Assignment Week 2:

Model fitting, over-fitting and validation

For this week's assignment, use the provided R Markdown file to write your answers, code, explanation, and interpretation of the results. Upload the final document as a PDF file to *Canvas* on or before 9.00 a.m. in the morning of next Thursday (April 13, 2023).

Applied

In this assignment you are asked to use data on hotel reservations. The data is provided on *Canvas* in the file *data_hotel_reservations.csv* and it contains information on 36,275 reservations. The reservation information includes the unique identifier of each booking (*Booking_ID*), the number of adults (*no_of_adults*) and children (*no_of_children*) in the reservation, the type of meal booked by the customer (*type_of_meal_plan*), the number of week (*no_of_week_nights*) and weekend (*no_of_weekend_nights*) that the customer booked at the hotel, whether the customer needs a parking space (*required_car_parking_space*), the room type reserved by the customer (*room_type_reserved*), the number of days between the booking and arrival date (*lead_time*), the arrival date (*arrival_date*), month (*arrival_month*), and year (*arrival_year*), the market segment type (*market_segment_type*), whether the customer is a repeated guest (*repeated_guest*), the number of previous booking by the customer that were canceled (*no_of_previous_cancellations*) and those that were not canncelled (*no_of_previous_bookings_not_canceled*), the reservation's price per day (*avg_price_per_room*), the total number of special requests made by the customer (*no_of_special_requests*), and a flag indicating whether the booking was canceled (*booking_status*).

[Q1] Load and inspect the data. Complete the following actions: (1) create a column named *booking_canceled* which takes the (numerical) value 1 if *booking_status* is equal to "Canceled" - otherwise it takes the (numerical) 0, (2) replace each NA value in column *no_of_special_requests* with the (numerical) value 0, and (3) remove the columns *Booking_ID* and *booking_status*. Print a summary of the data. (1 point)

[Q2] Use a linear probability model and all the available variables still contained in the dataset to predict whether a booking was canceled. Also run

a LASSO regression with $\lambda = 0.01$ and the same model set-up. Display the output of the two models. Which variables does the LASSO regression set to 0? (2 point)

[Q3] Select the first 10.000 observations and use the *set.seed()* command with setting "123" as the seed to randomly create 5 folds containing an equal number of observations. Using these folds for cross-validation, build and compute the average accuracy of an SVM, a classification tree, and a random forest predicting whether a booking was canceled. For SVM use type = "C-classification", for the classification tree use method = "class" and parms = list(split = "information"), and for the random forest use ntree = 100 and importance = TRUE (you might want to transform the target variable to type factor). Where applicable convert all predicted probabilities that are > 0.5 to 1, otherwise convert them to 0. Use all the attributes except *type_of_meal_plan*, *room_type_reserved*, and *market_segment_type* as variables in your models. Briefly explain which model performed the best. (3.5 points)

[Q4] We want to also create a neural network model. To do so, we need to select how many neurons to include in the hidden layer of the neural network. Write a function that takes as arguments the (1) data, (2) model (i.e. formula with dependent and independent variables), and (3) a parameter "n" as inputs. This function must take the first 70% of the observations as the train set and use it to train a neural network with "n" layers (the rest of the observations should be the test set). The function should then return the model performance in terms of accuracy on (1) the training set and (2) the test set (i.e. return two accuracy numbers one corresponding to each set). Use this function in a loop and compute the (train and test) model performance of neural networks with n = 1, 2, 3,..., 15 as the sizes of the hidden layers. Create a graph that looks similar to the graphs on page 78 and page 83 in the session slides.

Use the same attributes/predictors that you used in the question 3. Prior to training your model use the *set.seed()* command with setting "123" as the seed. Also, transform the variables *arrival_year*, *arrival_month*, *arrival_date*, *repeated_guest*, and *required_car_parking_space* to type factor. In the nnet function use maxit = 300, and trace = FALSE. Convert all predicted probabilities that are > 0.5 to 1, otherwise convert them to 0. In order to save on computing time, use only the first 10,000 observations of the data set in this exercise (as we did in the previous question). Does the graph look like what you would expect based on what we learned in the lecture? Explain why. (3.5 points)