Joseph Annand

DSE 6211

Final Project

## Executive Summary

## Approach & Data

To help ABC hotels identify bookings that have a high risk of cancellation, a data set containing 36,238 bookings and a supervised feed-forward dense neural network are used to determine the probability of cancellation for a given booking. Each observation in the given data set is a different booking at ABC hotels and includes information about each booking, including number of guests, room type, arrival data, etc. A neural network that predicts whether a booking will be cancelled or not was trained using ABC hotels data set. Of the 36,238 bookings, 11,878 bookings were cancelled, approximately 33%. To mitigate such high rates of cancellation, a model that predicts the likelihood of cancellation for a given booking can be employed by ABC hotels to determine which customers are most in need of targeted advertisements or special promotions and offers to reduce the risk of them cancelling.

Prior to training the neural network model, the data set had to be processed for the purpose of extracting more useful information from what was already provided and transforming the data into a format that the neural network can take as input. To begin, the ABC hotels data set was divided into a training set and a test set. The training set contained 75% of the observations from the original data set and was used to train the neural network. The test set contained the remaining quarter of the original data set and was used to evaluate the neural network once it was trained. A series of feature engineering steps were performed on the training set to prepare the data for the neural. Initially, the categorical features, non-numeric variables, of the training set needed to be one-hot encoded before it could serve as meaningful input in the model. One-hot encoding will create a separate feature for each of the unique values of all the categorical features in the training set. Some features, however, have rare cases, such as one observation with nine children listed on the booking, for example. To reduce the number of features produced in the training set while also ensuring that the training and test set will have the same number of features, rare values in a handful of categorical features are grouped together. Specifically, number of previous bookings not cancelled, number of children, number of previous cancellations, number of week nights, number of weekend nights, and number of special requests were the categorical features with rare cases that were grouped together. The arrival date of each booking does not appear to be useful on its own, but more useful information can be inferred from it and used in the model. From the arrival date, the season of the arrival date and whether the date is a weekday or weekend was determined and added to the training set. After converting all categorical features to factors, they were one-hot encoded. The same steps were performed on the test set.

After engineering the categorical features in both the training and test set, the numerical features were scaled. The two numerical features, lead time and average price per room, are represented in days and dollars, respectively. To ensure that neither feature dominates the other simply because of the difference in units, these features are centered using the means and standard deviations of each feature in the training set. Finally, the target variable or the labels are 0/1 valued so that 0 represents the booking not being cancelled and 1 represents the booking being cancelled.

Two feed-forward dense neural networks with slightly different architectures were trained and evaluated. The relu function is the default activation function for hidden layers in a multilayer perceptron, so the relu function is used for all hidden layers of the neural networks. For the output layer, the number of units is equal to 1 because the aim of the neural network is to provide a single output for predicting the status of a given booking. For a binary classification problem, the sigmoidal activation function is used so that all outputs are between 0 and 1, with 0 corresponding to a 0% chance the booking will be cancelled and 1 corresponding to a 100% chance the booking will be cancelled. The optimizer for the models is RMSprop, which is an adaptive learning rate algorithm that helps reduce the amount of computational effort required to train the neural network compared to other optimizers. The loss function used for the neural network is binary cross-entropy; this loss function calculates the dissimilarity between true labels and predicted labels when the output is between 0 and 1. The metric used in training the model is accuracy to measure how well the model is performing in terms of predicting booking status.

Initially, a model that could easily overfit the data was created. This model consisted of three hidden layers, two of which had 100 nodes and the third had 50 nodes. This architecture is much larger than what is likely required for the training set, as there are more nodes in the first two hidden layers than there are features in the training set. Once the model that can overfit is created, the overfit model is modified to increase generalization, or the model’s ability to make predictions on unseen data. L2 regularization, batch normalization, and dropout are implemented to do so. In L2 regularization, overfitting is mitigated by placing constraints on the weights of the model through adding to the loss function of the model at a cost proportional to the square of the value of the weight coefficients. In our model, this coefficient is 0.002, and it is used in every hidden layer. Batch normalization standardizes the inputs at every layer of the neural network, a similar action that was manually performed during the feature engineering steps. Dropout is a regularization technique that drops out some output features in each layer of the model. As stated previously, the model is much larger than what is probably required for this problem; dropout reduces the size of the output of each hidden layer by 50% to reduce overfitting. Finally, early stopping is used to stop training the model once the validation loss stops decreasing after two epochs, so that the model does not overfit and the model with the best performance on the validation set is found. After training this first model, a second model with a smaller capacity was created and trained. The smaller model consists of two hidden layers: one with 75 nodes and the other with 37 nodes. The same generalization techniques were employed with a change in the drop out rate from 0.5 to 0.2.

## Detailed Findings & Evaluation

## Recommendations

For example, the model could be developed to classify each booking as “high risk of cancellation” or “low risk of cancellation” to separate the bookings that need some intervention to prevent cancellation and the bookings that do not. The probability threshold to classify the booking as one of the two categories may vary for several reasons. If the company has committed a plethora of money and resources into cancellation prevention, the threshold may be set relatively low like 50% chance of cancellation. If the company has limited capacity for extra advertisements and offers, then the threshold may be set to a higher probability value like 75%. The model could also classify bookings with a high cancellation risk into priority groups. Then, the hotel could easily distinguish between a booking with a 90% chance of cancellation and a booking with a 75% chance so that it prioritizes sending advertisements and offers to the booking at a higher risk. Nonetheless, the model will provide useful information to ABC hotels that will help keep their rooms filled with customers.