikit. (n.d.-a). https://scikit-learn.org/stable/modules/ensemble.html

1.13. feature selection. scikit. (n.d.-a). https://scikit-learn.org/stable/modules/feature_selection.html

Aboelwafa, Y. (2022, September 19). *Hotel booking cancellations predictive analysis*. Kaggle. https://www.kaggle.com/code/gwen08/hotel-booking-cancellations-predictive-analysis

Sklearn.neural_network.MLPClassifier. scikit. (n.d.-b). https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Yiu, T. (2021, September 29). *Understanding Neural Networks*. Medium. https://towardsdatascience.com/understanding-neural-networks-19020b758230