Aaron Chan

STAT 425

5/9/2019

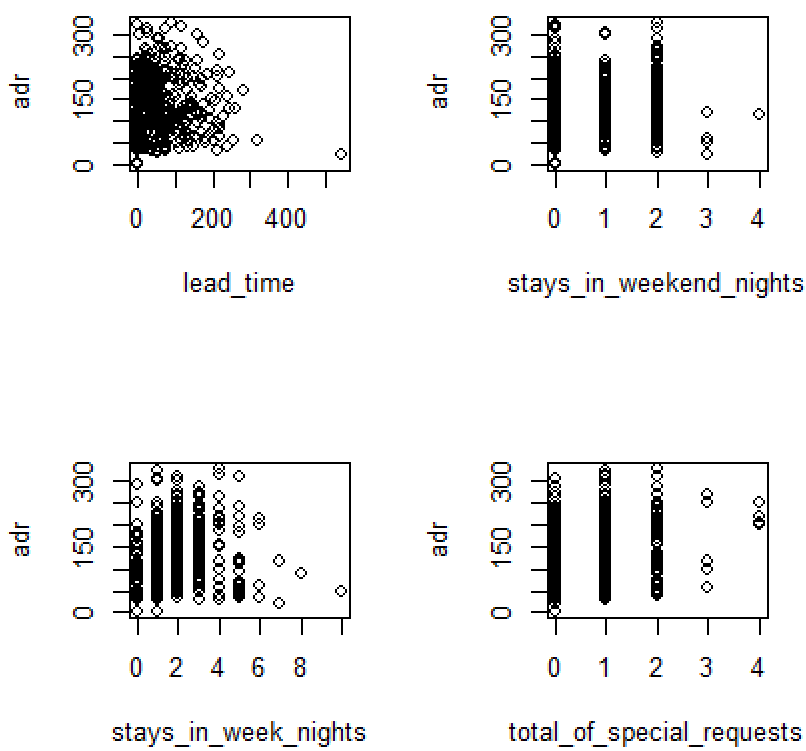
STAT 425 Final Project

The objective of this data analysis is to be able to find the best model to predict a certain outcome given a set of data. In this case, we are using hotel booking data given to us by our professor but taken from the paper: "Hotel Booking Demand Datasets". This data contains many categories of interest pertaining to hotel bookings, such as: average daily rate (*adr*), number of nights stayed, number of children/adults, and arrival times, just to name a few. There are over 400 rows of data for out specific dataset pertaining to "Resort Hotel", we should have enough data to accurately predict a certain variable if we choose to do so. In my case, I would like use average daily rate (*adr*) as a response variable since predicting future earnings depending on customer traits has some very useful real-world applications.

Right off the bat, we can see that some variables such as *is_canceled*, *arrival_date_month*, *meal*, *market_segment*, *reserved_room_type*, and *customer_type* are categorical variables. The number of occupants: *adults*, *children*, *babies*, can also be considered categorical variables as they are limited in value by available room sizes. Numeric variables in our data set are the remaining variables. Keep in mind that this data only pertains to data with *hotel* type "Resort Hotel". In this analysis, I decided not to keep variables such as *arrival_date_year*, *arrival_date_month,* and *arrival_date_week*, because *arrival_date_month* should be a good enough indication for seasonality. Assuming that economic conditions are similar each year, the month of the stay at "Resort Hotel" should be a good enough indicator.

Introduction of other unnecessary time variables may increase autocorrelation between variables. Interaction terms are needed between children and adults as well as babies and adults, for obvious reasons (children and babies cannot book hotel rooms by themselves). After graphing some of the numerical data against *adr* in Figure 1, we do see some evidence against linear trends, nothing in these graphs suggest a strictly linear relationship between explanatories and the response.

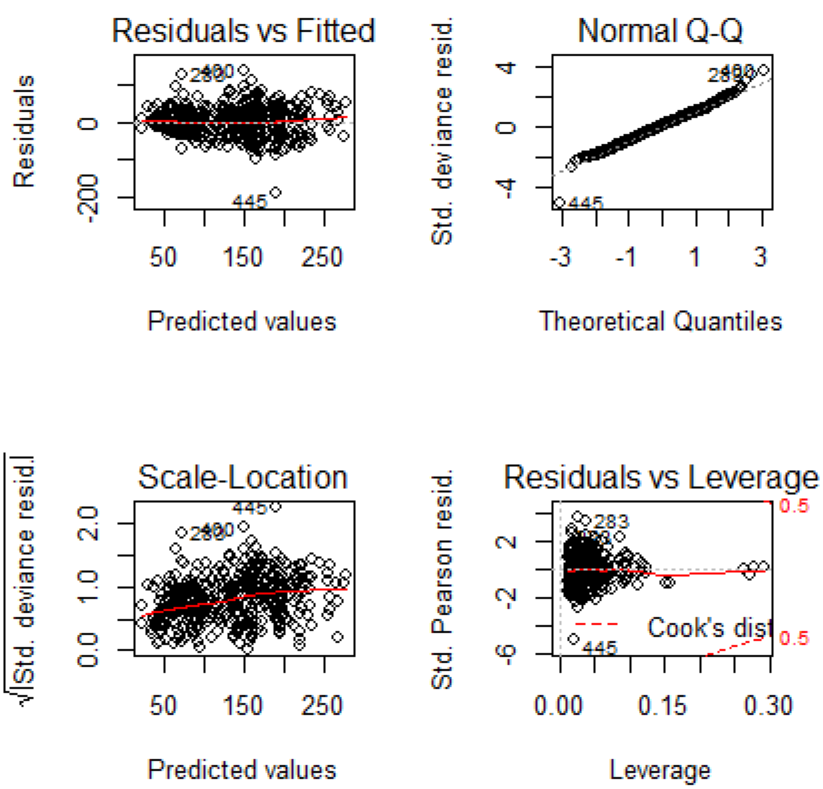
Figure 1



We will be using *adr* as our response variable, removing *is_canceled, reserved_room_type, meal, market_segment, customer_type,* and *meal* for simplicities sake, and all remaining variables as explanatory.

The first model we take a look at is *adr* against all other variables (*is_canceled, lead_time, arrival_date_month, stays_in_weekend_night, stays_in_week_night, adults, children,* *babies, total_of_special_requests*). This model is a simple multiple regression without interaction terms or transformations. We can analyze some of the assumptions of multiple regression using various plots as can be seen in Figure 2.

Figure 2



There are some assumption issues that we can take a look at in this model. For one, the Normal Q-Q plot (Figure 2) does not appear completely straight. In fact, at the ends, we can see bends, which are indicative of non-normality in our model. A Shapiro-Wilkes normality test proves our findings as well, testing H0 = data is normally distributed vs Ha = data is not normally distributed, shows we reject H0, data is not normally distributed.

```
        Shapiro-Wilk normality test

data:  residuals(mod1)
W = 0.98779, p-value = 0.0004878
```

The Scale-Location plot (Figure 2) shows that slight issues with homogeneity of variance, the variance of data points seem to decrease as our predicted values increase. There seems to be less accuracy in our model when predicted adr values are low which could be a result of the non-normality of our data. Figure 2.1 and well as Residuals vs Leverage (Figure 2) display leverage points using Cooks distance. We can see some points with large Cooks distance values due in part to non-normality of data. Taking look a multicollinearity we can use variance inflation factors, there doesn't seem to be anything out note in this case.

VIF

	GVIF	Df	GVIF^(1/(2*Df))
is_canceled	1.134484	1	1.065122
lead_time	1.421216	1	1.192148
arrival_date_month	1.489546	11	1.018277
stays_in_weekend_nights	1.224511	1	1.106576
stays_in_week_nights	1.214110	1	1.101867
adults	1.059027	1	1.029090
children	1.071874	1	1.035313
babies	1.052397	1	1.025864
total_of_special_requests	1.174494	1	1.083741

In our next model, instead of regression of all variables, we remove the children and babies variable and replace them with interaction terms, dependent on the number of adults. Doing so is simply more logical, children and babies cannot book hotel rooms without adults, their variables should be tied together. We also perform a backward regression base on AIC in order to determine the significant variables in our regression. Using AIC we can determine that the significant variables are *is_cancled, lead_time, arrival_date_month, adults, total_of_special_requests,* and *adults:children*. We can see these results largely repeated by using the R^2 to determine significant variables as well as using Mallow's cp (R^2, MCP).

R^2

```
                (Intercept)                is_canceled1                   lead_time     arrival_date_monthAugust
                       TRUE                        TRUE                        TRUE                         TRUE
   arrival_date_monthDecember  arrival_date_monthFebruary  arrival_date_monthJanuary      arrival_date_monthJuly
                      FALSE                       FALSE                       FALSE                         TRUE
       arrival_date_monthJune     arrival_date_monthMarch      arrival_date_monthMay  arrival_date_monthNovember
                       TRUE                       FALSE                        TRUE                        FALSE
    arrival_date_monthOctober arrival_date_monthSeptember     stays_in_weekend_nights       stays_in_week_nights
                      FALSE                        TRUE                       FALSE                        FALSE
                     adults    total_of_special_requests             adults:children               adults:babies
                      FALSE                       FALSE                        TRUE                        FALSE
```

MCP

```
[1] "is_canceled1"            "lead_time"              "arrival_date_monthAugust"    "arrival_date_monthJuly"
[5] "arrival_date_monthJune"  "arrival_date_monthMay"  "arrival_date_monthSeptember" "adults:children"
```

The only difference between these three tests are the presence of *total_of_special_requests*.

Using the best model as selected by AIC and backward regression, we receive the plots detailed in Figure 3. We also create a cooks distance plot in Figure 3.1 to view influential points.

Figure 3

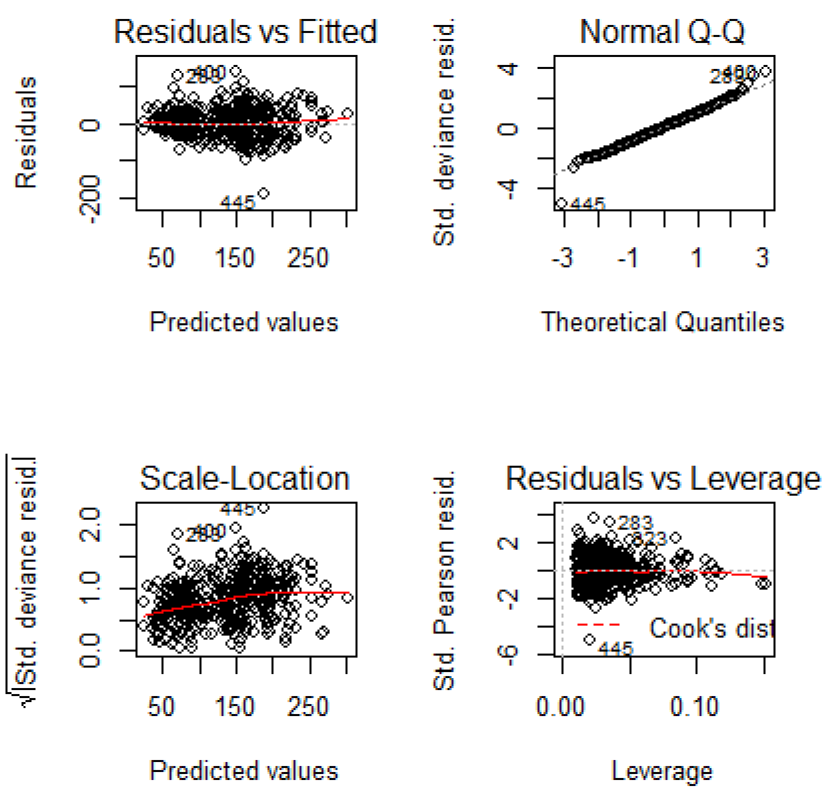
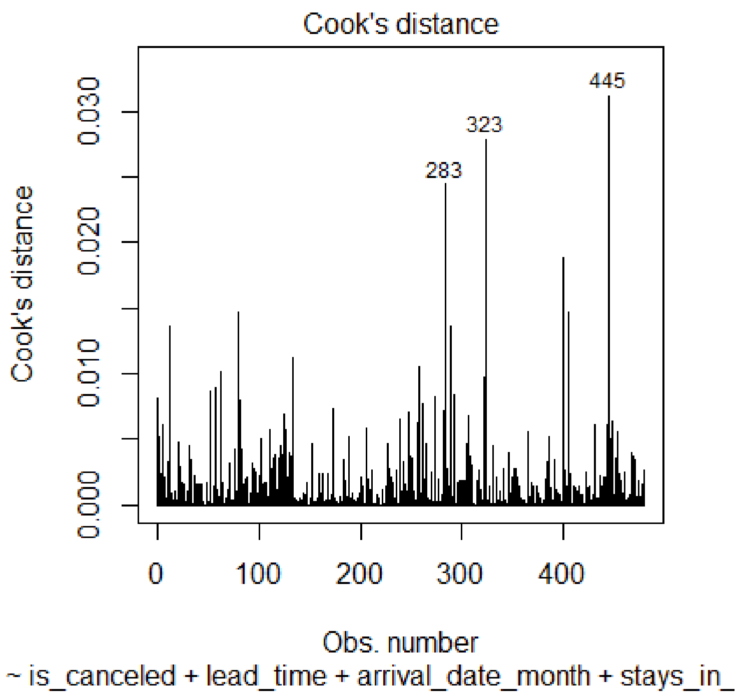


Figure 3.1



VIF2

	GVIF	Df	GVIF^(1/(2*Df))
is_canceled	1.135349	1	1.065528
lead_time	1.421561	1	1.192292
arrival_date_month	1.484931	11	1.018134
stays_in_weekend_nights	1.224226	1	1.106447
stays_in_week_nights	1.214139	1	1.101880
adults	1.065820	1	1.032386
total_of_special_requests	1.174836	1	1.083899
adults:children	1.078041	1	1.038288
adults:babies	1.052350	1	1.025841

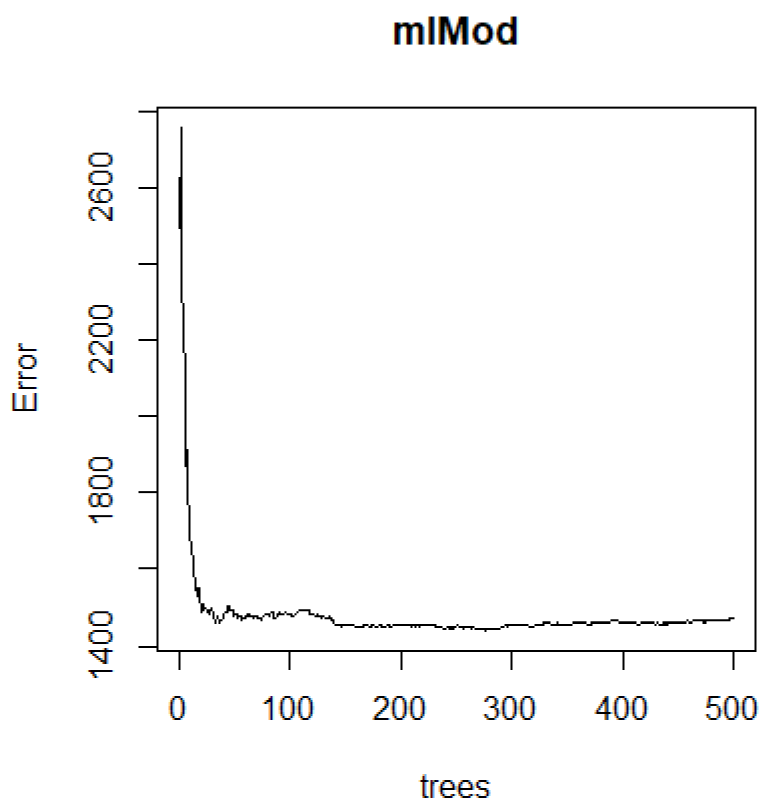
These results are largely the same as in our base model, no need to discuss them again. There also appears to be no large collinearity between variables as well (VIF2). Using this model, we grab a random sample of data from our overall data to perform a train test split prediction model. We can then calculate the mean squared error (MSE1) in order to compare this models effectiveness against others, specifically our final model, which uses Randomforest.

MSE1

```
[1] 1298.092
```

Randomforest uses supervised machines learning so that we can perform regressions accurately on a data set. We start by forming a training and testing data based off of our overall data set, which we will be forming predictions off of. Our Randomforest model does not require interaction terms as the model considers variables in sequence, so we don't need to specify interactions. This is especially true with a large enough "forest". In Figure 4, we can visualize how larger numbers of trees in a Randomforest model decreases our error.

Figure 4



To find the best model for our dataset, we run a loop using mtry values from 1 to 9. Mtry determines how many variables are used for splitting at each tree node. Since our selected data is using nine variables, we run predictions using mtry 1 to 9. This loop outputs the test MSE from each of those models with different mtry values (MSE2). Doing so allows us to find the best

Randomforest model for our data. We use a sufficiently large number of trees, as we have seen from Figure 4, different numbers of trees will affect our error.

MSE2

```
> test.err
[1] 2401.044 1698.834 1538.559 1497.924 1478.754 1482.068 1475.967 1496.270 1493.961
> #minmize MSE at mtry = 5
> min(test.err)
[1] 1475.967
```

The best model based on our train test split is a model with mtry = 5.

It can be seen that the best model using this data set is the Randomforest model. Even though the MSE of the multiple regression model with interaction terms is lower than any of the Randomforest models, the multiple regression model does not succeed as a model. This is due to the normality assumption being violated, leaving our model heavily biased. A Randomforest algorithm using a large number of decision trees should be able to approximate the best model with as little variance and bias as possible.