ABC Hotels Solutions for Reservation Cancellations

By Ben Kelley

DSE6211

We are looking to identify customers with an increased risk of cancelling their hotel reservations. To do this we will be using the customer dataset (project_data). Using this dataset, we will be able to identify the characteristics of customers who have cancelled reservations and create a model that will allow us to predict the likelihood of a cancelled reservation in the future.

Problem statement: Through analysis of the project_data dataset, we will be evaluating customers who have cancelled reservations in order to predict future reservations that have a high likelihood of being cancelled. Through supervised classification, we will be assigning a rank to each booking between 0 and 1 to predict whether it is likely the booking will be cancelled. Based on this rank the model will predict the booking will either be cancelled or not cancelled. Once created, the model will be able to assign a prediction to a set of future bookings to help target bookings with a higher risk of cancellation which can be addressed proactively.

Data Gathering: As described above, we will be using the project_data dataset which contains information from roughly 36,000 bookings between July 2017 and December 2018, including both cancelled and non-cancelled reservations. Within this dataset are 17 variables, some of which are numerical and some of which are categorical.

Data Cleaning: By working with this data set in R and RStudio, we will be performing various data cleaning functions which will allow us to separate relevant data from non-relevant data and identify correlations within the data that will be indicators of cancellation. I have begun to clean the dataset but have not made any significant changes to the data yet. As a precaution I have made sure there are no variables with any N/A or unknown values. I have also added a new column named “cancel” to which I have assigned a 1 to each cancelled booking and a 0 to each booking that was not cancelled. This will help during the data analysis instead of relying on the booking_status column.

The variables included in the analysis are mostly self-explanatory, but I have included some descriptions as I understand them. The variables are as follows:

no_of_adults = number of adults (numerical)

no_of_children = number of children(numerical)

no_of_weekend_nights = number of weekend nights (numerical)

no_of_week_nights = number of weeknights (numerical)

type_of_meal_plan = type of meal plan (categorical)

required_car_parking_space = required car parking spot (categorical) 1 = yes, 0 = no

room_type_reserved = categorical, I will need to find a way to classify these room types

Leadtime = days between booking and reservation start (numerical)

arrival_date = date the reservation is to begin.

market_segment_type = categorical

repeated_guest = whether the guest has stayed at this hotel in the past or not (categorical), 1 = yes, 0 = no

no_of_previous_cancellations = has this customer cancelled a reservation in the past (numerical)

no_of_previous_bookings_not_cancelled = has this customer kept a reservation in the past (numerical)

avg_price_per_room = average cost per room (numerical)

no_of_special_requests = number of special requests (numerical), I assume this would refer to any request the guest would have that is not automatically part of a hotel stay, i.e. cot in the room, bathrobe, champagne etc.

booking_status = categorical, cancelled or not cancelled

Approach & Data: During the data cleaning process, I found 4 categorical variables and performed numerical encoding using the one-hot encoding method. This allowed me to assign these variables numeric values so I could use them in the model instead of disregarding the information entirely. I found this particularly

important because I wanted to make sure I used all the relevant data.

After converting the categorical variables to numeric values, I scaled all the variables with the exception of the target. This converted all values on a scale between -1 and 1 with the mean at zero. This approach is very useful because it prevents one variable from overshadowing another just because it may use a larger scale. Once all the variables were scaled, I converted them into tensors using the array function.

Throughout the course of this project, I tested many variations of the feed forward, dense neural network. I tested variations including number of epochs, number of units in each hidden layer, number of hidden layers within the model and, once the model had run, I used different thresholds for the classification problem.

Analysis: In the analysis I plan to use neural networks to run models on this data to find the key predicting variables in reference to cancelled bookings. I still need to learn more about the data before I am ready to create a model. I have not yet identified the loss function or optimizers. I will be expecting to use a logistic regression model since the output I will be looking for is a yes or no prediction on whether the reservation will be cancelled.

Once this model has been trained using the data set, it will be possible to use it on a list of current bookings for future dates and predict which bookings will be cancelled and which will not.

Expected Outcomes: Before performing the data analysis, I have a few predictions about which variables will be the key indicators of cancellation. My hypothesis is that lead time, number of special requests, average price per room and number of weekend nights will be the variables that best predict the probability of cancellation.

Executive Summary: Using the labs in class as a blueprint, I have built and run a neural network classification model to predict the outcome of the 'booking status' variable. So far, the results are satisfactory but I believe they can be improved through edits and modifications to this model. The following are the results for each model:

Model 1: For the preliminary classification model, I used 3 layers in the neural network with 10, 10 and 1 units for each layer respectively. For the activation functions for each layer, I used ReLU, ReLU and sigmoid respectively. I chose these because ReLU is a good option for the hidden layers and sigmoid was a good option for the output since we are performing a binary classification.

Using 100 epochs, the loss and accuracy curves are both trending in the right direction and have seemed to level off. I will note these in comparison to the subsequent models.

The AUC for Model 1 is 0.8878 – I will compare this AUC value to the other models and take this into account when choosing the best model to use in this application. I also calculated the false positive rate at 0.0963 and the true positive rate at 0.6653. I'm hoping that I can reduce the FPR while increasing the TPR in the following models.

This model is overfitting from 0.25 to 0.50, then underfitting from about 0.50 to 0.60, and then overfitting again from 0.60 to 0.90. Overall, the calibration curve remains close to the line but this can likely be improved.

After running this initial model (named Model 1), I reworked the hidden layers and epochs in differing variations to create 3 more predictive models. The history plots, ROC curves and calibration curves for all models I ran will be listed below so the differences can be seen close together.

Model 2: For the second model, I increased the number of hidden layers to 4, keeping the number of units in each hidden layer at 10, like Model 1. However, I increased the number of epochs from 50 to 75. I wanted to increase the epochs to give the model a higher chance of finding patterns and predicting more accurately but I am also aware that the model may begin to memorize the data instead of making accurate predictions. Since the data set is fairly large, I am confident 75 epochs is not too many. After running this model, I calculated the AUC at 0.8881 with a false positive rate (FPR) of 0.0909 and a true positive rate (TPR) of 0.6680.

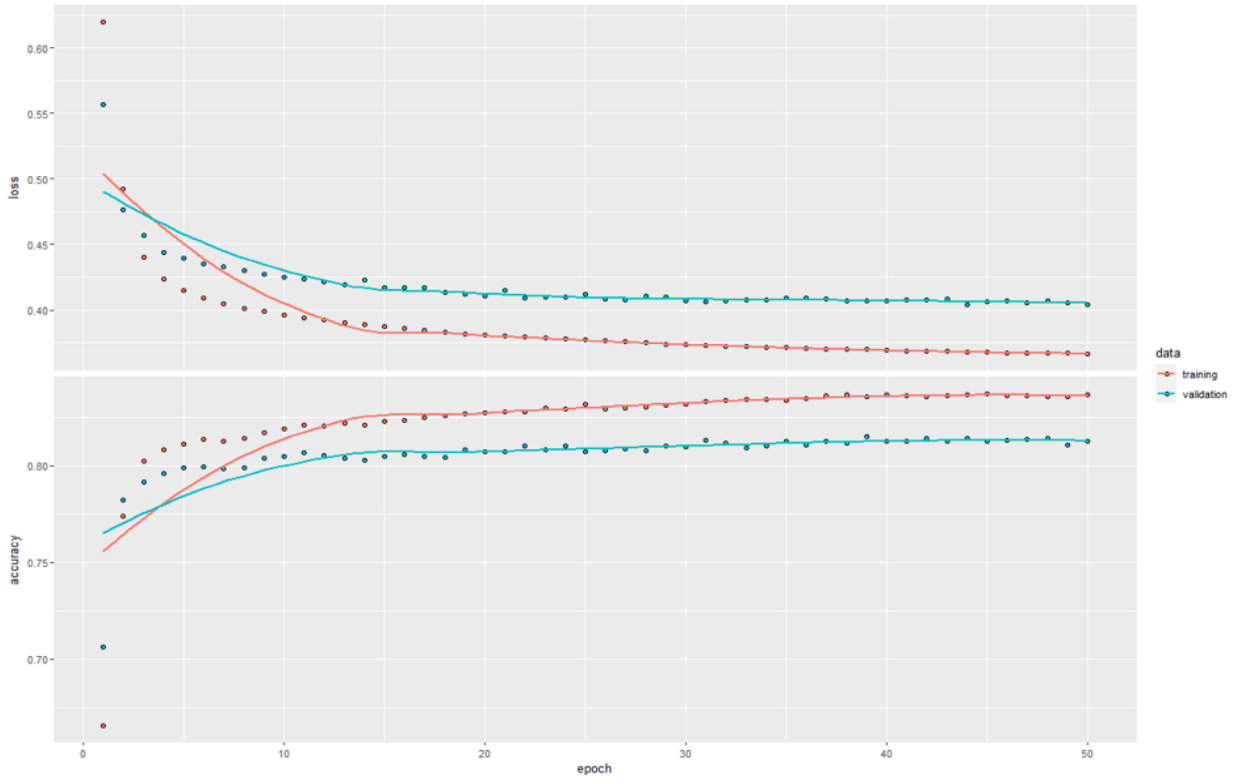
Model 3: For the third model I wanted to take things a step further. I reduced the number of hidden layers down to 3 since 4 hidden layers in Model 2 did not seem to make a big difference. However, for Model 3 I changed the number of units in each hidden layer to

20, 15 and 10 respectively. I also increased the number of epochs again, this time up to 100. I feel that 100 epochs is still a reasonable number and will not overtrain on the data. After running Model 3 I found an AUC of 0.9005, FPR of 0.0725 and a TPR of 0.6700. All 3 of these values are an improvement over both of the first 2 models. The history plot does not show a point at which the loss begins to increase and I feel confident that the model is not starting to memorize the data.

High Epoch Model: I also ran 1 final model with 500 epochs in the hope of finding the limit where the model may begin to overfit. According to the history plot, that limit is somewhere around 175-200 epochs as the curve begins to increase again. This can be deceptive because the AUC, FPR and TPR are all improved from the other models with 0.9055, 0.0881 and 0.7231 respectively. I ran this model only as a test and would not recommend it's use in a real world application.

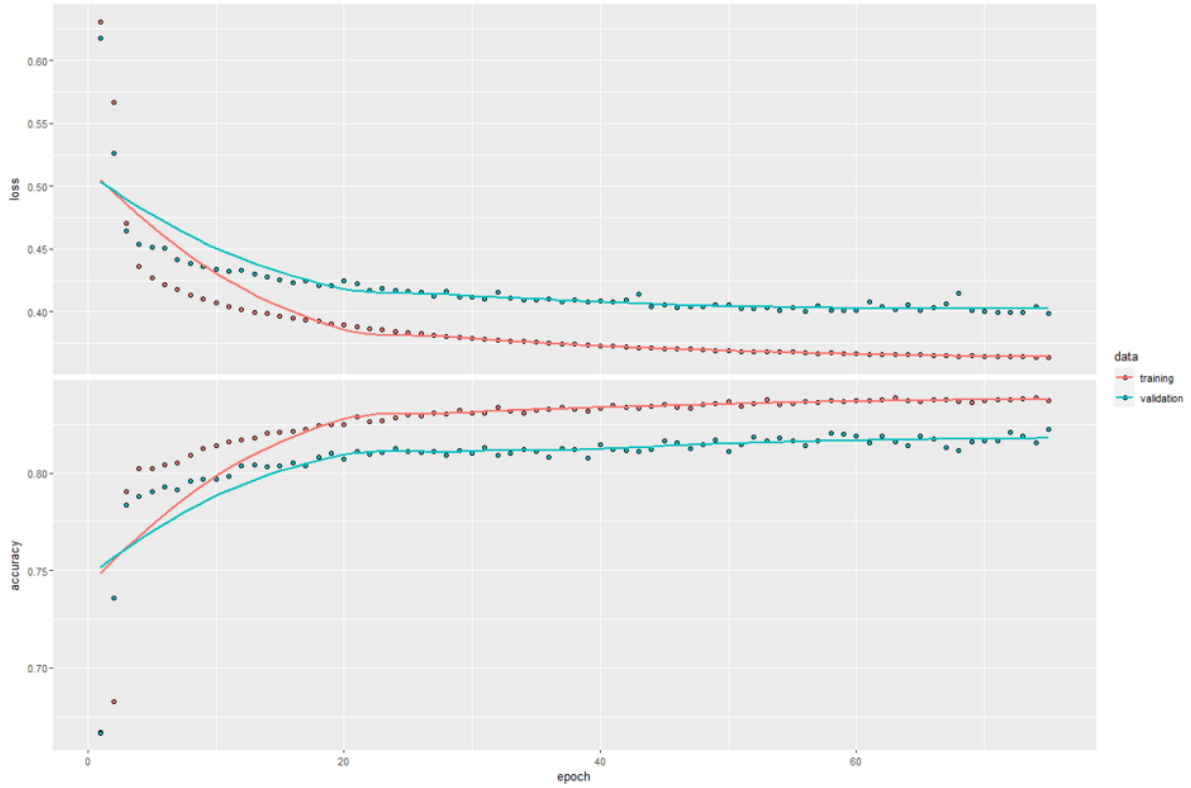
Detailed Findings and Evaluation: Below are the plots for all 4 models I ran which can be used as a visual aid in determining which model will work best for this business application:

History Plot - Model 1



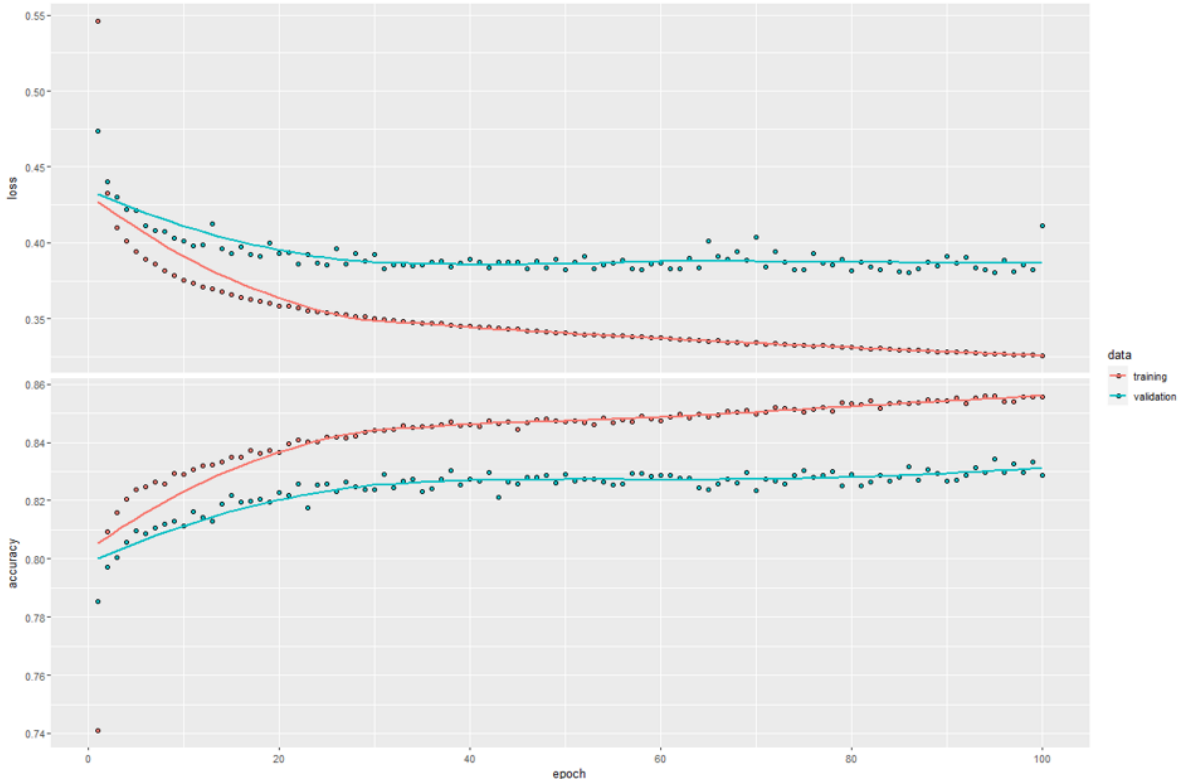
The history plot for Model 1 shows consistency. Both the training and validation points reduce loss and increase accuracy as the epochs approach 50.

Model 2



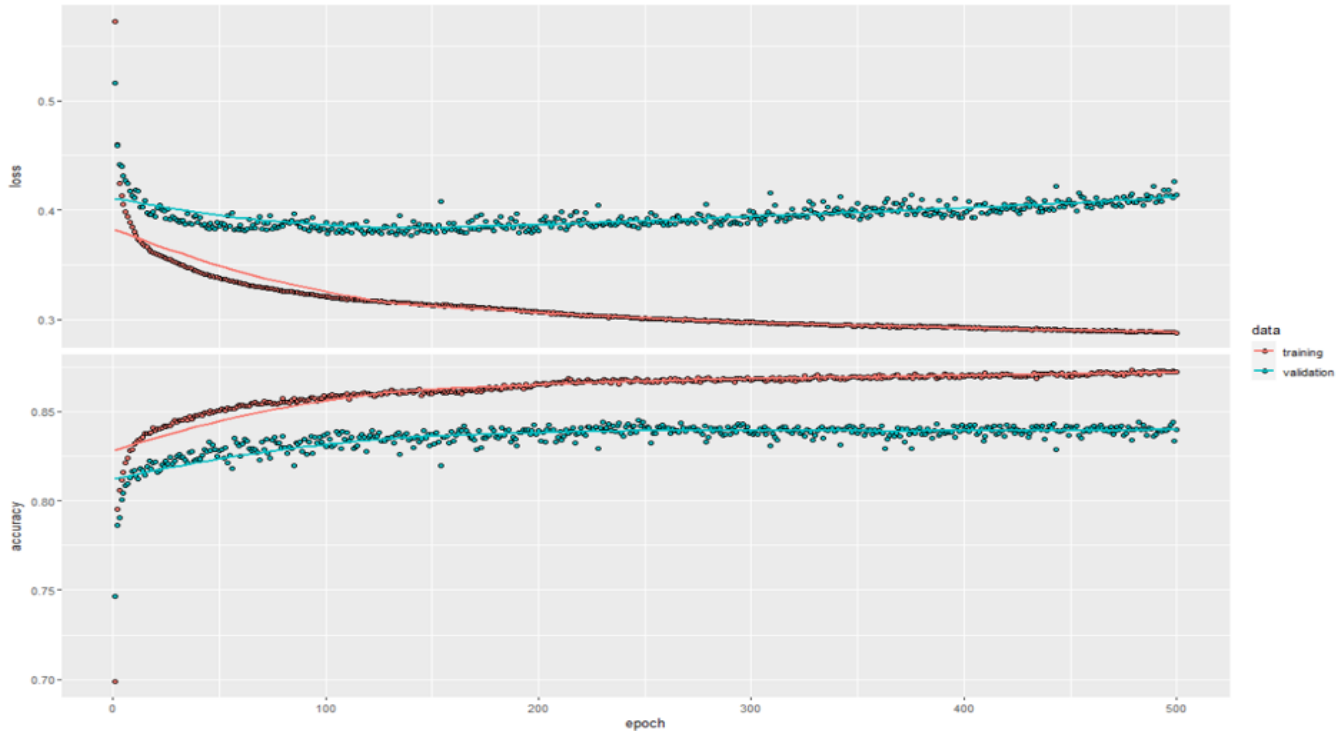
Model 2 also shows a reduction in loss and increase in accuracy as the epochs approach 75. However, you can see the points are becoming a bit less consistent along the curve.

Model 3



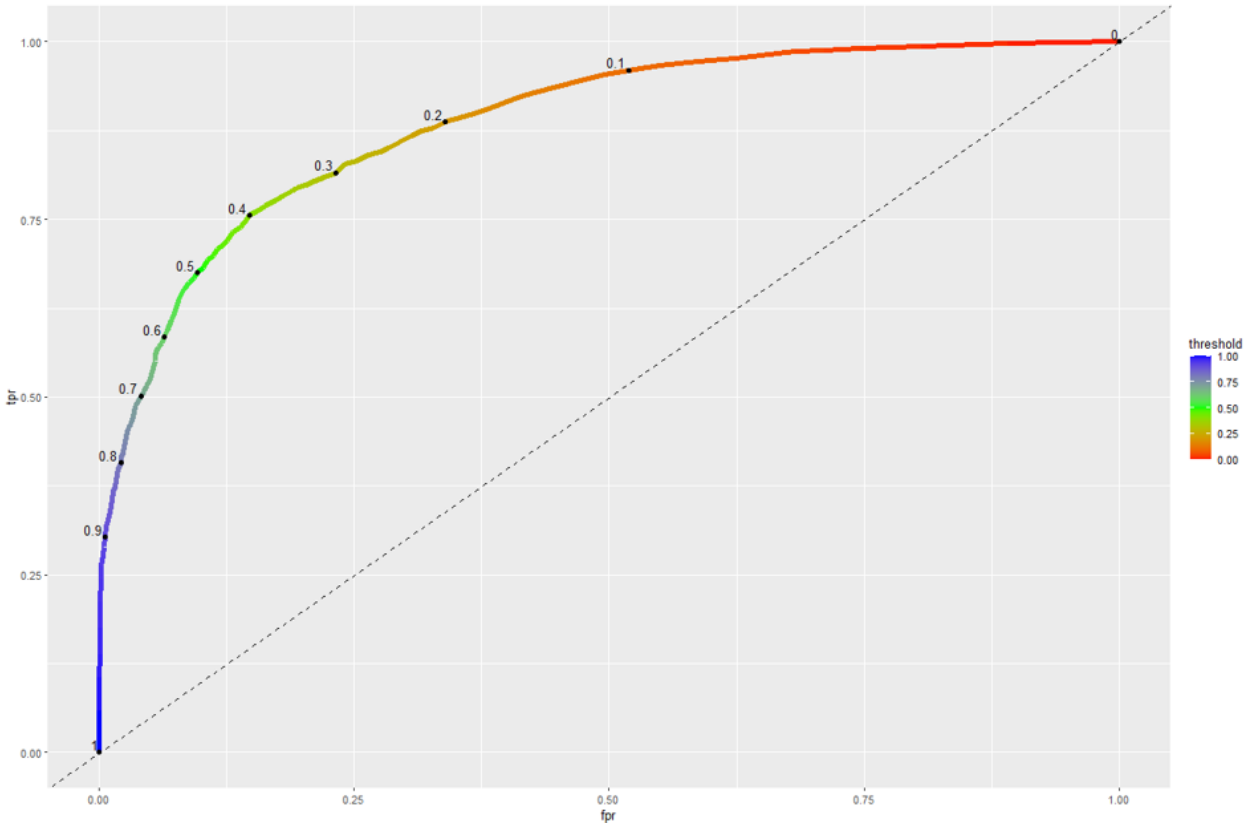
Again, in Model 3 we see the validation data is a bit more varied along the curve. We also see the loss in the validation curve level out a bit earlier and slightly diverge from the training curve.

High Epoch Model



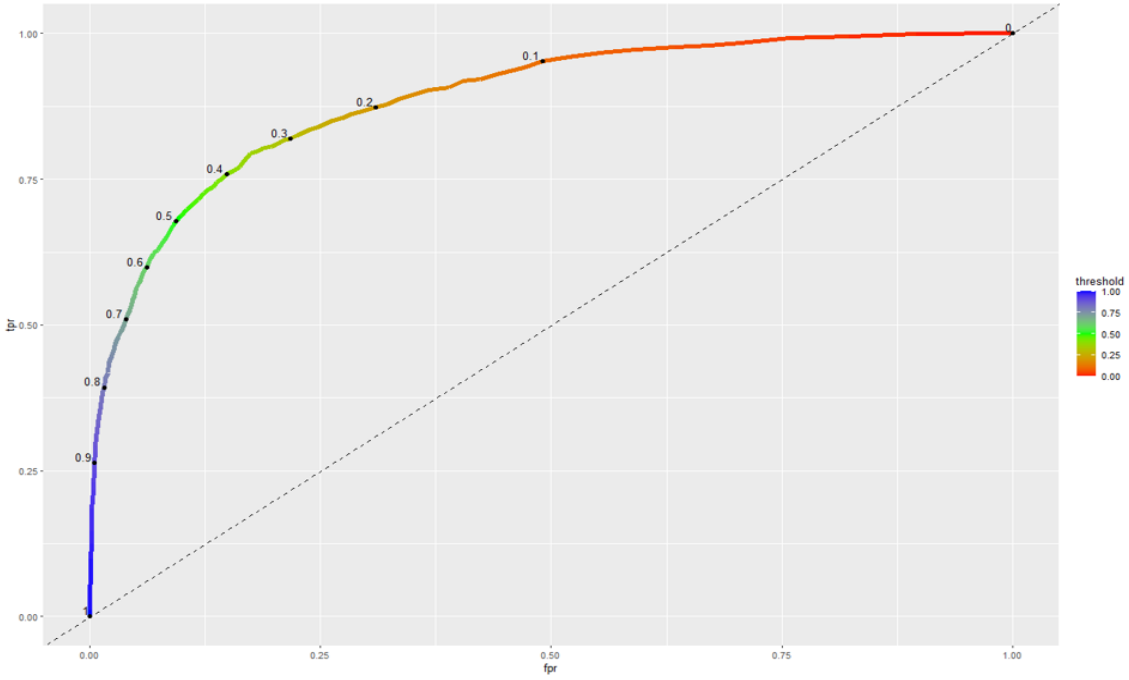
The curve in the High Epoch Model shows exactly what I was hoping it would. The points for the loss in the validation data not only level off but begin to increase around 175 epochs. This indicates to me that this is the point at which the model memorizes the training data and becomes less capable on the validation data.

ROC Curve – Model 1



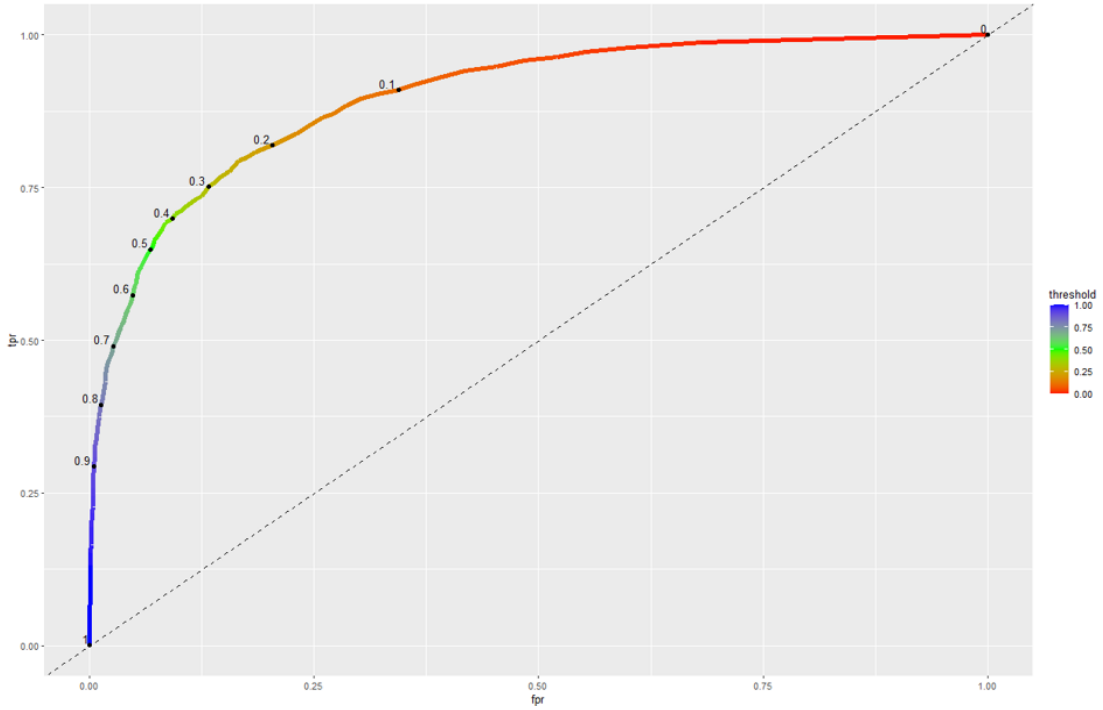
AUC = 0.8878

Model 2



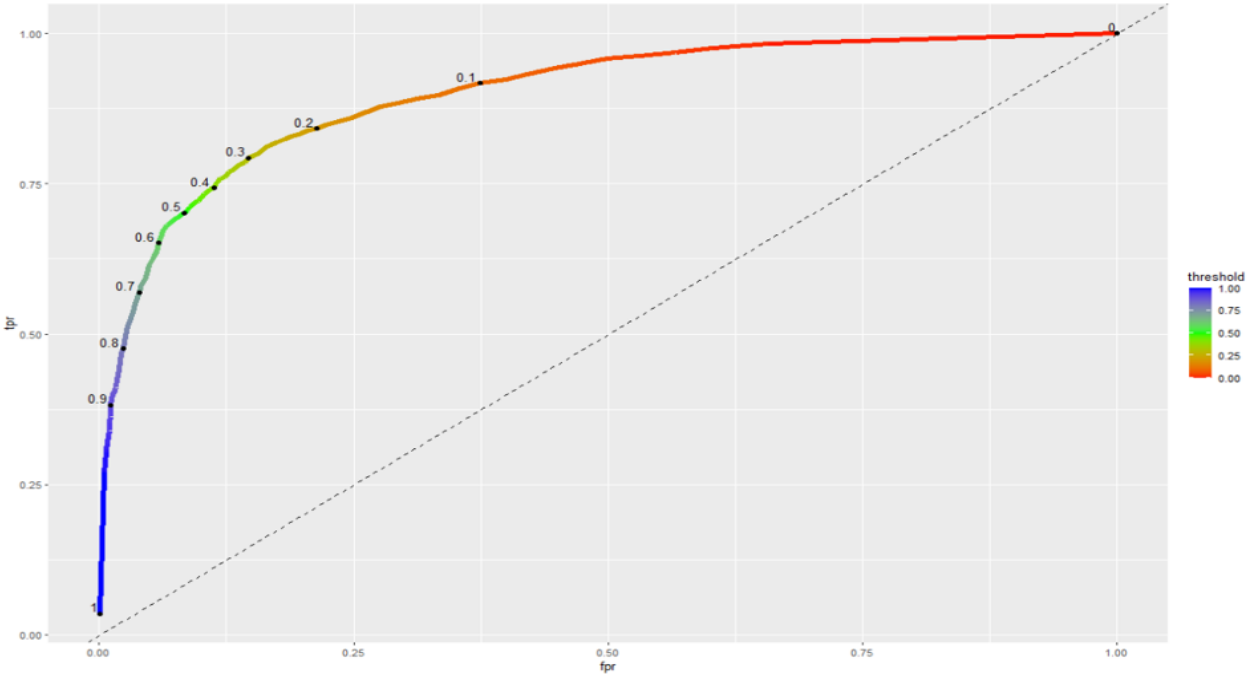
AUC = 0.8881

Model 3



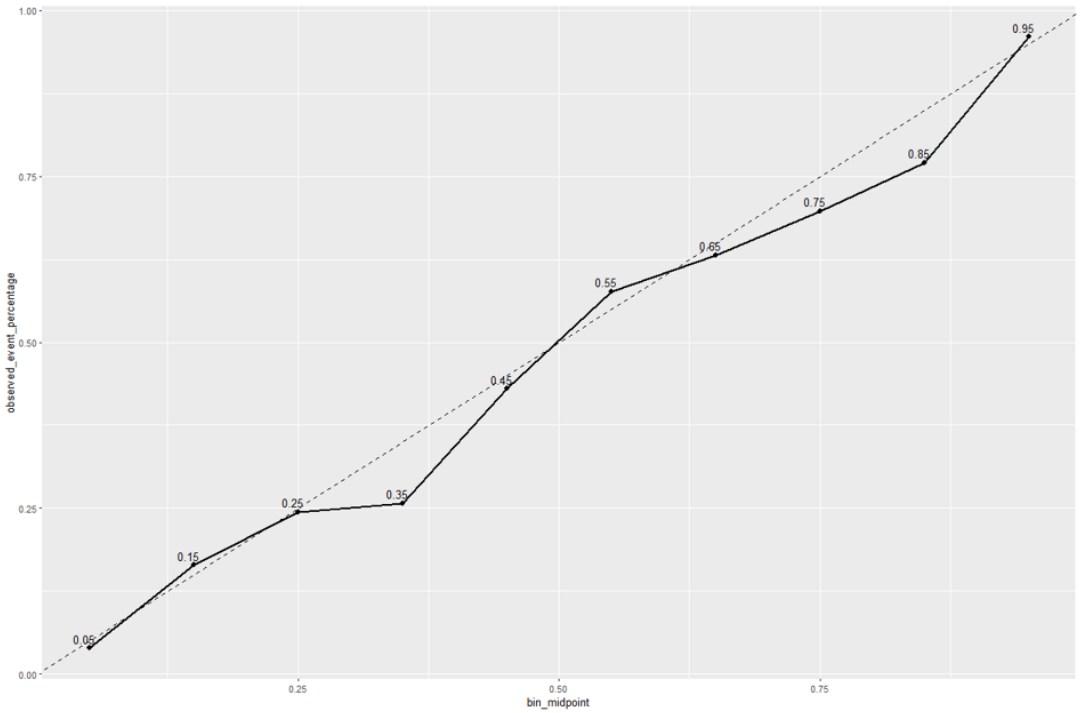
AUC = 0.9005

High Epoch Model



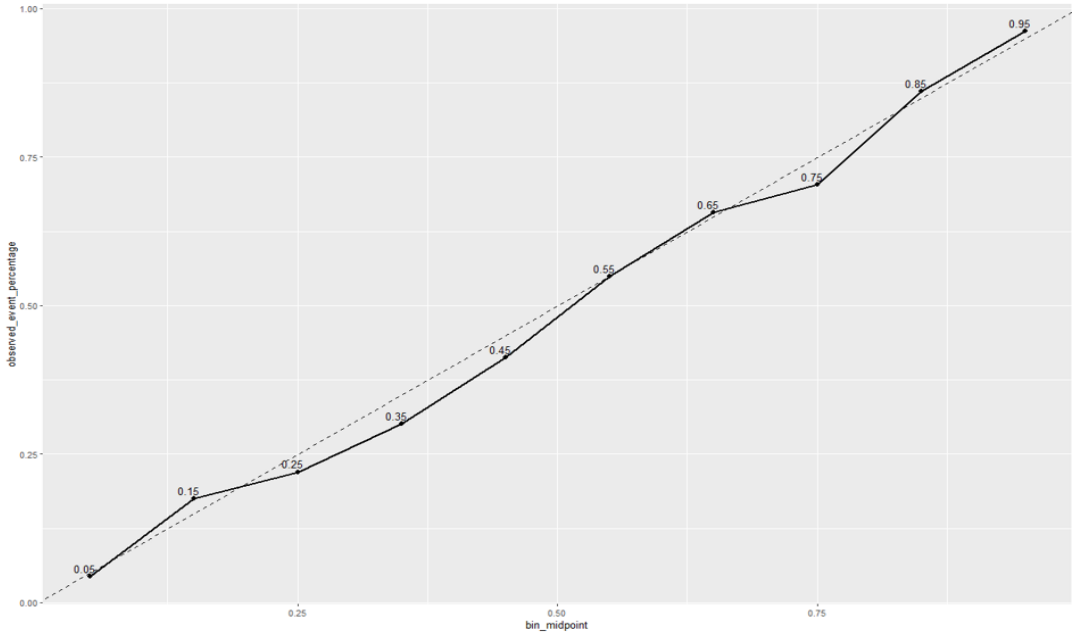
AUC = 0.9056

Calibration Curve – Model 1



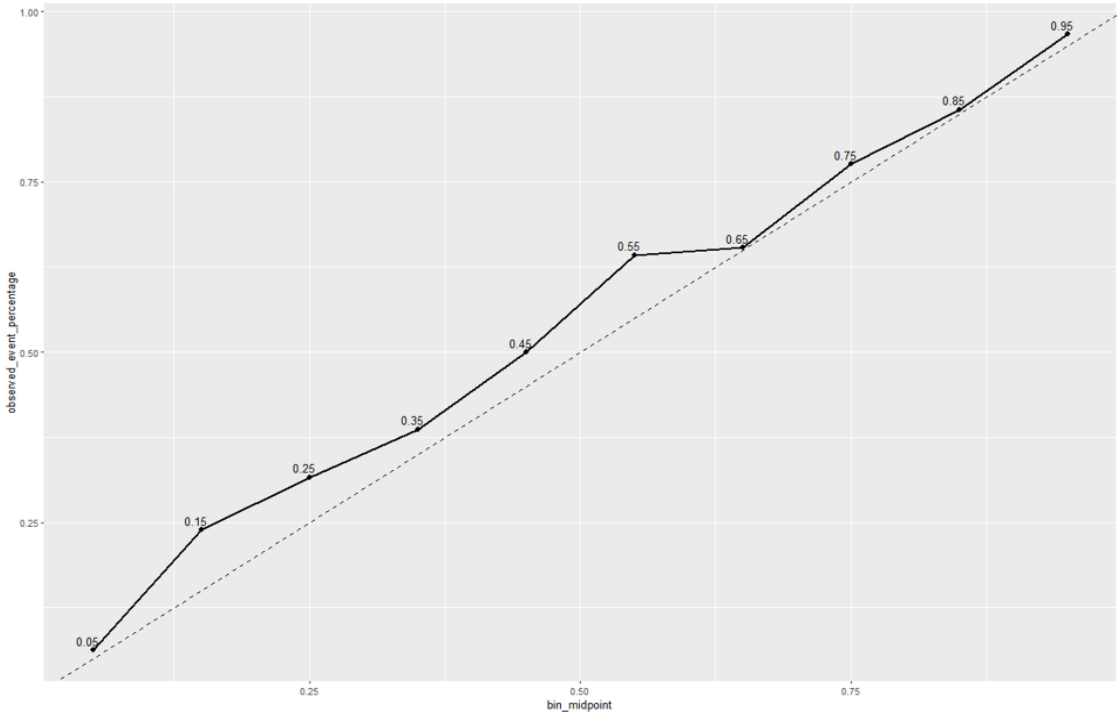
The calibration curve for Model 1 indicates a bit of overfitting but generally stays close to the dotted line.

Calibration Curve – Model 2



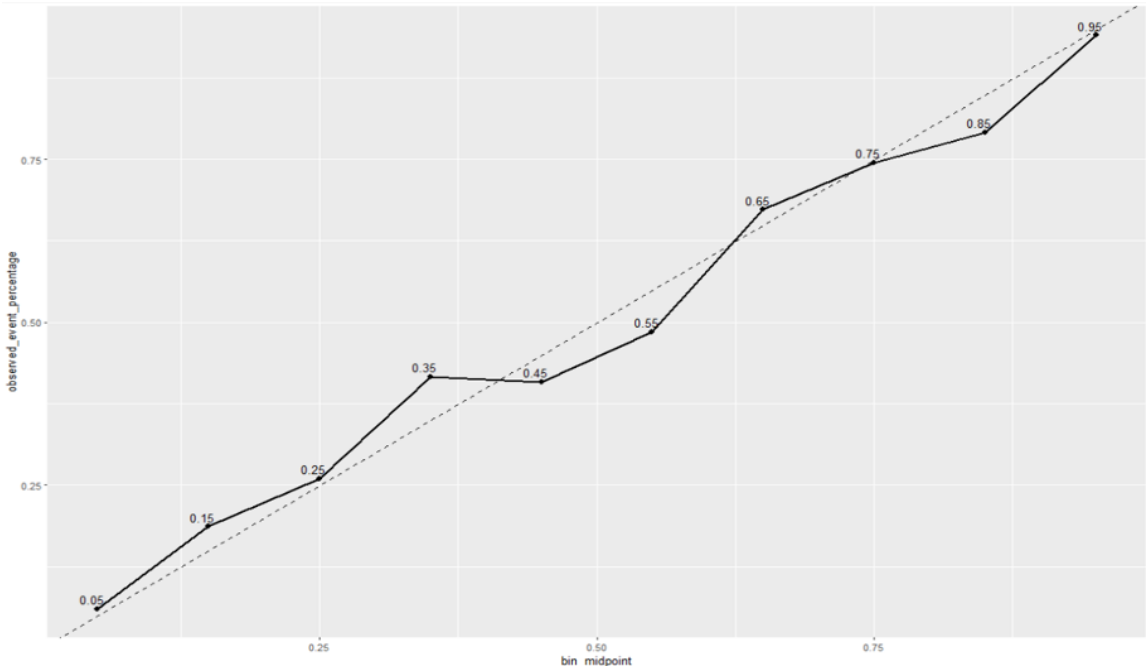
The calibration curve for Model 2 is closer to the diagonal line indicating less overfitting since the curve is still below the line in a couple segments as it was for Model 1.

Calibration Curve – Model 3



The calibration curve for Model 3 shows increased underfitting for a substantial portion of the diagonal line. Along with the history plot, this calibration curve shows that Model 3 is not the best model for this business application.

Calibration Curve – High Epoch Model



This calibration curve shows high variability and both underfitting and overfitting. This model was never intended to be used as a solution to this business application and this calibration curve confirms it would not be suitable.

In addition to running these models as shown above, I also ran each model at a threshold of 75% which in each case improved the false positive rate, usually lowering it considerably. The issue was that the true positive rate was greatly reduced in each model usually by close to 20%. After finding these results I chose to keep the classification threshold at 50%.

Recommendations: After running each of these models, I would recommend using Model 2 for this business application. Although the AUC, FPR and TPR are slightly better in Model 3, the history plot for Model 2 was more consistent showing less variance. After seeing the history plot in the High Epoch Model, I am confident that using 100 epochs in Model 2 is not going to overfit and the predictions will be reliable. The results in Model 2 are marginally better than for Model 1 but also only

marginally worse than Model 3 while maintaining the consistency in the history plot. Using Model 2 will be effective in predicting future booking cancellations and allow marketing steps to be taken to prevent those cancellations.