Analysis of Hyatt Hotels Net Promoter Score

Nate Hoffelmeyer, Kirby Hood, Joshua Shusterman, Monika Taylor

Project completed as a requirement of IST 687-37225: *Applied Data Science*
School of Information Studies, Syracuse University

Table of Contents

Description

With over 210 hotels in 43 countries, the Hyatt Corporation is a leader in the hotelier and resort industry. Hyatt places their hotels in not just major cities, but smaller cities, as well as by airports, and major vacation destinations. The Hyatt Corporation has many different brands of hotels, offering many different experiences for each guest, even if they are traveling for business or pleasure. Being one of the major hotel chains in the world, Hyatt is concerned with making sure that the guest that stays at one of their hotels is delivered a distinctive experience and will recommend their hotel to others. Hyatt's Corporations' mission statement says it the best. "Every day we care for our guests. Care is at the heart of our business, and it's this distinct guest experience that makes Hyatt one of the world's best hospitality brands." (https://about.hyatt.com/en.html)

Every time a guest stays at a Hyatt property upon departure they are asked to fill out a survey. This project will analyze those guest surveys and the likelihood that a guest would recommend a Hyatt Corporation property. Several factors will be compared, including amenities and service, in order to provide a recommendation for hotels to use in determining how to improve survey results.

Project Scope and Objective

The scope of the project centers on the locations of Hyatt hotels in the United States and the time frame in which the data set was collected. The data set had been preselected for analysis and trimmed for the current analysis. During the initial phase, a decision was made to limit the scope

to only the three states with the highest return of completed surveys. These states are: California, Florida, and Texas.

The overall objective is to make actionable recommendations from the analysis based on the likelihood to recommend and the satisfaction of the customers who stay at a Hyatt Corporation property. The project will also make sure that the recommendations that are made meet or exceed the goals of the Net Promoter Score. It will also consist of descriptive statistics, different modeling techniques to show relationships between different variables, and result in solutions for the business questions.

Business Questions

The following business questions will be answered with the analysis on the Hyatt data set.

a. What is the primary driver of the Net Promoter Score? Over the course of the project a deeper dive will be taken to determine the exact variable that drives the Net Promoter Score.

b. Is there a correlation between Net Promoter Score and Revenue? Looking at the data set for Hyatt hotels in the United States, revenue needs to be maximized as well as the Net Promoter Score. Running a correlation between the two will determine if there is a relationship between the two variables.

c. Are we able to predict if a guest would be a promoter based on the amenities of the hotel, customer service, and/or condition of the hotel? Looking at the variables that include all the amenities, we hope to see what would predict if a person was a promoter. There might also be other factors that might indicate if a person is a promoter, such as guest room condition, condition of the overall hotel and the customer service of the hotel staff.

Data Acquisition

The data used for this project was provided and pre-filtered to exclude international properties, as well as some variables. The final data file provided contained 118 variables with 3 million observations.

In order to further cut down the number of variables as well as the number of observations, making the data more manageable, a quality assessment was performed. During the assessment, it was revealed that a great number of observations had no data in several key variables. It was decided that these observations be removed from the dataset. It was also decided that the scope of the project be limited to only the three states with the highest amount of completed surveys.

Data Processing

Once the data set was loaded, a data frame was created as well as a function to execute the changing of null data to NA. Below is a sample of the code that was used to achieve this data cleansing:

```
nullToNA <- function(npsdata){
  #split df into numeric & non-numeric functions
  a <- npsdata[,sapply(df, is.numeric), drop = FALSE]
  b <- npsdata[,sapply(df, Negate(is.numeric)), drop = FALSE]
  # Change empty strings to NA
  b <- b[lapply(b,function(x) levels(x) <- c(levels(x),NA)),] #add NA level
  b <- b[lapply(b,function(x) x[x==""]<-NA),] #change Null to NA
  #Put the columns back together
  d<-cbind(a,b)
  d[,names(npsdata)]
}
```

The data was separated into numeric and non-numeric strings. All of the nulls or empty strings were then set to NA and then all columns were then recombined. NA's were then converted to the mean of each of the columns. This allows us to be able to still use the rows. Commas were then taken out of the numeric columns to ensure just the numbers existed and made this easier to work with. The top three states of California, Texas and Florida were chosen to be the states to focus on, as they are the states with the most completed surveys.

The variables that contained binary information were changed to yes being equal to 1, and no being equal to 0. The rows that did not contain any information were then changed to a zero representing no. This was done so that the information could still be used, and it was safe to say that if a column was left blank on a survey, it was a no response.

Variables

After crafting our business questions, the following variables were chosen to direct our focus.

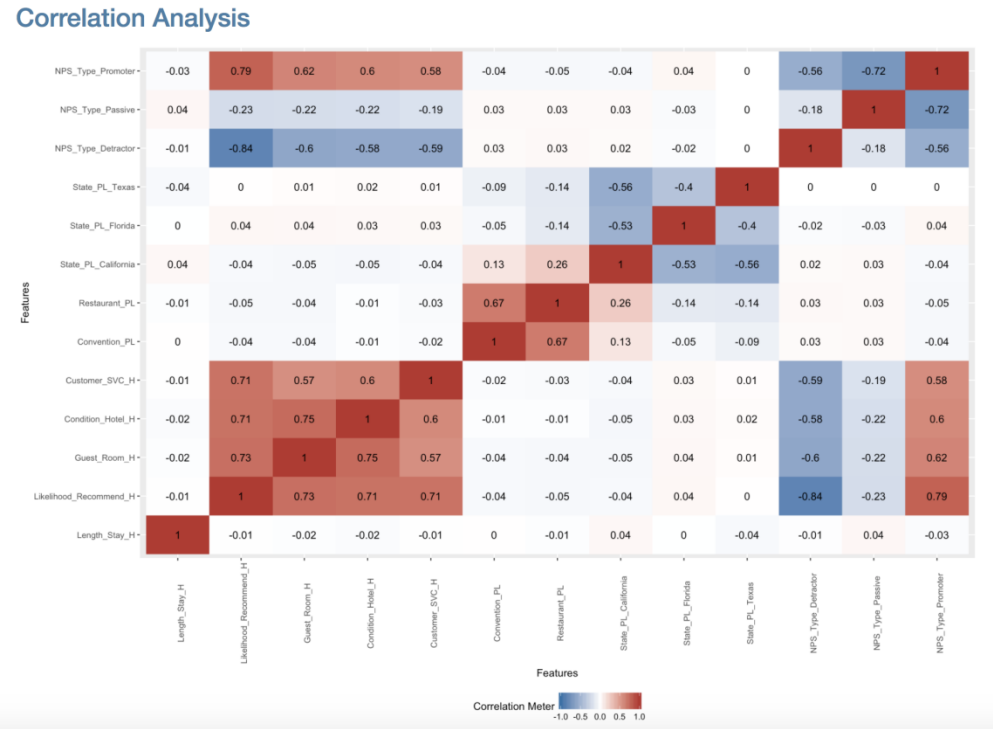
Likelihood_Recommend_H: Likelihood to recommend metric; value on a 1 to 10 scale
Guest_Room_H: Guest room satisfaction metric; value on a 1 to 10 scale
Condition_Hotel_H: Condition of hotel metric; value on a 1 to 10 scale
Customer_SVC_H: Quality of customer service metric; value on a 1 to 10 scale

State_PL: State in which the hotel is located
Convention_PL: Flag indicating if the hotel has a convention space
NPS_Type: Indicates if the guest's HySat responses mark them as a promoter, a passive, or a detractor
Restaurant_PL: Flag indicating id the hotel has a onsite restaurant
Length_Stay_H: length of stay
REVENUE_USD_R: total USD revenue

Descriptive Statistics

Simple descriptive statistics were ran to see the average length of stay per state, average likelihood to recommend score, as well as counting all the likelihood to recommend scores by state, and display a list in descending order. When we executed the code to see the scores by state, this allowed us to choose to work with only the top three states' data.

A correlation analysis was performed to give more visualization to our hypothesis. The hypothesis states that the likelihood to recommend a Hyatt Corporation property will be based on a few amenities, and how the hotel's condition is kept. The following correlation was done with the package dataExplorer. It helps to prove that there is a strong correlation between the variables Likelihood_Recommend_H, Guest_Room_H, Condition_Hotel_H and Customer_SVC_H.

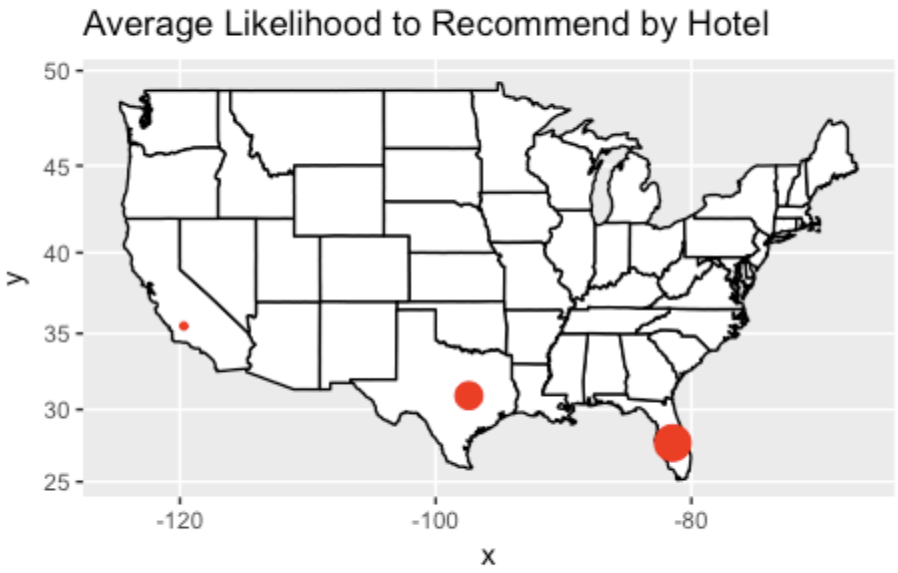


This correlation analysis shows that there is a high correlation between likelihood to recommend and guest room, condition of the hotel and customer service. This indicates that a person will more than likely recommend a Hyatt hotel if they receive wonderful customer service and the hotel is in pristine condition as well as the guest room in which they stay. The NPS type of promoter has a positive correlation with the same variables, with the highest correlation to the guest room.

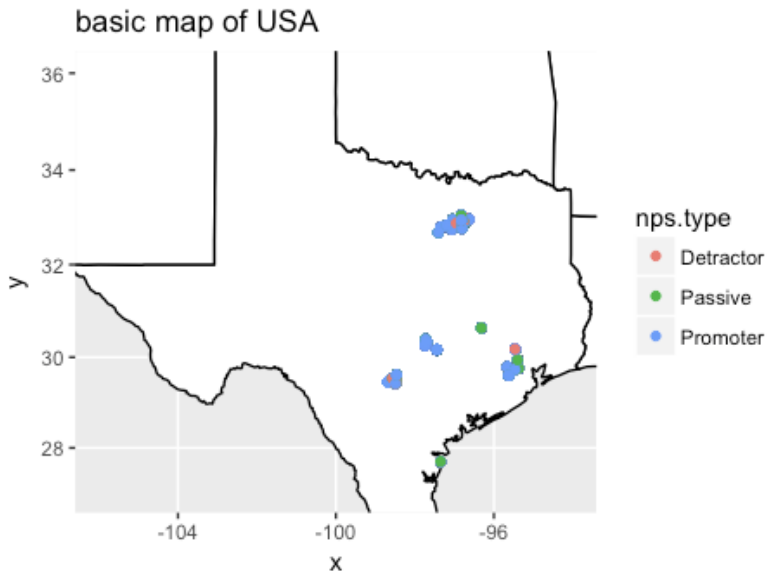
A correlation was also done to see if there was any correlation between the likelihood to recommend and revenue. It turns out that these two variables are negatively correlated and there is not a strong correlation between the two.

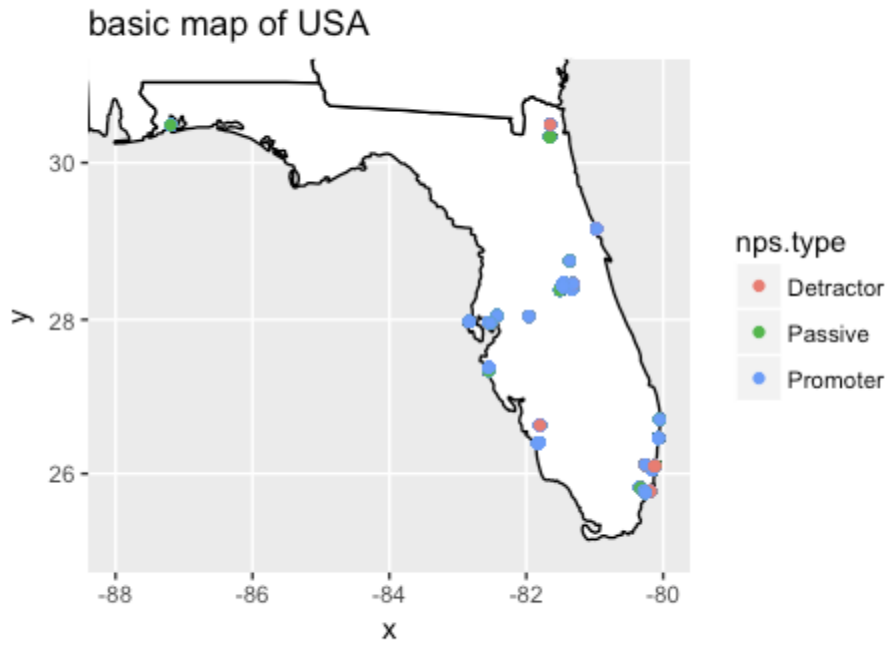
NPS Analysis

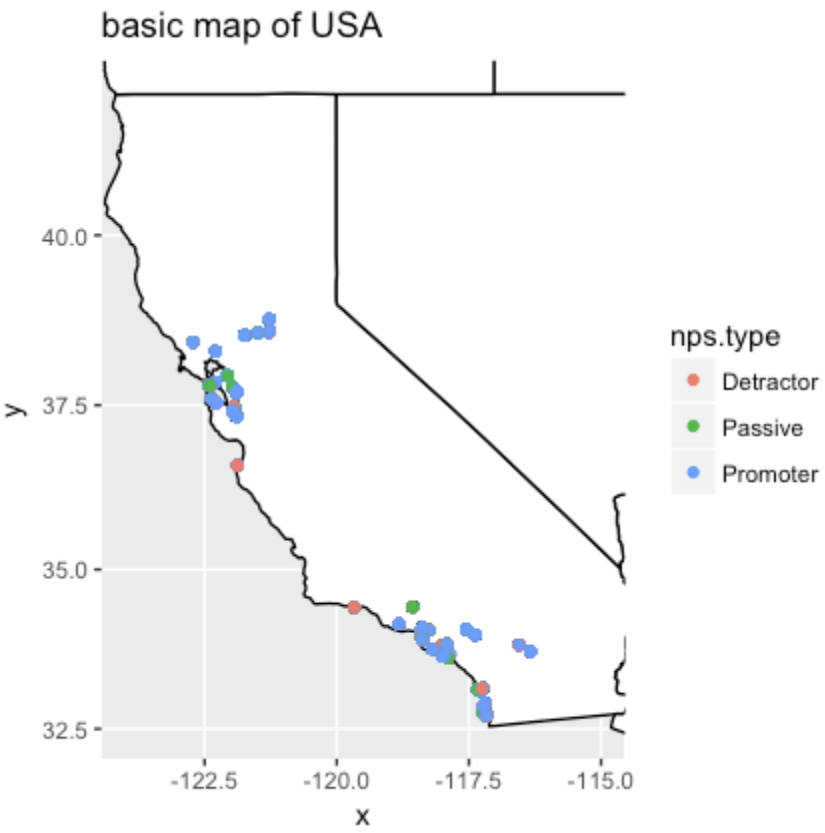
The average likelihood to recommend by State (codified by size - bigger dot, more likely to recommend) was mapped to see which state had the highest NPS. This indicates that Florida is the state with the highest average of the likelihood to recommend.



We zoomed in on each state to see where the local scores were highest.







Modeling

Several modeling techniques were used while assessing the data. We created these models in order to create visual tools that could help answer our business questions, as well as to see any patterns and predict how and which variables affect the Net Promoter Score.

❖ Linear Modeling

We tried to predict whether having a convention center or restaurant has an effect on likelihood to recommend. When the variable customer service was added, the model was better at predicting the likelihood to recommend, than other amenities. We can predict if someone is a promoter using customer service and condition of room/hotel. Amenities seem to be obsolete

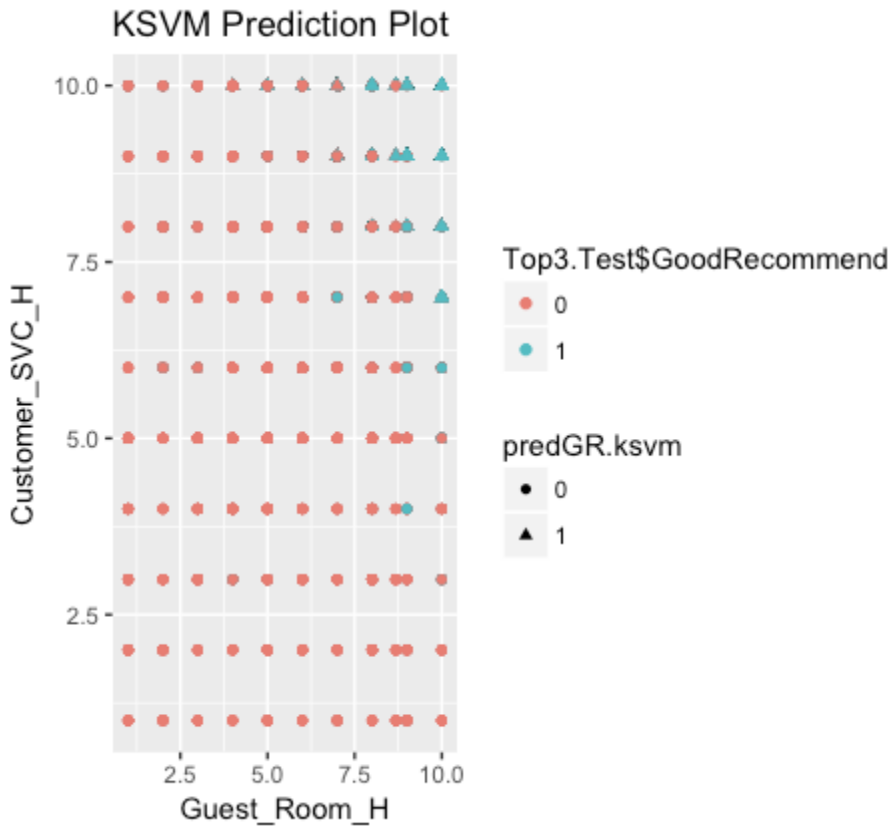
when predicting a recommendation. Overall, the linear models gave a very high level of error and did not seem a useful tool for analysis in this case.

❖ Support Vector

We ran a Naive Bayes model to predict whether or not a guest would be a promoter, as defined by if they gave a likelihood to recommend value of 9 or 10. The model using Hotel condition, customer service, and room condition was 87% accurate at predicting whether a person was a promoter. This variable combination had the highest predictive percent (even over restuarant_pl and convention_pl) and lowest RMSE (root mean square error)

We also ran a KSVM model to predict the same as the above, and it proved to be the better model with a better prediction rate and a lower RMSE

This prediction plot shows that as customer service and hotel room condition ratings increase, the likelihood of the guest being a promoter goes up.



Recommendations

Based on the analysis of the data, different models, and descriptive statistics performed, it is recommended that the hotels provide more staff training to improve customer service. Ensuring that staff members are able to handle all situations pertaining to customers' distinctive experiences is vital in the importance of the recommendation of the Hyatt Corporation hotels. It is also recommended that more housekeeping staff is hired to improve overall cleanliness of the rooms. The guest room conditions need to live up to the customers' expectations and experience.

Business Questions Revisited

a. What is the primary driver of the Net Promoter Score? The primary driver of the Net Promoter Score of Hyatt Corporation surveys, are customer service and guest room condition. These are both reflected in the recommendations to ensure better customer service training and that each of the hotels can make sure the guest room and overall hotel condition is superb.

b. Is there a correlation between Net Promoter Score and Revenue? After looking at the data, and doing a correlation analysis, there is a negative correlation and it is not very strong. This indicates that other factors drive the likelihood to recommend and determine the Net Promoter Score.

c. Are we able to predict if a guest would be a promoter based on the amenities of the hotel, customer service, and/or condition of the hotel? We were able to predict if a guest would be a promoter based on the variables customer service and condition of hotel. We also found that the condition of the guest room also played an important role in indicating if someone would be a promoter. However, we were not able to predict with high accuracy, any amenities that would factor in to indicating if a guest was a promoter.

R Code

Below is all of the R code that was used to create this project and analysis.

```
# Load a sample dataset into R
load("~/Desktop/IST 687/nps_subset.RData")

# View the dataset
View(DT_exp_final_set)

# Dataframe it for ease of use w/ the naming convention
npsdata <- data.frame(DT_exp_final_set)
npsdata[1:5,]
summary(npsdata)
str(npsdata)
ncol(npsdata) # verify number of columns in the df
nrow(npsdata) # verify number of rows in the df

#********************************************** Null to NA
**********************************************
nullToNA <- function(npsdata){
  #split df into numeric & non-numeric functions
  a <- npsdata[,sapply(df, is.numeric), drop = FALSE]
  b <- npsdata[,sapply(df, Negate(is.numeric)), drop = FALSE]
  # Change empty strings to NA
  b <- b[lapply(b,function(x) levels(x) <- c(levels(x),NA)),] #add NA level
  b <- b[lapply(b,function(x) x[x==""]<-NA),] #change Null to NA
  #Put the columns back together
  d<-cbind(a,b)
  d[,names(npsdata)]
}
nullToNA(npsdata) #run the function above on the df
is.null(npsdata) # check if there are any null values
View(npsdata) # look and see that nulls were converted to NA
nrow(npsdata) # count rows to make sure nothing was deleted
#**********************************************
npsdataCL<-npsdata[complete.cases(npsdata$Likelihood_Recommend_H),] #filter out incomplete data
for likelihood to recommend
View(npsdataCL$Likelihood_Recommend_H) #look at the likelihood to recommend data for sanity
check
nrow(npsdataCL) # see how many rows (observations) exist after filtering to only complete likelihood
cases
any(is.na(npsdataCL$Likelihood_Recommend_H)) #are there any NA remaining in likelihood to
recommend?

# count up the likelihood to recommend scores, grouped by state, and display list in descending order of
the counts (states with most count on top)
sqldf("select State_PL,count(Likelihood_Recommend_H) from npsdataCL group by State_PL order by
count(Likelihood_Recommend_H) desc")

# save the data set to a text file so others can load it into their local instance
```

```
write.table(npsdataCL, "~/Desktop/IST 687/npsdataCL.txt", sep="\t")

# take all data from the 3 states with the highest counts of complete likelihood to recommend survey scores.
npsdataCL_Top3 <- sqldf("select * from npsdataCL where State_PL = 'California' or State_PL = 'Texas' or State_PL = 'Florida'") # dataframe the new dataset
View(npsdataCL_Top3) # view the df for sanity check

# Let's make a test data set we can mess with and not worry about having to remove and redo
npsdataCL_Top3_Test <- npsdataCL_Top3

#Cleanup the columns we don't want to look at further & other stuff in the data
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-1:-3] #remove first 3 to get length of stay c as column 1
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-1:-33] #remove columns to get length stay h as col 1 (has more data)
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-2:-13] #remove columns to get likelihood recommend as 2 col
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-3] #remove overall_sat_h
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-4] #remove tranquility
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-6] #remove staff_cared_h
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-6] #remove Internet_Sat_H
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-6:-16] #remove Check_In_H through City_PL
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-7:-29] #remove US.Region_PL through Conference_PL
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-8:-9] #remove Dry.Cleaning_PL through Elevators_PL
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-9:-14] #remove Fitness.Trainer_PL through Mini.Bar_PL
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-11:-12] #remove Regency.Grand.Club_PL and Resort_PL
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-13:-21] #remove Shuttle.Service_PL through Booking_Channel

#The binary columns remaining have blanks. Let's see how many of those there are & modify as needed
#work with Convention_PL column
npsdataCL_Top3_Test$Convention_PL <- ifelse((npsdataCL_Top3_Test$Convention_PL == "Y"),1,ifelse((npsdataCL_Top3_Test$Convention_PL == "N"),2,0)) #convert char to num binary (1,0)
sqldf("select count(Convention_PL) from npsdataCL_Top3_Test where Convention_PL = 0") #shows us only 383 0 values --> let's remove them but keep the column
npsdataCL_Top3_Test <- sqldf("select * from npsdataCL_Top3_Test where Convention_PL = 1 or Convention_PL = 2") #remove zero values (run line above again once done to check)
#Work with Fitness.Center_PL
npsdataCL_Top3_Test$Fitness.Center_PL <- ifelse((npsdataCL_Top3_Test$Fitness.Center_PL == "Y"),1,ifelse((npsdataCL_Top3_Test$Fitness.Center_PL == "N"),2,0))
npsdataCL_Top3_Test$FitnessCenter_PL <- npsdataCL_Top3_Test$Fitness.Center_PL #rename the column
sqldf("select count(FitnessCenter_PL) from npsdataCL_Top3_Test where FitnessCenter_PL = 0") #shows that we have 18k missing, so let's just nix this colum
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-8] #nix the fitnesscenter column due to so much missing data
```

```
#Work with Pool.Indoor_PL
npsdataCL_Top3_Test$Pool.Indoor_PL <- ifelse((npsdataCL_Top3_Test$Pool.Indoor_PL ==
"Y"),1,ifelse((npsdataCL_Top3_Test$Pool.Indoor_PL == "N"),2,0)) #convert char to num binary (1,0)
npsdataCL_Top3_Test$PoolIndoor_PL <- npsdataCL_Top3_Test$Pool.Indoor_PL #rename the column
since the . messes with sqldf
sqldf("select count(PoolIndoor_PL) from npsdataCL_Top3_Test where PoolIndoor_PL = 0") #shows us
18k missing values --> let's nix the column
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-8] #nix the poolindoor_pl column due to so much
missing data
#Work with Pool.Outdoor_PL
npsdataCL_Top3_Test$Pool.Outdoor_PL <- ifelse((npsdataCL_Top3_Test$Pool.Outdoor_PL ==
"Y"),1,ifelse((npsdataCL_Top3_Test$Pool.Outdoor_PL == "N"),2,0)) #convert char to num binary (1,0)
npsdataCL_Top3_Test$PoolOutdoor_PL <- npsdataCL_Top3_Test$Pool.Outdoor_PL #rename the
column since the . messes with sqldf
sqldf("select count(PoolOutdoor_PL) from npsdataCL_Top3_Test where PoolOutdoor_PL = 0") #shows
us 18k missing values --> let's nix the column
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-8] #nix the pooloutdoor_pl column due to so much
missing data
#Work with Restaurant_PL
npsdataCL_Top3_Test$Restaurant_PL <- ifelse((npsdataCL_Top3_Test$Restaurant_PL ==
"Y"),1,ifelse((npsdataCL_Top3_Test$Restaurant_PL == "N"),2,0)) #convert char to num binary (1,0)
sqldf("select count(Restaurant_PL) from npsdataCL_Top3_Test where Restaurant_PL = 0") #shows us 0
missing values! --> woo hoo, let's keep it
#Work with Self.Parking_PL
npsdataCL_Top3_Test$Self.Parking_PL <- ifelse((npsdataCL_Top3_Test$Self.Parking_PL ==
"Y"),1,ifelse((npsdataCL_Top3_Test$Self.Parking_PL == "N"),2,0)) #convert char to num binary (1,0)
npsdataCL_Top3_Test$SelfParking_PL <- npsdataCL_Top3_Test$Self.Parking_PL #rename the column
since the . messes with sqldf
sqldf("select count(SelfParking_PL) from npsdataCL_Top3_Test where SelfParking_PL = 0") #shows us
18k missing values --> let's nix the column
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-9] #nix the pooloutdoor_pl column due to so much
missing data
npsdataCL_Top3_Test <- npsdataCL_Top3_Test[,-10:-13] #nix the columns we made at the end from the
renaming to remove .
# set the 2's in the binary columns = to 0 to represent "no"
npsdataCL_Top3_Test$Convention_PL <- ifelse((npsdataCL_Top3_Test$Convention_PL == 2),0,1)
npsdataCL_Top3_Test$Restaurant_PL <- ifelse((npsdataCL_Top3_Test$Restaurant_PL == 2),0,1)


# Let's get descriptive information on our variables
mean(npsdataCL_Top3$Likelihood_Recommend_H) #average likelihood to recommend
npsdataCL_Top3$Guest.NPS.Goal_PL[is.na(npsdataCL_Top3$Guest.NPS.Goal_PL)] <-
mean(npsdataCL_Top3$Guest.NPS.Goal_PL, na.rm=TRUE) #treat any nas in Guest.NPS.Goal_PL as the
mean of the distribution
sum(is.na(npsdataCL_Top3$Internet_Sat_H)) #how many nas are there for internet_sat_H
sum(is.na(npsdataCL_Top3$Guest.NPS.Goal_PL)) #are there any nas in guest nps goal?
sum(is.na(npsdataCL_Top3$Length_Stay_H)) #how many nas are in length stay hotel
sum(is.na(npsdataCL_Top3$LENGTH_OF_STAY_C)) #how many nas are in length stay c
sum(is.na(npsdataCL_Top3$Likelihood_Recommend_H)) #how many nas are in likelihood to
recommend
sum(is.na(npsdataCL_Top3$Guest_Room_H)) #how many nas are in guest_room_h (1043)
```

```
sum(is.na(npsdataCL_Top3$Condition_Hotel_H)) #how many nas in hotel condition scale (1235)
sum(is.na(npsdataCL_Top3$Customer_SVC_H)) #how many nas in sust svc scale (1568)
sum(is.na(npsdataCL_Top3$Internet_Sat_H)) #how many nas in internet satisfaction (56095) --> nix this column
sum(is.na(npsdataCL_Top3$Convention_PL))


######### Descriptive visuals & stats on our data ###########
g0 <- ggplot(npsdataCL_Top3_Test, aes(x=factor(State_PL), y=Length_Stay_H)) +
stat_summary(fun.y="mean", geom="bar") #average LOS by state
g0 + ggtitle("Average LOS by State")
# list of average length of stay by state
sqldf("select State_PL,avg(Length_Stay_H) from npsdataCL_Top3_Test group by State_Pl order by avg(Length_Stay_H) desc")

g <- ggplot(npsdataCL_Top3_Test,aes(x=Likelihood_Recommend_H)) + geom_histogram(binwidth=0.5, color="blue", fill="orange") #histogram of likelihood to recommend
g + ggtitle("Frequency of Likelihood to Recommend Score") # hisogram of Likelihood_Recommend_H
mean(npsdataCL_Top3_Test$Likelihood_Recommend_H) #mean is 8.626665

g1 <- ggplot(npsdataCL_Top3, aes(x=factor(State_PL), y=Likelihood_Recommend_H)) +
stat_summary(fun.y="mean", geom="bar") #average likelihood to recommend across the states
g1 #chart of average likelihood to recommend by state
# list of average likelihood to recommend by state
sqldf("select State_PL,avg(Likelihood_Recommend_H) from npsdataCL_Top3_Test group by State_Pl order by avg(Likelihood_Recommend_H) desc")

# --> looks like LOS and likelihood to recommend are correlated. let's check it out with a scatterplot
g2 <- ggplot(npsdataCL_Top3_Test,aes(x=Likelihood_Recommend_H,y=Length_Stay_H)) +
geom_point(aes(color=State_PL))
g2

ggplot(npsdataCL_Top3_Test, aes(x=Length_Stay_H, y=Likelihood_Recommend_H)) +
stat_summary(fun.y="mean", geom="line")

#visuals for guest room rating
g3 <- ggplot(npsdataCL_Top3_Test,aes(x=Guest_Room_H)) + geom_histogram(binwidth=0.5, color="blue", fill="orange") #histogram of guest room quality rating
g3 + ggtitle("Frequency of Guest Room Rating") # hisogram of Guest_Room_H
any(is.na(npsdataCL_Top3_Test$Guest_Room_H))
#looks like there are NAs. Clean them up, and then re-run the above
npsdataCL_Top3_Test$Guest_Room_H[is.na(npsdataCL_Top3_Test$Guest_Room_H)] <- mean(npsdataCL_Top3_Test$Guest_Room_H, na.rm=TRUE) #treat any nas in Guest_Room_H as the mean of the distribution
mean(npsdataCL_Top3_Test$Guest_Room_H) #mean is 8.682414

#scatterplot of guest room recommendation and likelihood to recommend to get visual of correlation
g4 <- ggplot(npsdataCL_Top3_Test,aes(x=Likelihood_Recommend_H,y=Guest_Room_H)) +
geom_point(aes(color=NPS_Type))
g4 # chart shows
```

```
#visuals for hotel condition
any(is.na(npsdataCL_Top3_Test$Condition_Hotel_H)) #check for nas
npsdataCL_Top3_Test$Condition_Hotel_H[is.na(npsdataCL_Top3_Test$Condition_Hotel_H)] <-
mean(npsdataCL_Top3_Test$Condition_Hotel_H, na.rm=TRUE) #treat any nas in Condition_Hotel_H as
the mean of the distribution
g5 <- ggplot(npsdataCL_Top3_Test,aes(x=Condition_Hotel_H)) + geom_histogram(binwidth=0.5,
color="blue", fill="orange") #histogram of hotel condition quality rating
g5 + ggtitle("Distribution of Hotel Quality Ratings")
ggplot(npsdataCL_Top3_Test, aes(x=Condition_Hotel_H, y=Likelihood_Recommend_H)) +
stat_summary(fun.y="mean", geom="bar")


#scatterplot of hotel quality rating and likelihood to recommend
g6 <- ggplot(npsdataCL_Top3_Test,aes(x=Condition_Hotel_H,y=Likelihood_Recommend_H)) +
geom_point(aes(color=NPS_Type))
g6

#visuals for customer service ratings
any(is.na(npsdataCL_Top3_Test$Customer_SVC_H)) #check for NA data
sum(is.na(npsdataCL_Top3_Test$Customer_SVC_H)) #see how many NA data
npsdataCL_Top3_Test<-na.omit(npsdataCL_Top3_Test)  # omit the NA's, store in a new df, review the
impact
g7 <- ggplot(npsdataCL_Top3_Test,aes(x=Customer_SVC_H)) + geom_histogram(binwidth=0.5,
color="blue", fill="orange") #histogram of hotel condition quality rating
g7
ggplot(npsdataCL_Top3_Test, aes(x=Customer_SVC_H, y=Likelihood_Recommend_H)) +
stat_summary(fun.y="mean", geom="bar")

#scatterplot to see correlation between customer service and likelihood to recommend
g8 <- ggplot(npsdataCL_Top3_Test,aes(x=Customer_SVC_H,y=Likelihood_Recommend_H)) +
geom_point(aes(color=NPS_Type))
g8

#frequency distribution of binary convention center attached or not
g9 <- ggplot(npsdataCL_Top3_Test,aes(x=Convention_PL)) + geom_histogram(binwidth = 0.5,
color="blue", fill="orange") #histogram of hotel condition quality rating
g9

#npstype by convention
g10 <- ggplot(npsdataCL_Top3_Test,aes(x=NPS_Type,fill=factor(Convention_PL))) +
geom_bar(position="dodge")
g10

#frequency distribution of restaurant on premise
g11 <- ggplot(npsdataCL_Top3_Test,aes(x=Restaurant_PL)) + geom_histogram(binwidth = 0.5, color =
"blue", fill = "orange")
g11

#show count of nps type by restaurant or not
ggplot(npsdataCL_Top3_Test,aes(x=NPS_Type,fill=factor(Restaurant_PL))) +
geom_bar(position="dodge")
```

```
# chart of restuarant by average likelihood to recommend
ggplot(npsdataCL_Top3_Test, aes(x=factor(Restaurant_PL), y=Likelihood_Recommend_H)) +
stat_summary(fun.y="mean", geom="bar")

# average likelihood to recommend with convention or not
ggplot(npsdataCL_Top3_Test, aes(x=factor(Convention_PL), y=Likelihood_Recommend_H)) +
stat_summary(fun.y="mean", geom="bar")

# average likelihood to recommend based on hotel condition rating
ggplot(npsdataCL_Top3_Test, aes(x=Condition_Hotel_H, y=Likelihood_Recommend_H)) +
stat_summary(fun.y="mean", geom="bar")

# look at promoter type by state counted by nps_type
sqldf("select State_PL,NPS_Type,count(NPS_Type) from npsdataCL_Top3 group by
State_PL,NPS_Type order by count(NPS_Type) desc")


# save the data set to a text file so others it can be shared in group drive
write.table(npsdataCL_Top3_Test, "~/Desktop/IST 687/npsdataCL_Top3_Test.txt", sep="\t")

##### Add some more map visuals
sd <- read.xls("~/Desktop/IST 687/npsdataCL_Top3_maps.xlsx")
sd <- sd[,-1]
str(sd)
head(sd)

colnames(sd) <- c("likelihood.recommend","city","state","lat","lon","brand","nps.type")

sd$city <- tolower(sd$city)
sd$state <- tolower(sd$state)

avgRec <- aggregate(sd, by=list(sd$lalo),FUN=mean,na.rm=T)
str(avgRec)
head(avgRec)

#map of average likelihood to recommend by state
m <- map.simple + geom_point(data = avgRec,aes(x=avgRec$lon,y=avgRec$lat,
color=likelihood.recommend), show.legend = F)
m <- m + ggtitle("Average Likelihood to Recommend by Hotel")
m

#map with every survey point plotted
m1 <- map.simple + geom_point(data = sd,aes(x=sd$lon,y=sd$lat, color=nps.type), show.legend = T)
m1 + ggtitle("Hotel NPS Survey")

#zoom in on California - all survey plot
mzoom.ca <- m1 + xlim(-124,-115) + ylim(32.5,42) + coord_map()
mzoom.ca

#zoom in on Texas - all survey plot
```

```
mzoom.tx <- m1 + xlim(-106,-94) + ylim(27,36) + coord_map()
mzoom.tx

#zoom in on Florida - all survey plot
mzoom.fl <- m1 + xlim(-88,-80) + ylim(25,31) + coord_map()
mzoom.fl


##########################
#Support Vector Machine
EnsurePackage('gdata')
df.test <- read.xls("~/Desktop/IST 687/npsdataCL_Top3_Test.xlsx")
packages=c("kernlab","e1071","gridExtra","ggplot2", "caret")

#use this function to check if each package is on the local machine
#if a package is installed, it will be loaded
#if any are not, the missing package(s) will be installed and loaded
package.check <- lapply(packages, FUN = function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
})
#
#
dim(df.test) #looking to see how many observations (row) are in the data frame to gain insight of where
the cutpoints should be
#
#creating a random index
RandomIndex <- sample(1:dim(df.test)[1])
CutPoint2_3 <- floor(2*dim(df.test)[1]/3)
#
#creating a training data set
Top3.Train <- df.test[RandomIndex[1:CutPoint2_3],]
head(Top3.Train)
#creating a test data set
Top3.Test <- df.test[RandomIndex[(CutPoint2_3+1):dim(df.test)[1]],]
head(Top3.Test)
#
#
#build a model using ksvm to predict Likelihood to Recommend
ksvm.model <- ksvm(Likelihood_Recommend_H ~ NPS_Type + Condition_Hotel_H, data = Top3.Train)
#model created on all variables
summary(ksvm.model)
ksvm.Predict <- predict(ksvm.model, Top3.Test) #predicting and testing
str(ksvm.Predict)
summary(ksvm.Predict)
table(ksvm.Predict,Top3.Test$Likelihood_Recommend_H)
#
#creation of a compairison data frame
compTable1 <- data.frame(Top3.Test[,1], ksvm.Predict[,1])
```

```
colnames(compTable1) <- c("test", "predict")
compTable1
#
#compute the Root Mean Squared Error
sqrt(mean(compTable1$test-compTable1$predict)^2)
#
#computing the absolute error
compTable1$error <- abs(compTable1$test - compTable1$predict)
#
#new data frame created to contain error, temp, and wind
ksvm.plot <- data.frame(compTable1$error, Top3.Test$NPS_Type, Top3.Test$Condition_Hotel_H,
Top3.Test$Likelihood_Recommend_H)
colnames(ksvm.plot) <- c("error", "NPS_Type", "Hotel_Condition","Likelihood_Recommend")
ksvm.plot
#
#plot the results using a scatter plot with the x axis represented by NPS Type, and y axis represented by
Hotel Condiditon, the point size and color represent
#the error, as defined by the actual values of Likelihood to Recommend minus the predicted values.
ksvm.sctr <- ggplot(ksvm.plot, aes(x = Likelihood_Recommend, y = Hotel_Condition)) +
geom_point(aes(size = error, color = error)) + ggtitle("KSVM Scatter Plot")
ksvm.sctr

#
###################################
#Calculate the average Likelihood to Recommend
RecommendMean <- mean(df.test$Likelihood_Recommend_H, na.rm = TRUE)
RecommendMean
#
#create a new variable named "Good Recommend" in the training data set
#GoodRecommend = 0 if the Likelihood to Recommend is below average
#GoodRecommend = 1 if the Likelihood to Recommend is equal or above the average Likelihood to
Recommend
Top3.Train$GoodRecommend <- ifelse(Top3.Train$Likelihood_Recommend_H<RecommendMean,0,1)
str(Top3.Train)
head(Top3.Train)
#
#same thing is done for the test data set
Top3.Test$GoodRecommend <- ifelse(Top3.Test$Likelihood_Recommend_H<RecommendMean,0,1)
str(Top3.Test)
tail(Top3.Test)
#
#
############### See if a different model is better at predicting Likelihood to Recommend
#
#convert GoodRecommend from num to factor
Top3.Train$GoodRecommend <- as.factor(Top3.Train$GoodRecommend)
str(Top3.Train)
Top3.Test$GoodRecommend <- as.factor(Top3.Test$GoodRecommend)
str(Top3.Test)
#
#building a ksvm model and using NPS Type and Condition of hotel variables to predict
```

```
Good.ksvm <- ksvm(GoodRecommend ~ NPS_Type + Condition_Hotel_H, data = Top3.Train)
Good.predict <- predict(Good.ksvm, Top3.Test)
Good.predict
table(Good.predict,Top3.Test$Likelihood_Recommend_H)
#
#******data frame created to contain the exact Good Recommend values and the predicted Good
Recommend values
goodComp1 <- data.frame(Top3.Test[,10], Good.predict)
colnames(goodComp1) <- c("test", "predicted") #changing column names
goodComp1
#
#computing the percentage of GoodOzone that was correctly predicted
perc_ksvm_good <- length(which(goodComp1$test==goodComp1$predicted))/dim(goodComp1)[1]
perc_ksvm_good
#
#confusion matrix created
results <- table(goodComp1$test, goodComp1$predicted)
results
#
#accuracy of KSVM model
totTest <- nrow(Top3.Test)
PercentCorrectKSVM <- round((totalCorrectKSVM/totTest)*100,1)
PercentCorrectKSVM
#
#plot the results, x axis to represent the NPS Type, y axsi to represent the Hotel Condition, shape
represents what is predicted (good or bad recommendation)
#color representing the actual value of GoodRecommend, size representing if the prediction was correct
goodComp1$correct <- ifelse(goodComp1$test==goodComp1$predicted, "correct", "wrong")
#
#new data frame created to contain correct, NPS Type, Hotel Condition, GoodRecommend
plot_good_ksvm <- data.frame(goodComp1$correct, Top3.Test$NPS_Type,
Top3.Test$Condition_Hotel_H, Top3.Test$GoodRecommend, goodComp1$predicted)
colnames(plot_good_ksvm) <- c("Correct", "NPS Type", "Hotel Condition", "GoodRecommend",
"Predicted")
str(plot_good_ksvm)
#
#plot the results, x axis to represent the NPS Type, y axsi to represent the Hotel Condition, shape
represents what is predicted (good or bad recommendation)
#color representing the actual value of GoodRecommend, size representing if the prediction was correct
Plot.ksvm <- ggplot(plot_good_ksvm, aes(x = NPS_Type, y = Condition_Hotel_H)) +
geom_point(aes(size = Correct, color = GoodRecommend, shape = Predicted)) + ggtitle("Good/Bad
Recommending -KSVM")
Plot.ksvm

# More prediction models

head(Top3.Train)
# Kernel model to predict Likelihood to Recommend
model.ksvm <- ksvm(Likelihood_Recommend_H ~ Condition_Hotel_H + Customer_SVC_H +
Guest_Room_H + Length_Stay_H, data=Top3.Train)
model.ksvm
```

```
pred.ksvm <- predict(model.ksvm,Top3.Test)
table(pred.ksvm,Top3.Test$Likelihood_Recommend_H)
str(pred.ksvm)

compTable <- data.frame(Top3.Test[,1], pred.ksvm[,1])
colnames(compTable) <- c("test","Pred")
sqrt(mean((compTable$test-compTable$Pred)^2)) # if error is high run again?

compTable$error <- abs(compTable$test - compTable$Pred)
ksvmPlot <- data.frame(compTable$error, Top3.Test$Condition_Hotel_H,
Top3.Test$Customer_SVC_H, Top3.Test$Guest_Room_H, Top3.Test$Length_Stay_H)
colnames(ksvmPlot) <-
c("error","Hotel_Condition","Customer_Service","Room_Condition","Length_Stay")
ggplot(ksvmPlot, aes(x=Length_Stay,y=Customer_Service)) + geom_point(aes(size=error, color=error))
+ ggtitle("KSVM Scatter Plot")
ggplot(ksvmPlot, aes(x=Length_Stay,y=Hotel_Condition)) + geom_point(aes(size=error, color=error)) +
ggtitle("KSVM Scatter Plot")

# try the model predicting likelihood to recommend by convention_pl and restaurant_pl
model.ksvm <- ksvm(Likelihood_Recommend_H ~ Convention_PL + Restaurant_PL, data=Top3.Train)
model.ksvm
pred.ksvm <- predict(model.ksvm,Top3.Test)
table(pred.ksvm,Top3.Test$Likelihood_Recommend_H)
str(pred.ksvm)

compTable <- data.frame(Top3.Test[,1], pred.ksvm[,1])
colnames(compTable) <- c("test","Pred")
sqrt(mean((compTable$test-compTable$Pred)^2)) # if error is high run again?

compTable$error <- abs(compTable$test - compTable$Pred)
ksvmPlot <- data.frame(compTable$error, Top3.Test$Condition_Hotel_H,
Top3.Test$Customer_SVC_H, Top3.Test$Guest_Room_H, Top3.Test$Length_Stay_H)
colnames(ksvmPlot) <-
c("error","Hotel_Condition","Customer_Service","Room_Condition","Length_Stay")
ggplot(ksvmPlot, aes(x=Length_Stay,y=Customer_Service)) + geom_point(aes(size=error, color=error))
+ ggtitle("KSVM Scatter Plot")
ggplot(ksvmPlot, aes(x=Length_Stay,y=Hotel_Condition)) + geom_point(aes(size=error, color=error)) +
ggtitle("KSVM Scatter Plot")


# use linear modeling
model.lm <- lm(formula = Likelihood_Recommend_H ~ Convention_PL, data = Top3.Train)
summary(model.lm)
pred.lm <- predict(model.lm, Top3.Test, type="response")
table(pred.lm,Top3.Test$Likelihood_Recommend_H)

compTable <- data.frame(Top3.Test[,1], pred.lm)
colnames(compTable) <- c("test","Pred")
sqrt(mean((compTable$test-compTable$Pred)^2))

# model using naive bayes
```

```
head(Top3.Test)
m.nb <- naiveBayes(as.factor(GoodRecommend) ~ Customer_SVC_H + Guest_Room_H +
Condition_Hotel_H, Top3.Train)
predgoodrec.nb <- predict(m.nb, Top3.Test)
str(predgoodrec.nb)
summary(predgoodrec.nb)
results.grnb <- table(predgoodrec.nb,Top3.Test$GoodRecommend)
results.grnb

# totintest <- nrow(Top3.Test)

totCorrect.grnb <- results.grnb[1,1] + results.grnb[2,2]
totCorrect.grnb
pctCorrect.grnb <- round((totCorrect.grnb/totintest)*100,1)
paste(pctCorrect.grnb,"%",sep="")

err.grnb <- as.numeric(Top3.Test$GoodRecommend) - as.numeric(predgoodrec.nb)
sqrt(mean(err.grnb^2))

g.grnb <- ggplot(data = Top3.Test, aes(x=Guest_Room_H,y=Customer_SVC_H)) +
geom_point(aes(shape=predgoodrec.nb, col = Top3.Test$GoodRecommend))
g.grnb

# model goodrec using svm
m.svm <- svm(GoodRecommend ~ Customer_SVC_H + Guest_Room_H + Condition_Hotel_H,
Top3.Train)
predGR.svm <- predict(m.svm, Top3.Test)
summary(predGR.svm)
str(predGR.svm)
results.grsvm <- table(predGR.svm,Top3.Test$GoodRecommend)
results.grsvm

totCorrect.grsvm <- results.grsvm[1,1] + results.grsvm[2,2]
totCorrect.grsvm
pctCorrect.grsvm <- round((totCorrect.grsvm/totintest)*100,1)
paste(pctCorrect.grsvm,"%",sep="")

err.grsvm <- as.numeric(Top3.Test$GoodRecommend) - as.numeric(predGR.svm)
sqrt(mean(err.grsvm^2))

g.grsvm <- ggplot(data = Top3.Test, aes(x=Guest_Room_H,y=Customer_SVC_H)) +
geom_point(aes(shape=predGR.svm, col = Top3.Test$GoodRecommend))
g.grksvm + ggtitle("SVM Prediction Plot")

# model using ksvm
m.ksvm <- ksvm(GoodRecommend ~ Customer_SVC_H + Guest_Room_H + Condition_Hotel_H,
Top3.Train)
predGR.ksvm <- predict(m.ksvm, Top3.Test)
summary(predGR.ksvm)
str(predGR.ksvm)
results.grksvm <- table(predGR.ksvm,Top3.Test$GoodRecommend)
```

```
results.grksvm

totCorrect.grksvm <- results.grksvm[1,1] + results.grksvm[2,2]
totCorrect.grksvm
pctCorrect.grksvm <- round((totCorrect.grksvm/totintest)*100,1)
paste(pctCorrect.grksvm,"%",sep="")

err.grksvm <- as.numeric(Top3.Test$GoodRecommend) - as.numeric(predGR.ksvm)
sqrt(mean(err.grksvm^2))

g.grksvm <- ggplot(data = Top3.Test, aes(x=Guest_Room_H,y=Customer_SVC_H)) +
geom_point(aes(shape=predGR.ksvm, col = Top3.Test$GoodRecommend))
g.grksvm + ggtitle("KSVM Prediction Plot")

##correlation between NPS type and revenue
x <- npsdataCL$Likelihood_Recommend_H
y <- npsdataCL$REVENUE_USD_R
cor(x,y)
```