Predicting Factors Influencing The Booking Of A Hotel

Harry Chen, Marsela Kapetanovic, Yimei Tang, Ruoxi Wang, Yuedi Wang
Dr. John McDonald

CSC 424
6/4/2017

Table of Contents

Non-Technical Report

Today's modern world provides consumers with a wide range of features to consider when it comes to hospitality. Consumers have many choices to choose from when booking hotels, making the competition in modern hospitality industry quite fierce. To determine whether Expedia is a better choice for customers, we conducted some analyses to identify the most significant feature that contributes to obtaining realistic predictions on whether a hotel will be booked. The results obtained could provide insights to travel agencies who may be providing consumers with similar offerings to Expedia.

The original data set is obtained from Kaggle.com and consists of a representative sample of 99,917,530 hotels. Also, 52 features are included, such as: price, user history, competitors, etc.

The following sections provide an outline of how we approached the data to achieve the ultimate goal of identifying the feature that contributes the most in whether a hotel will be booked, what we identified, and the conclusion.

Data exploration

Real-world data mainly is not clean, meaning that some values are missing or recorded incorrectly. Only three variables in our data set were complete across all observations, while all others were affected by missing values. The missing values plot shown in Fig.12 illustrates this finding. To combat this issue, we removed all of the "dirty" data by excluding all rows (hotels) that had missing values from the analysis.

After having cleaned the data set, we were left with 1,535 people in this dataset. They have conducted 1,535 searches and Expedia has provided 20,000 hotels in total for their considerations. Of the 1,535 people, 547 of them eventually booked a hotel from the recommendation results. 888 hotels out of the 20,000 search results were clicked. Therefore, after conducting some calculations, the following was identified: the booking rate turns out to be 2.74%, the click-through-rate is 4.44%, and the conversion rate is 61.6 %.

Data analysis and conclusion

Almost ten algorithms were selected to work on the data set. Below is a brief introduction of some of them that provided satisfactory results which were interpretable.

First, we tried to simplify and condense the data set by removing unnecessary variables. In order to reduce the complexity, dimension reduction was conducted using principal component analysis (PCA). This approach left us with only 4 components for numeric variables, but we also kept 6 ordinal variables.

Next, we used logistic regression. This statistical tecnique revealed that when the star rating goes from one to four, the odds of booking increases from 0.11 to 0.36. It is important to note that five-star hotels differ from the other ratings and this may be due to the fact that those hotels are more expensive, so the customer decides not to book. Another conclusion that can be drawn is that people tend to book a hotel that is currently being promoted and is part of a chain.

After that, we continued to explore the data with the K-Nearest Neighbor (KNN) method, which is a simple algorithm that stores all variables/cases and then classifies new cases based on a similarity measure.

The model was fit with the top 20 most important variables across all and this caused the model to significantly improve in comparison to the previous models. For practicality reasons, Expedia should increase the clicking rate first, and then use the model that has the highest recall score in order to

increase the booking rate of the hotels. However, under a statistics setting, the models that have the highest f1 score might be preferred.

Finally, to achieve the best possible results, we also used a method called multidimensional scaling (MDS) and discovered that the historical price of the hotel and visitor history are important factors in the booking of a hotel.

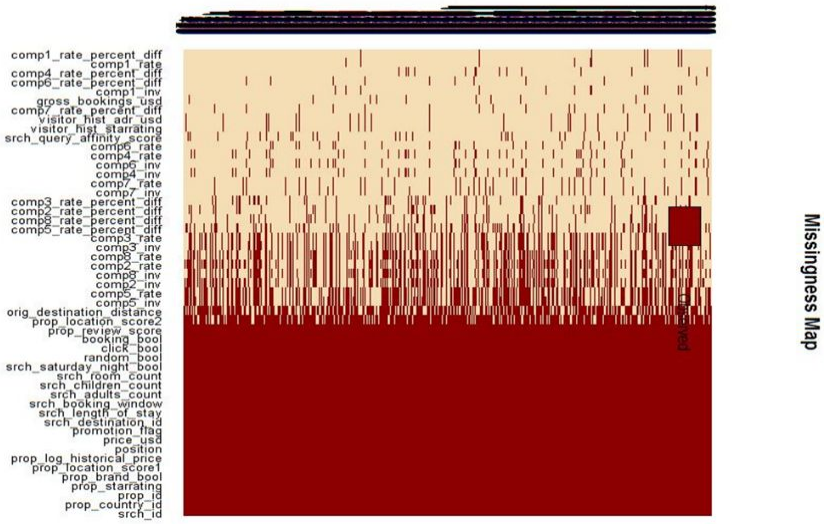


Fig.1: Missing Value Plot

Technical Report

Abstract

Booking a hotel can depend on multiple factors. With numerous features available in today's competitive hospitality industry, the goal of our analysis was to identify the most significant feature that contributes to obtaining realistic predictions of whether a hotel will be booked.

Multiple techniques were used to reach this goal, including: principal component analysis (PCA), common factor analysis (CFA), logitstic regression, linear discriminant analysis, k nearest neighbor (KNN), Decision Tree and Random Forest.

The analysis revealed that 'HotelViewStar' has the highest influence on whether a hotel will be booked and positioning (the hotel's listing position on the search results page - a page will list 50 hotels) has the highest influence on whether a hotel will be clicked. The following sections of the report support these findings.

Introduction

The decision of booking a hotel depends on multiple considerations on the consumer's behalf. To determine whether a deal is good or not, various characteristics and features need to be analyzed. A thorough analysis was conducted to determine what effects of the features from the data set had on the booking of a hotel, helping us identify the key influencer.

Methodology

The data was obtained from a study consisting of a representative sample of 99,917,530 rows. The original data set can be found on Kaggle.com, published in 2013. The data set includes hotel 52 features, some of which are: prices, ratings, room count, length of stay, and many other special features.

The analysis required multiple revisions and edits for the optimal model to be created. To reach the goal that was set, the following techniques were explored: conducting dimension reduction, transforming the binary dependent variable using logistic regression, linear discriminant analysis, and the creation of decision trees.

To check the validity of the model, validation with a 25% test set was applied to assess the predictive performance of the final model.

Data Exploration

The original data set consisted of approximately 99 million observations and 52 variables, of which 7 were nominal (identification numbers), 16 numeric (scores), 28 ordinal (Boolean), and 1 time variable. All numeric variables were normally distributed after the log transformation. An example of a histogram illustrating the distribution is shown in Fig.11.

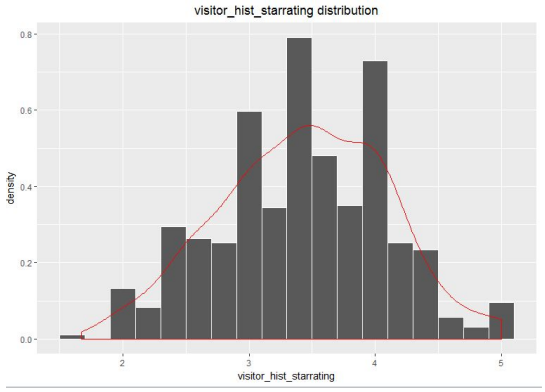


Fig.11: vistitor_hist_starrating distribution

Before we could start with any analysis, major cleaning of the data set was necessary. Only half of all variables were complete across all observations, while all others were affected by missing values. The missing values plot shown in Fig.1 in non-technical illustartes this finding. To combat this issue, the MICE function in R Studio and different bin methods were utilized. This approach produced a data set consisting of 127,359 observations and 20 variables. Because we still had a relatively large data set, we first condensed it even further by removing all ID nominal variables (they don't contribute to the goal of our analysis – leaving us with 44 variables), followed by removing competitor variables (leaving us with 20 variables because we have 8 competitors, multiplied by 3 variables each). Competitor variables were removed due to the fact that they all had missing values for the same variables, illustrated in Fig.13. After that, we conducted dimension reduction using the PCA and CFA that left us with 10 variables that were used for logistic regression and LDA. In a separate study, we used the first 20,000 rows and 52 independent variables in predicting the clicking outcome.

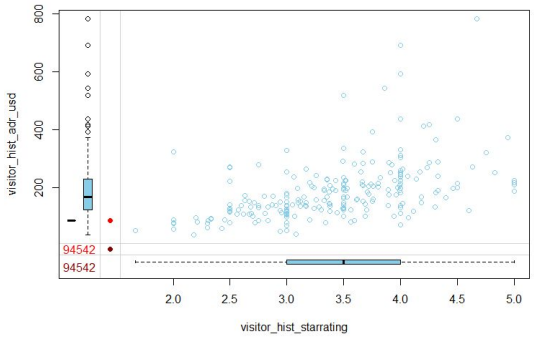


Fig.12: Competitor missing detect

Statistical Techniques

Principal Component Analysis & Common Factor Analysis

Due to our data set being significantly large, it was important to avoid data with large variances that dominate the components - so we first scaled the data. Our data set included numeric and ordinal data, requiring different methods to be used to identify the correlations between variables.

First, the Pearson matrix was used to determine the correlations between numeric variables. Nearly all of the variables turned out to be independent. There was only one correlation between 'Number of Adult' (NAdult) and 'Number of rooms' (NRoom) that was a bit higher (0.50).

After that, the Spearman and Polychoric matrices were used to identify the correlations between ordinal variables. Based on the Spearman output, there was only one higher correlation between 'Hotel Review Score' and 'Hotel Star' (0.48), while other variables were independent. Similarly, the Polychoric output showed only one higher correlation that was between 'Hotel Review Score' and 'Hotel Star' (0.53).

Then the polyserial matrix was used to explore the mix correlation, Only 'Hotel Purchase Price' (H_PurPrice) and 'Hotel Purchase Star' (H_PurStar) had a 0.57 correlation value, while the other variables were independent.

The results of this approach lead us to conclude that our data set is not suitable for using PCA and CFA. However, (due to this being a class setting and us wanting to match the class requirements) we decided to use PCA and CFA to explore the results that it would provide us with. PCA was used to analyze the numeric data and CFA was used to analyze ordinal data.

```
Loadings:
           RC2    RC1    RC3    RC4
NAdult      0.856
NRoom       0.861
Location1          0.796
Location2          0.859
StayLength                0.591
HStay                     0.615
NKid                      0.567
H_PurPrice                       0.544
LogHotelP                        0.678
DisplayP                         0.607
CompePD

                 RC2   RC1   RC3   RC4
SS loadings    1.511 1.511 1.272 1.214
Proportion Var 0.137 0.137 0.116 0.110
Cumulative Var 0.137 0.275 0.390 0.501
```

```
Loadings:
                  Factor1 Factor2 Factor3 Factor4
H_PurStar          0.342           -0.201
HotelStar          0.959                   0.162
HotelReviewScore   0.476            0.256
C?I                                 0.550
Promotion?                         -0.171   0.482
Saturday?
RamdomDisp?                0.995
Click?

                Factor1 Factor2 Factor3 Factor4
SS loadings       1.268   0.995   0.450   0.270
Proportion Var    0.159   0.124   0.056   0.034
Cumulative Var    0.159   0.283   0.339   0.373
```

Fig.1: R output of PCA & CFA

Based on the results of the PCA, the following formulas were obtained:

PC1 = 0.796Location1 + 0.859Location2
PC2 = 0.856NAdult + 0.861NRoom
PC3 = 0.591StayLength + 0.615HStay + 0.567NKid
PC4 = 0.544H_PurPrice + 0.678LogHotelP + 0.607DisplayP

For PC1, it is clear that location plays an important role in considering to book a hotel.
For PC2, the contributors are the number of adults specified in the hotel room and the number of hotel rooms specified in the search. Therefore, PC2 represents hotel room information.
For PC3, the length of stay at the hotel is the main consideration.
For PC4, the main reason for considering is the price.

According to the output of the CFA, the following information has been identified:

For F1, the main factor is the hotel's star rating.
For F2, whether the hotel's display of information was random or not is also the main factor.
For F3, the main factor is whether the hotel is part of a chain or is an independent hotel.
For F4, it appears that the hotel's promotion is an important factor for consideration.

We have 11 numeric variables. After using PCA to analyze the numeric data, we identified that 8 components can explain over 90% of the variance. In addition, we have 8 ordinal variables and after using CFA to analyze the ordinal data, it revealed that 7 factors can explain over 90% of the variance. Therefore, the results indicate that the PCA and CFA do not achieve the purpose of dimensional reduction. As a result, we decided to use 4 components to explain 50% of the variance and 4 factors to explain 37% of the variance.

Based on data obtained from these techniques, the next steps were to apply Logistic Regression and Linear Discriminate Analysis to build several models and test the predictive ability of them.

Logistic Regression

1. Statistic Test of Assumption of Logistic Regression

When the dependent variable is binary, the logistic model assumes that the response variable is followed by a binomial distribution. Ideally, the variance of Y, which is followed by the binomial distribution, should equal to np(1-p). When the variance of the sample Y is higher than the expected value of the variance, it causes over-dispersion - which further causes an inaccurate result of the significant test and a biased standard error. In our case, we needed to test whether the over-dispersion problem existed in our sample. The definition of dispersion is as follows:

$$\phi = \frac{Bias\ of\ Residual}{DOF\ of\ Residual}$$

When the ϕ is far higher than 1, we can say that over-dispersion exists. In our case, $\phi = 1.32$. We can also do the hypothesis test on ϕ. A function in R called "pchisq" allowed us to do the hypothesis test.

$$\text{Ho: } \phi = 0$$
$$H_{\partial}: \phi \neq 0$$

In our case, the p-value equals to 0.4947 which is higher than 0.05. Therefore, we cannot reject the null hypothesis which reinforced our assumption that the sample is not over-dispersed.

2. Two Logistic Regression Models

The first model uses all variables in the data set, while the second model was computed by the stepwise variable selection method using the AIC as the criteria.

The results of the first model indicate that some of the variables don't significantly affect the response variable so we could have either removed those variables or used a variable selection method to obtain another model. After the variable selection method, there were only 7 variables that were significant in Model #2.

In order to determine whether the variable deduction will reduce the fit level of the model, the ANOVA function was used to test whether removing the variables can significantly increase the accuracy of the prediction. Results indicate that the p-value is low, so the simple model can be used for the prediction.

```
> anova(model2,model1,test='Chisq')
Analysis of Deviance Table

Model 1: Y.f ~ RC1 + RC3 + RC4 + HotelStar + `C?I` + `Promotion?` + `RamdomDisp?`
Model 2: Y.f ~ RC1 + RC2 + RC3 + RC4 + H_PurStar + HotelStar + HotelReviewScore +
    `C?I` + `Promotion?` + `Saturday?` + `RamdomDisp?`
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      7927      10516
2      7912      10466 15   50.285 1.082e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Fig.2: Result of Anova test

3. Parameter Interpretation

```
> exp(coef(model2))
   (Intercept)            RC1            RC3            RC4     HotelStar2     HotelStar3     HotelStar4     HotelStar5
     0.2271288      1.1134888      0.8688175      0.8014332      2.0480143      3.3812550      4.3513980      3.7256257
        `C?I`1  `Promotion?`1 `RamdomDisp?`1
     1.3792809      1.2725751      0.3199291
```

Fig.3: Exponential Coefficient of Each variable

We can see that the star rating of the hotel will increase the probability of a customer eventually booking a hotel. Fig. 3 shows us that the probability of booking increases as the star rating increases, but decreases for the five-star rating. This could possibly be due to the fact that five-star hotels are more expensive. For example, when a hotel has a two-star rating, keeping the other variables constant, the odds of booking will be 2.04 times higher. On the other hand, when the hotel information is not randomly displayed, the customer will probably not book the hotel because the estimator is less than one.

In addition to the information above, insights on how each variable at the different levels will influence the probability of booking can also be provided. To illustrate, when we want to compare the levels of the ‘HotelStar’ variable, we need to keep the means of other numeric variables constant, while other ordinal variables stay constant at any level.

```
  HotelStar        RC1        RC3         RC4 C?I RamdomDisp? Promotion?      prob
1         1 0.03985938 -0.0561581 -0.05596895   1          1          1 0.1156057
2         2 0.03985938 -0.0561581 -0.05596895   1          1          1 0.2111766
3         3 0.03985938 -0.0561581 -0.05596895   1          1          1 0.3065133
4         4 0.03985938 -0.0561581 -0.05596895   1          1          1 0.3625714
5         5 0.03985938 -0.0561581 -0.05596895   1          1          1 0.3275068
```

```
> head(testdata1)
  Promotion?        RC1        RC3         RC4 C?I RamdomDisp? HotelStar      prob
1          0 0.03985938 -0.0561581 -0.05596895   1          1         3 0.2577849
2          1 0.03985938 -0.0561581 -0.05596895   1          1         3 0.3065133
> head(testdata)
  C?I        RC1        RC3         RC4 Promotion? RamdomDisp? HotelStar      prob
1   0 0.03985938 -0.0561581 -0.05596895          1          1         3 0.2426816
2   1 0.03985938 -0.0561581 -0.05596895          1          1         3 0.3065133
```

Fig.4:Potential Probability

As we can see, when the star rating goes from one to four, the odds of booking increases from 0.11 to 0.36. It is important to note that five-star hotels are different, so we might assume that the hotel is too expensive and that the customer will not book it. The conclusion that can be drawn is that people tend to book a hotel that is currently being promoted and is a chain hotel.

4. Model Validation

In order to test the generalization of the logistic regression model, the test data needed to be fit into it - computing the accuracy and confusion matrix. The output shows that accuracy equals to 0.58. The confusion matrix also shows the incorrect and correct number of cases.

```
> table(test$Y.f,fitted.results2)
   fitted.results2
      0   1
  0 716 607
  1 493 830
```

Fig.5: Confusion Matrix for Logistic Regression

Obviously, the logistic model’s ability to predict is relatively poor, so we considered using another technique to model our data.

Linear Discriminate Analysis

1. Statistic Test of Assumption of Linear Discriminate Analysis

The discriminant model has the following assumptions:

- Multivariate Normality - the Shapiro-Wilk test was applied on each numeric variable and so we can assume that all the variables are normally distributed or that their normality is statistically significant as the p-values are less than 0.05.
- Equality of variance-covariance within group - The covariance matrix within each group should be equal. The likelihood-ratio test was applied to verify it. Since the p-value was greater than 0.05, we can say that the covariance matrices are equal.
- Low multicollinearity of the variables - Since the PCA and CFA variables have already been applied, the multicollinearity issue has already been fixed.

2. Result Interpretation and Model Testing

There are three kinds of LDA methods that can be used.The build-in function in R to conduct LDA uses Bayes Discriminate. Because our Y is binary, there is only a single LDA that best separates the two classes. The confusion matrix shown below provides information about how many cases our model predicted incorrectly.

```
> table(test$Y.f,p)
   p
       0    1
  0 1308   15
  1    0 1323
```

Fig.6:Confusion Matrix for Linear Discriminate Analysis

The result of the LDA is relatively reasonable as there are only 15 cases that have been incorrectly classified by our model. More information about the manual computation of the LDA can be found in Harry's individual report.

Additional Study - KNN & Decision Tree

Use Clicking Behavior As The Target Variable

1. Data Overview

A different approach that we tried was using only the first 20,000 rows from the original 99 million rows and keeping all 52 independent variables in the data set. Instead of using booking or not booking a hotel as the dependent variable, we tried using clicking or not clicking on a hotel as the dependent binary variable. We tried this because each hotel (row) could be clicked (have a '1' in the click column) while only one hotel (row) could be booked (have a '1' in the book column) within one query.

There are 1,535 people from this specific data set. They have conducted 1,535 searches and Expedia has provided 20,000 hotels in total for their considerations. Of the 20,000 search results, 888 hotels have been clicked (have a '1' in the click column). The click-through-rate is 4.44%.

2. Data Preprocessing:

(1) Finding Correlation/Association: Using all 52 independent variables, we first needed to identify whether there are associations between the independent variables and the dependent variable. Since our dependent variable is a binary, the following methods were used to identify the associations:

a. **For independent variables that are numerical:** We first discretized them into groups based on our knowledge, then (1) identified whether each group has significantly different clicking rates than other groups and (2) conducted the Chi-Squared Test.

b. **For independent variables that are ordinal:** The Spearman and Kendall correlation tests were used.

c. **For independent variables that are nominal:** The Chi-Squared Test was used.

(2) Creating New Features: Based on the original data set, we created several features. An example would be SRMD (Star Rating Matching Distance), which is the visitor's historical average star rating of purchased hotels minus the hotel's star rating.

(3) Filling Missing Values: The ggplot or other methods were first used to determine which complete feature has a relationship with the missing features. Then, the MICE function in R was used to fill out the missing values. To illustrate, to fill out the missing values of visitor's historical ratings, we first identified whether the visitor's historical rating had a relationship with how many adults and rooms they specified when conducting a search. Then these relevant features are used in the MICE function.

3. Model Building and Fitting:

Pre-processing the data left us with 30 independent variables and 1 dependent variable. The 20,000 rows of data were split into a training set (80%) and a test set (20%).

(1) KNN:

The hyperparameters of KNN were tuned to search for the best model that has the highest f1 score under the condition that it was within 2 to 5 neighbors.

The model selected indicates that the best number of neighbours is 2.

```
In [327]: predictions=est.predict(x_test)

In [328]: confusion_matrix(y_test,predictions)
Out[328]:
array([[3669,  156],
       [ 165,   10]])
```

```
In [329]: print(classification_report(y_test,predictions))
             precision    recall  f1-score   support

          0       0.96      0.96      0.96      3825
          1       0.06      0.06      0.06       175

avg / total       0.92      0.92      0.92      4000
```

```
In [326]: cvs=cross_val_score(est,x_train,y_train,cv=10,scoring='f1')
     ...: cvs
Out[326]:
array([ 0.07407407,  0.07462687,  0.08633094,  0.07194245,  0.04724409,
        0.0794702 ,  0.06756757,  0.05925926,  0.08108108,  0.12162162])
```

Fig.7: KNN Model Confusion Matrix and Cross Validation Score

However, the f1 score is low. The reason could possibly be due to having included too many variables (30 variables) in the model. Therefore, the f1 score is low in the test set as well.

After using the PCA, the dimensions could only be reduced by 3 numerical dimensions in order to preserve 86% of the total variance. The model is not significantly better with 27 independent variables.

(2) Decision Tree and Random Forest:

Same as for KNN, the hyperparameters of the Decision Tree and Random Forest were first tuned to search for the best model that has the highest f1 score.

With 30 independent variables, both the Decision Tree and Random Forest have an f1 score that is better than the KNN model's.

Decision Tree:

```
In [264]: cvs=cross_val_score(est,x_train,y_train,cv=10,scoring='f1')
     ...: cvs
Out[264]:
array([ 0.22335025,  0.18018018,  0.16037736,  0.2038835 ,  0.18627451,
        0.16981132,  0.16666667,  0.15763547,  0.1793722 ,  0.1682243 ])
```

```
In [266]: print(classification_report(y_test,predictions))
             precision    recall  f1-score   support

          0       0.97      0.91      0.93      3825
          1       0.13      0.30      0.18       175

avg / total       0.93      0.88      0.90      4000
```

Fig.8: Decision Tree Model Cross Validation Score and Classification Report

Random Forest:

```
In [305]: cvs=cross_val_score(est,x_train,y_train,cv=10,scoring='f1')
     ...: cvs
Out[305]:
array([ 0.24742268,  0.18181818,  0.18867925,  0.20725389,  0.19704433,
        0.1719457 ,  0.18556701,  0.18719212,  0.19178082,  0.15962441])
```

```
In [308]: print(classification_report(y_test,predictions))
             precision    recall  f1-score   support

          0       0.97      0.91      0.94      3825
          1       0.12      0.29      0.17       175

avg / total       0.93      0.88      0.90      4000
```

Fig.9: Random Forest Model Cross-Validation Score and Classification Report

The model was then fit with the top 20 most important variables found by the ExtraTreesClassifier. The models have improved significantly compared to using 30 variables.

Below is the summary:

```
Out[408]:
  F1_score                                  Model  Numbers of Features Recall_score
3     0.18   Random Forest After Feature Selection                   20         0.29
2     0.17  Random Forest Before Feature Selection                   30         0.29
0     0.14  Decision Tree Before Feature Selection                   30         0.25
1     0.12    Decison Tree After Feature Selection                   20         0.65
5     0.07         KNN After PCA Feature Selection                   27         0.07
4     0.06         KNN Beore PCA Feature Selection                   30         0.06
```

Fig.10: Models Comparison

Therefore, for practical significance, Expedia should first increase the clicking rate in order to increase the booking rate of hotels. Then the model that has the highest recall score should be used because recall is more important than the accuracy score in this case. However, for statistical significance, the models that have the highest f1 score might be preferred.

Conclusion and Potential Shortfalls

For logistic Regression, we found that when the star rating goes from one to four, the odds of booking increases from 0.11 to 0.36. It is important to note that five-star hotels are different, so we might assume that the hotel is too expensive and that the customer will not book it. Another conclusion that can be drawn is that people tend to book a hotel that is currently being promoted and is a part of a chain.

For Linear Discriminate Analysis, the result were quite satisfactory as it revealed that there are only 15 cases that have been incorrectly classified by our model.

For the additional study that treated ‘click_bool’ as the target variable, we found that the positioning is the most important features in predicting the clicking outcome. Meanwhile, other features such as ‘property_location_score2’ and ‘price_usd’ are crucial factors as well. In addition to these original features, some features that we engineered such as L2D (prop_location_score2 minus the mean prop_location_score2) in the query are also important. Therefore, we should try to generate more important features following this information. In addition, we have included too many features (30) in the model which was not helpful. In future studies, only the significant variables will be used and we won’t be inflating the model by just adding to the size of independent variables.

We have conducted several analyses and identified some important variables that pertain to booking a hotel, such as: promotion, chain store, or history price - which are great contributions to hospitality companies. However, there are still some problems which need to be tackled.

First of all, even though we have cleaned our dataset, some aspects of it cannot be cleaned. Therefore, our model could be infected by some bias and outliers. Next, there are 8 competitor variables in our dataset and we computed the average for these 8 variables in order to simplify analyzing competitor data. However, because the computed variables were not original variables, they may have jeopardized our model. After that, we left out the search ID's and other ID variables, which didn't appear to be useful in the analyses we conducted - or in statistics in general. On the other hand, it does have relatively significant meaning in the real world. We could divide them into several continent groups and research deeply into it geographically.

Individual Report

Ruoxi Wang

Part I: Summary of the Technique

I tried to use PCA and CFA to analysis our data.

- Due to our dataset is huge, we need avoid the data with large variance dominate the components, so I scaled data first.
- Our dataset after cleaning sill include numeric and ordinal two types of data. Therefore, I use different methods to see the correlation.

-Using Pearson matrix to see the correlation of numeric variables.

```
> c <- cor(numericMatrix)
> round(c,2)
           H_PurPrice Location1 Location2 LogHotelP DisplayP StayLength HStay NAdult  NKid NRoom CompePD
H_PurPrice       1.00      0.15      0.04      0.03     0.20       0.06  0.10   0.01  0.04  0.01    0.01
Location1        0.15      1.00      0.43      0.08     0.29       0.12  0.16   0.03  0.03  0.01    0.03
Location2        0.04      0.43      1.00      0.04     0.12       0.03  0.04   0.01 -0.04 -0.04    0.01
LogHotelP        0.03      0.08      0.04      1.00     0.11       0.02  0.03  -0.01 -0.04 -0.01    0.00
DisplayP         0.20      0.29      0.12      0.11     1.00       0.08  0.10   0.07  0.06  0.03    0.02
StayLength       0.06      0.12      0.03      0.02     0.08       1.00  0.16   0.03  0.05  0.00    0.01
HStay            0.10      0.16      0.04      0.03     0.10       0.16  1.00   0.14  0.10  0.04    0.01
NAdult           0.01      0.03      0.01     -0.01     0.07       0.03  0.14   1.00  0.03  0.50    0.00
NKid             0.04      0.03     -0.04     -0.04     0.06       0.05  0.10   0.03  1.00  0.12    0.01
NRoom            0.01      0.01     -0.04     -0.01     0.03       0.00  0.04   0.50  0.12  1.00    0.02
CompePD          0.01      0.03      0.01      0.00     0.02       0.01  0.01   0.00  0.01  0.02    1.00
```

Fig.1:Pearson Matrix

-Using Spearman and Polychoric matrix to see the ordinal variables correlation.

Spearman:

```
> cS <- cor(ordinalMatrix,method = 'spearman')
> round(cS,2)
                H_PurStar HotelStar HotelReviewScore   C?I Promotion? Saturday? RamdomDisp? Click?
H_PurStar            1.00      0.31             0.11 -0.11       0.04     -0.01       -0.03  -0.01
HotelStar            0.31      1.00             0.48  0.01       0.12      0.01       -0.01   0.02
HotelReviewScore     0.11      0.48             1.00  0.13       0.01      0.02       -0.03   0.02
C?I                 -0.11      0.01             0.13  1.00      -0.12      0.02       -0.03   0.02
Promotion?           0.04      0.12             0.01 -0.12       1.00     -0.01       -0.03   0.03
Saturday?           -0.01      0.01             0.02  0.02      -0.01      1.00        0.03   0.00
RamdomDisp?         -0.03     -0.01            -0.03 -0.03      -0.03      0.03        1.00   0.01
Click?              -0.01      0.02             0.02  0.02       0.03      0.00        0.01   1.00
```

Fig.2:Spearman Matrix

Ploychoric

```
> Polychoric <- hetcor(ord.f)$cor
There were 21 warnings (use warnings() to see them)
> round(Polychoric,2)
                H_PurStar HotelStar HotelReviewScore   C.I Promotion. Saturday. RamdomDisp. Click.
H_PurStar            1.00      0.35             0.12 -0.15       0.06     -0.01       -0.05  -0.02
HotelStar            0.35      1.00             0.53  0.02       0.16      0.01       -0.01   0.05
HotelReviewScore     0.12      0.53             1.00  0.18       0.02      0.03       -0.05   0.04
C.I                 -0.15      0.02             0.18  1.00      -0.19      0.04       -0.05   0.06
Promotion.           0.06      0.16             0.02 -0.19       1.00     -0.01       -0.05   0.08
Saturday.           -0.01      0.01             0.03  0.04      -0.01      1.00        0.05  -0.01
RamdomDisp.         -0.05     -0.01            -0.05 -0.05      -0.05      0.05        1.00   0.03
Click.              -0.02      0.05             0.04  0.06       0.08     -0.01        0.03   1.00
```

Fig.3:Ploychoric Matrix

-Using polyserial matrix to see the mix correlation.

	Promotion?	NAdult	NKid	NRoom	Saturday?	RamdomDisp?	Click?
H_PurStar	0.06	-0.03	0.02	0.05	-0.02	-0.03	-0.01
H_PurPrice	0.02	0.01	0.09	0.05	-0.01	-0.01	-0.01
Location1	0.18	0.05	0.05	0.03	0.00	0.02	-0.02
Location2	0.04	0.03	-0.06	-0.09	0.01	0.04	0.12
LogHotelP	0.01	-0.01	-0.06	-0.02	0.00	-0.03	0.01
DisplayP	-0.08	0.07	0.09	0.04	0.02	0.05	-0.04
HStay	0.10	0.18	0.12	0.06	0.07	0.16	-0.02
CompePD	NA	NA	NA	NA	NA	NA	NA
HotelStar	0.18	0.01	0.00	0.05	0.01	-0.01	0.06
HotelReviewScore	0.03	0.02	0.01	0.02	0.03	-0.05	0.04
C?I	-0.19	-0.04	0.03	0.09	0.04	-0.05	0.06
Promotion?	1.00	0.05	0.00	-0.03	-0.01	-0.05	0.08
NAdult	0.05	1.00	0.02	0.49	0.13	0.16	-0.02
NKid	0.00	0.02	1.00	0.32	0.02	0.08	0.03
NRoom	-0.03	0.49	0.32	1.00	-0.06	0.07	0.06
Saturday?	-0.01	0.13	0.02	-0.06	1.00	0.05	-0.01
RamdomDisp?	-0.05	0.16	0.08	0.07	0.05	1.00	0.03
Click?	0.08	-0.02	0.03	0.06	-0.01	0.03	1.00

Fig.4:Polyserial Matrix

According to these results, I think our dataset is not suitable for using PCA and CFA. However, due to our class requirement and we want to match it, so I decided to use PCA and CFA to see what happened. I use PCA to analysis the numeric data and CFA to analysis the ordinal data.

Loadings:

	RC2	RC1	RC3	RC4
NAdult	0.856			
NRoom	0.861			
Location1		0.796		
Location2		0.859		
StayLength			0.591	
HStay			0.615	
NKid			0.567	
H_PurPrice				0.544
LogHotelP				0.678
DisplayP				0.607
CompePD				

	RC2	RC1	RC3	RC4
SS loadings	1.511	1.511	1.272	1.214
Proportion Var	0.137	0.137	0.116	0.110
Cumulative Var	0.137	0.275	0.390	0.501

Loadings:

	Factor1	Factor2	Factor3	Factor4
H_PurStar	0.342		-0.201	
HotelStar	0.959			0.162
HotelReviewScore	0.476		0.256	
C?I			0.550	
Promotion?			-0.171	0.482
Saturday?				
RamdomDisp?		0.995		
Click?				

	Factor1	Factor2	Factor3	Factor4
SS loadings	1.268	0.995	0.450	0.270
Proportion Var	0.159	0.124	0.056	0.034
Cumulative Var	0.159	0.283	0.339	0.373

Fig.5:Loading of PCA and CFA

I find that PCA and CFA are both not achieve the purpose of dimensional reduction. So I decided to use 4 components to explain 50% of the variance and 4 factors to explain 37% of the variance.

Part II:What I learn from multivariate statistics

I learned that when we deal with the different type of data, we should be careful about it. Because sometimes we use the same method to deal with it! However, different data type need different methods. Also, I learned that it is common that the dataset could not use PCA and CFA or some other methods to analysis the data. Before, I thought that most of the dataset can use it. However, just because our example and homework's dataset which are almost completely, so we can always use it. In fact, when we are working in our daily life, we will always meet the raw dataset, which is complex and messy. So not all the methods or technology we learned can be always use to analysis.

Harry Chen

The individual Report consists of two part, First, I am going to basically summarize how I apply Logistic regression and Linear Discriminate Analysis on our data. The second part is mainly for the summary of the whole class.

Part I: Summary of the Technique

Logistic Regression:

1. **Test the assumption of the regression model**
   - Use Chi-square test on Y to see whether they are following binominal distribution
   - Test the normality of each numeric variable using QQ plot. (if the point is almost distribute alongside the 45^o slope line, we can say it's normally distributed).

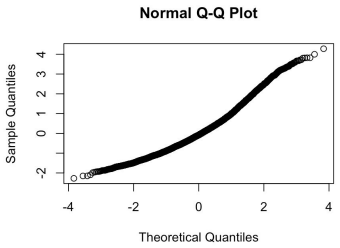


Fig.1: Q-Q plot

2. **Building model**
   - Split the dataset, fit the training data into logistic regression function and use test set to test the model
   - Apply ANOVA test on each variable to see if they are significant or not.
   - Remove insignificant variable the rebuilt the model
3. **Interpretation**
   - Interpret the parameter by keeping other variables stay and focus on one and then compute the confidential interval for each level in ordinal variable.
   - What I find is promotion will increase the probability of a customer to eventually book a hotel. If the hotel is a chain one, it has more chance to be booked.

Discriminate Analysis

I want to insight more in how LDA works, so in the following part, I will summary how I manually compute the result of LDA by Two methods.

1. **Mahalanobis Distance.**

   Assume that we have two populations G1 and G2, one is people who booked a hotel, second is people who have not booked a hotel. Now that we have a customer X, we want to predict whether he will book the hotel or not based on the characteristics (variables in our case). We compute the distance between the target point and two populations, and choose the population with the smallest distance from the target point. The following equation is the distance between X and population G.

$$D^2(X,G) = (X-\mu)\sum^{-1}(X-\mu)$$

   The criteria of classification as follow

   When $\mu 1 \neq u2 \; and \sum 1 \neq \sum 2$

$$W(X) = D^2(X,G1) - D^2(X,G2)$$

$$X \in G1 \;\; if \qquad W(X) \leq 0$$
$$X \in G2 \;\; if \qquad W(X) > 0$$

   First, I compute the mean vector and covariance matrix of each population

```
M0 <- colMeans(G0)
M1 <- colMeans(G1)
V0 <- cov(G0)
V1 <- cov(G1)
```

Fig.2: Code for computing Mean Vector

Second, I compute the MD between the cases that in test set and G1 and G2

```
Dis0 <- mahalanobis(target,M0,V0)
Dis1 <- mahalanobis(target,M1,V1)
```

Fig,3: Code for computing Mahalanobis Distance

Third, Using the criteria to do the classification.

```
> table(test$Y.f,result)
   result
       0    1
  0 1044  279
  1  835  488
```

Fig.4: Confusion Matrix MD

As we can see from the result, the discriminate is not good at test set by using Mahalanobis distance.

2. **Bayes Discriminate.**

Assume population G1 and G2 following $N(\mu 1, \sum 1), N(\mu 2, \sum 2)$, the prior probability of G1 and G2 are q_1 and q_2, and the cost of misjudge are C(Y=0|Y=1) and C(Y=1|Y=0). In our case, we assume $q_1=q_2$ and the loss of predict a customer who are willing book a hotel incorrectly is more severe, in other word, C(Y=0|Y=1) > C(Y=1|Y=0).

The criteria of classification is as follow:

$$R1 = \{X | q * f_1(X) * C(Y=1) \leq q * f_2(X) * C(Y=0)\}$$
$$R1 = \{X | q * f_1(X) * C(Y=1) > q * f_2(X) * C(Y=0)\}$$
$$V(X) = \frac{f_1(X)}{f_2(X)}, d = \frac{C(Y=0)}{C(Y=1)} = 1/2$$
$$X \in G1 \ if \ \ V(X) > d$$
$$X \in G2 \ if \ \ V(X) < d$$

The result of this method is shown as below:

```
> table(test$Y.f,result1)
   result1
       0    1
  0  490  833
  1  245 1078
```

Fig.5: Confusion Matrix BD

Part II:What I learn from multivariate statistics

From the beginning of the course, I learn how to apply regularization method to optimize the model performance when the data are sparse. We could use lasso and ridge method to solve the overfitting problem.

The main acknowledge of this course is how to apply principle component analysis and factor analysis to deal with multicollinearity properly, and how to do the dimension reduction and explore latent factors in the dataset.

I also learn how to apply correspondent analysis on nominal data, how to do canonical analysis when we have multiple response variable, and how to properly dealing with ordinal variables (instead of using Pearson correlation, we use polychoric, spearman and Kendell correlation). LAD and Clustering help us to deal with classification problems.

Yimei Tang

Part I: Summary of the Technique

In this project, I mostly deal with the subset of the original data(the first 20000 rows of 1 million rows) and treat the clicking_bool as target variable.

To predict the outcome of clicking on a hotel or not, I first analyze the association with each independent variable with the dependent variable and decided to keep them in my model building or not. During this process, I mainly use Chi-Squared Test, Spearman/Kendall Correlation Test and analyze the clicking rate difference. Below are some visualizations:

```
> chisq.test(as.data.frame.matrix(table(full$RG,full$click_bool)),simulate.p.value=T)

        Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)

data:  as.data.frame.matrix(table(full$RG, full$click_bool))
X-squared = 6.8602, df = NA, p-value = 0.985
```

Fig.1: Chi-Squared Test Results

I also created new features and fill the missing values with the most probable values by MICE. Below are some visualizations:

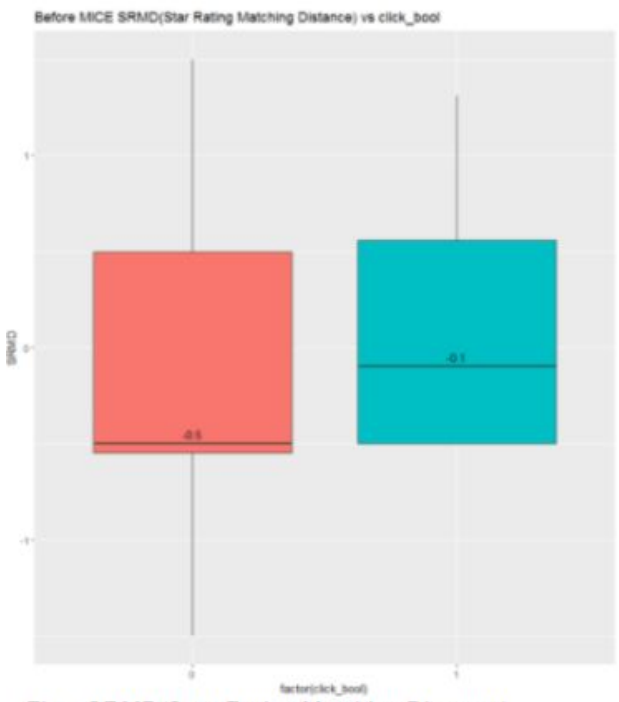


Fig.2 SRMD(Start Rating Matching Distance) Vs Click_bool(Red one stands for no clicking, Blue one stands for clicking)Before MICE Imputation

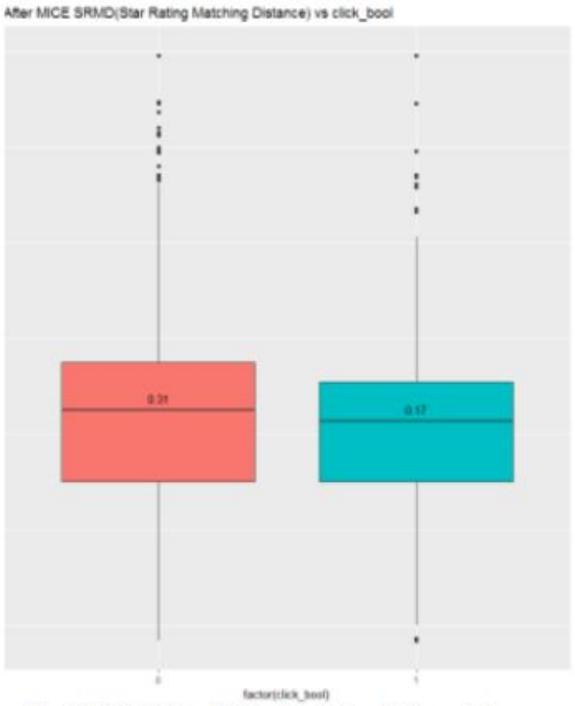


Fig.3 SRMD(Star Rating Matching Distance) Vs Click_book(Red one stands for no clicking, Blue one stands for clicking)After MICE Imputation

I use ExtraTreeClassfier, PCA in feature selection process and KNN, Decision Tree and Random Forest these three algorithms in model building:

PCA:

```
> summary(trainPCA)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8    PC9    PC10    PC11    PC12    PC13
Standard deviation     1.3020 1.2000 1.1603 1.13549 1.02036 1.00534 0.97010 0.94471 0.9192 0.87841 0.82826 0.7924 0.6429
Proportion of Variance 0.1304 0.1108 0.1036 0.09918 0.08009 0.07775 0.07239 0.06865 0.0650 0.05935 0.05277 0.0483 0.0318
Cumulative Proportion  0.1304 0.2412 0.3447 0.44390 0.52399 0.60174 0.67413 0.74278 0.8078 0.86714 0.91991 0.9682 1.0000
```

```
> round(trainPCA$rotation,2)
                            PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10
visitor_hist_starrating   -0.23  0.07 -0.21  0.45 -0.10  0.08 -0.57  0.12 -0.18 -0.49
position                   0.07 -0.33 -0.30 -0.09  0.08  0.46 -0.11  0.46  0.54  0.05
price_usd                 -0.04  0.42 -0.51 -0.22 -0.09 -0.11  0.02 -0.02  0.05 -0.06
promotion_flag            -0.07  0.07  0.13  0.27 -0.52 -0.55  0.10  0.30  0.47  0.04
srch_booking_window       -0.27  0.06 -0.19  0.48  0.20  0.00  0.00  0.17 -0.22  0.73
srch_adults_count         -0.62 -0.09  0.06 -0.17 -0.06  0.01 -0.25 -0.15  0.12  0.02
srch_children_count       -0.33 -0.04  0.01 -0.07  0.41 -0.20  0.44  0.55 -0.20 -0.37
srch_room_count           -0.58 -0.11  0.12 -0.33 -0.12  0.07  0.06 -0.16  0.09  0.11
srch_saturday_night_bool   0.11  0.03  0.03 -0.21  0.57 -0.53 -0.50 -0.02  0.24  0.10
random_bool               -0.13  0.11 -0.23  0.39  0.33  0.04  0.36 -0.50  0.46 -0.20
L1D                       -0.02  0.49  0.33  0.00  0.12  0.33 -0.07  0.22  0.26  0.07
L2D                       -0.03  0.49  0.42  0.05  0.08  0.17 -0.04  0.04  0.03 -0.06
PD                        -0.02  0.43 -0.45 -0.30 -0.13 -0.01  0.04  0.09 -0.08  0.10
```

Fig.4:PCA Results

ExtraTreeClassifer:

```
###ExtraTreesClassifier
forest= ExtraTreesClassifier(n_estimators=500, class_weight='balanced')

In [294]: feature_idx = sel.get_support()
     ...: feature_name = train_df.columns[feature_idx]
     ...: feature_name
Out[294]:
Index(['visitor_hist_starrating', 'prop_starrating', 'prop_review_score',
       'prop_brand_bool', 'prop_location_score1', 'prop_location_score2',
       'prop_log_historical_price', 'position', 'price_usd', 'promotion_flag'
       'srch_booking_window', 'srch_saturday_night_bool', 'random_bool',
       'SRMD', 'RTD', 'RSD', 'L1D', 'L2D', 'PD', 'TofD_Afternoon',
       'TofD_Morning', 'TofD_Night'],
      dtype='object')
```

Fig.5: ExtraTreesClassifier Feature Selection Results

Part II:What I learn from multivariate statistics

First, I learnt that the curse of dimension is a true problem.In this dataset, I tried to preserve as many features as I think they are related to the click_bool and the result is that my models are not doing great with 30 variables. All of my models have improved their f1 score on test set after using fewer but more important features by PCA or ExtraTreeClassifier.

Second, master the application of PCA , CFA and MDS will give us great insights in how to reduce certain features so to only preserve the most important features. More importantly, LDA is a very useful algorithm in non-linear classification problems.

Third, when one dependent variable is hard to predict, we could consider other related dependent variables and conduct canonical correlations to identify underlying relationship

Yuedi Wang

The research goal of our project is to identify the most significant feature that contributes to obtaining realistic predictions on whether a hotel will be booked. The original data set is obtained from Kaggle.com, and consists of a representative sample of 9,000,000 hotels. Also, fifty-two features are included, i.e., price, user history, competitors.

Besides the whole non-technical report, my analysis part is also focused on two methods – PCA and CFA, and multidimensional scaling.

In the first milestone, I conduct the PCA and CFA for the dataset. And my analysis consists of 4 parts - data exploration, data analysis, and evaluation.

1. Data exploration

Because only numeric data could be applied in PCA and CFA analysis, I remove all ordinal and nominal variables from the data set. And only 18 variables have been left. Then I remove the missing value for these 18 variables and 263907 rows are left.

2. Data analysis

Let's continue to the data analysis part. Firstly, I compute PCA using covariance matrix. PC1 contributes to 96% of the variation. And PC2 contributes to 3.8% of the variation. The difference between PC1 and PC2 is huge. Secondly, I compute PCA using correlation matrix. 15 principal components are required to explain 90% of the total variation, but we only have 18 variables. We should try to rotate the components, site_id, orig_destination_distance, srch_length_of_stay, srch_booking_window contribute positively to RC1, visitor_location_country_id, prop_country_id contribute negatively to RC1, and prop_starrating, prop_review_score, prop_location_score1, prop_location_score2 contribute positively to RC2. RC1 contributes to 12.6% of the variation and RC2 contributes to 10% of the variation. The results are not ideal. For the CFA factanal analysis, PC1 contributes to 8.6% of the variation and PC2 contributes to 6% of the variation.

3. Evaluation

This issue stops my analysis work and I need to analyze the reason behind this problem. On one hand, some variables are not correlated with others, so they work as a component on their own. On the other hand, I still need to study on correlation matrix and Spearson matrix deeply and find out the true relation between different variables.

In the second milestone, I reconduct PCA and CFA and perform multidimensional scaling for the dataset. And my analysis consists of 4 parts - data exploration, data analysis, and evaluation.

1. Data exploration

In the second milestone, I change my dataset. Because the value of our data is huge, I clean the dataset and compute the average for competitor variables. Also, in order to avoid large variance on variables, I scale the data.

The cleaned dataset still includes numeric and ordinal variables. Therefore, I use different methods to compute the correlation.

First, use Pearson matrix to see the correlation of numeric variables. Second, use Spearman and Polychoric matrix to see the ordinal variables correlation. Also, use polyserial matrix to see the mix

correlation. Next, I use PCA to analyze the numeric data and use CFA to analyze the ordinal data. After these three steps, my result is much better.

2. Data analysis

Based on the result from last milestone, I have learnt a lot about how to compute PCA and CFA in a more proper way. To compute PCA, we should use numeric data, but our variables are including numeric and Boolean, which is not suitable for analysis. So, person correlation matrix is not enough. Instead, importing the Spearman and Polychoric matrix could help us solve the problem. The final result is that 4 components could explain 60% of the dataset and 8 components could explain 90% of the dataset, which is way better than the result in homework 3.

For the multidimensional scaling part, I run multidimensional scaling on the distance matrix with the "cmdscale" command first. Also I check my result with a non-metric version of MDS. I find out that "prop_log_historical_price" and "visitor_hist_adr_usd" are two important predictors.

3. Evaluation

For the multidimensional scaling part, one variable is about history price, which is a big thing in hotel booking. The other one is about visitor history. It does make sense in real world.

However, my output is not as good as Harry's, so we decide to use Harry's output in the final presentation.

Above all, I have learnt a lot from this project. In terms of the example in class, the first component could explain at least 60% of the data and two or three components could explain 90% of the data. But in our dataset, two components could only explain 10% of the data. When I try to explore the reason behind the question, I find out that we should compute Pearson and Spearman and Polychoric matrix before any PCA or CFA analysis. Also, I have improved the proficiency of using PCA and CFA.

Besides, the non-technical report is finished by me. We cannot just focus on our code, instead, interpretation part is the most important. To perform better in non-technical report, I have done a lot of research on the structure of the report. Also, I learn how to integrate the output of all my team members and find out the standout point for the report.

Finally, say thank you to the Professor McDonald sincerely, he is charming and his lecture is lively. From the PCA, CFA, to correspondence analysis to CA to discriminant analysis, I gain a lot of knowledge from each lecture and review them by videos, homework and office hour. After this class, I feel more confident in data analysis part and believe that I could perform well in my future work.

Marsela Kapetanovic

The following three sections outline the approaches I have taken in contributing to our team's efforts. The fourth section highlights the key takeaways about multivariate statistics I have learned from this course.

1. Principal Component Analysis (PCA) – Unscaled

My first approach was to to analyze the entire cleaned data set, consisting of 127,359 observations (after removing nominal and ordinal variables and filling in missing values), in order to pursue dimension reduction. Dimension reduction would help us in condensing the data set even further, while capturing the most variance.

To avoid potentially producing misleading results, I first conducted a heterogeneous scatter matrix of the data to determine whether we had variables that were highly correlated with multiple other variables. This issue is known as multicollinearity. However, there weren't any that were identified as highly correlated, but there was only one relatively strong positive relationship between pairs 'booking_bool' and 'click_bool".

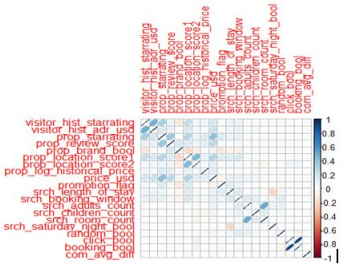
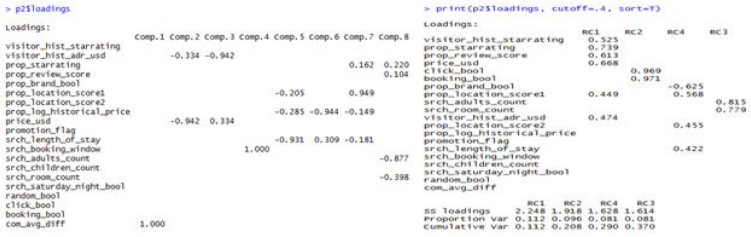


Figure 1 - Heterogeneous Scatter Matrix

Then I proceeded with running the importance of the loadings and determined that two principal components are required to explain at least 90% of the total variation (96.08% to be exact). Without rotating the components, the first principal component only had a single component contributing to it. Rotating the components provided better results and loadings for four components that were interpretable.



```
> p2$loadings

Loadings:
                       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
visitor_hist_starrating
visitor_hist_adr_usd          -0.334 -0.942
prop_starrating                                                0.162  0.220
prop_review_score                                              0.104
prop_brand_bool
prop_location_score1                                 -0.205  0.949
prop_location_score2
prop_log_historical_price                            -0.285 -0.944 -0.149
price_usd                     -0.942  0.334
promotion_flag
srch_length_of_stay                                  -0.931  0.309 -0.181
srch_booking_window                         1.000
srch_adults_count                                                 -0.877
srch_children_count
srch_room_count                                                   -0.398
srch_saturday_night_bool
random_bool
click_bool
booking_bool
com_avg_diff           1.000
```

```
> print(p2$loadings, cutoff=.4, sort=T)

Loadings:
                       RC1    RC2    RC4    RC3
visitor_hist_starrating  0.525
prop_starrating          0.739
prop_review_score        0.613
price_usd                0.668
click_bool                      0.969
booking_bool                    0.971
prop_brand_bool                        -0.625
prop_location_score1     0.449         0.568
srch_adults_count                              0.815
srch_room_count                                0.779
visitor_hist_adr_usd     0.474
prop_location_score2                   0.455
prop_log_historical_price
promotion_flag
srch_length_of_stay                    0.422
srch_booking_window
srch_children_count
srch_saturday_night_bool
random_bool
com_avg_diff

                 RC1   RC2   RC4   RC3
SS loadings    2.248 1.918 1.628 1.614
Proportion Var 0.112 0.096 0.081 0.081
Cumulative Var 0.112 0.208 0.290 0.370
```

Figure 2. a) Without Rotation, b.) With Rotation

PC1 represented the standard common characteristics of a hotel that consumers consider when booking (price, star rating, etc.). PC2 represented whether the hotel had been booked or clicked on. PC3 represented hotel room information. Finally, PC4 represented "fixed" specifications (that can't be changed easily) such as the location and nature of the hotel (chain or independent).

I also tried conducting the PCA with scaled data, but this caused the majority of the components to be represented by only one variable. Therefore, I have come to the conclusion that scaling the data is not the right approach for this data set because it is better to leave the price variables as is (as they are causing the large gap in the values) instead of having them scaled to better match the other variables.

2. Common Factor Analysis (CFA)

Because the PCA (without scaling) indicated that most of the total variability is explained by PC1 and PC2, I first tried conducting the CFA based on 2 factors. This approach produced results relatively similar to PC1 as it pertains to the main contributors.

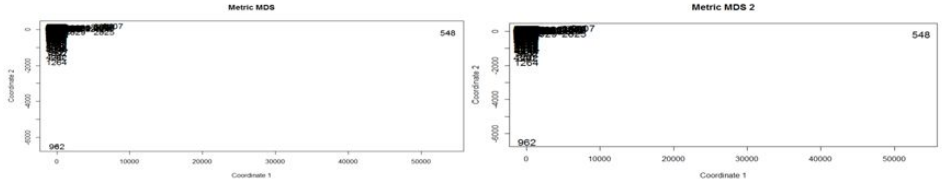
After that, I tried conducting the CFA based on 3 factors, and then 4. Adding more than 2 factors created results that were unsatisfactory because it produced one variable per factor. In conclusion, the results of the CFA for 2 factors produced results that were somewhat satisfactory and were relatively similar to those of the PCA (specifically the main contributors of each component).

Figure 3 - a.) CFA with 2 factors, b.) CFA with 3 factors, c.) CFA with 4 factors

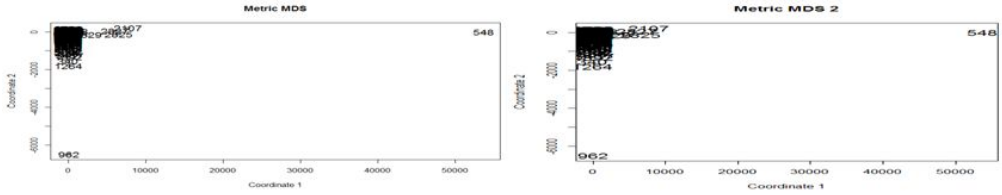
3. Multidimensional Scaling (MDS)

I conducted the MDS analysis with the goal of identifying groupings of variables that could potentially indicate which variables go together when considering to book a hotel. To illustrate, I was hoping that it would reveal all variables related to price as one grouping, all variables related to room information in another, etc. This would have helped us in supporting the interpretations of our PCA results.

To run the analysis, I first reduced the number of observations to 5,000. Only two points were distinct, while all the others were piled on top of each other in the plot and were not clearly distinguishable (plot below on the left). I then reran multidimensional scaling with “isoMDS” (non-metric version of MDS) and the plot of the points indicated that the same two points again represent their own groupings. All the other points were still clustered together in what appears to look like a 90 degree angle (plot below on the right).



I then reran the both the metric (below on the left) and non-metric (below on the right) versions of MDS analysis with the number of observations equaling 3,500. However, I obtained almost identical results as when running 5,000 observations.The only difference is that this time, the angle was a bit looser. Therefore, the results of the MDS indicated that the approach is not suitable for our data set. The poor results were most likely due to the fact that we have a large data set and the points (each observation represents a hotel) are not necessarily representative of the variables that we were hoping would reveal groupings.



4. Key Takeaways

The overall most important thing that I have learned is the importance of knowing the data set I am dealing with in order to be able to approach it properly (as it relates to selecting statistical techniques, cleaning the data set, condensing it, interpretation, etc.). Contributing to the PCA and CFA analyses

taught me that the techniques are sensitive to strong correlations as it causes multicollinearity that may impact an analysis by producing misleading results. It is advantageous detect highly positive or highly negative relationships early in the analysis. From working on MDS, I have learned that the size of a data set can impact whether an analysis can be used. I can understand why it wasn't suitable in our case, but I was surprised by the results as I was expecting the points to be scattered around – and not piled on top of each other in the shape of a 90 degree angle and having only 2 distinguishable points.

Appendix

Fig in Non-Technical Report

Fig in Technical Report

Fig in Individual Report (RuoXi Wang)

Fig in Individual Report (Harry Chen)

Fig in Individual Report (Yimei Tang)

Fig in Individual Report (Marsela Kapetanovic)

R code

```
library(Amelia)
library(ggplot2)
library(GGally)
library(corrplot)
library(mice)
library(data.table)
library(sqldf)
library(VIM)

# data exploration part
drops=
c('date_time',"comp1_rate","comp1_inv","comp1_rate_percent_diff","comp2_rate","comp2_inv","
comp2_rate_percent_diff","comp3_rate","comp3_inv","comp3_rate_percent_diff","comp4_rate","
comp4_inv","comp4_rate_percent_diff","comp5_rate","comp5_inv","comp5_rate_percent_diff","c
omp6_rate","comp6_inv","comp6_rate_percent_diff","comp7_rate","comp7_inv","comp7_rate_p
ercent_diff","comp8_rate","comp8_inv","comp8_rate_percent_diff")
df_old <- fread('train.csv',drop ='date_time',nrows = 100000,header = TRUE,na.strings =
c('NULL'))
df <- fread('train.csv',drop =drops,nrows = 100000,header = TRUE,na.strings = c('NULL'))

#is there any missing value in data?
data.frame(number_na=apply(is.na(df_old),2,sum))
dim(df_old)

# rate of missing value
pmiss <- function(x){sum(is.na(x))/length(x)}
data.frame(missingrate=apply(df_old, 2, pmiss))# how aggregate a column are
apply(df_old, 1, pmiss)# how aggregate a record are
sum(complete.cases(df))#how mant cases are totally complete.


#visualize the missing
aggr_plot <- aggr(df,col=c('blue','red'),numbers=TRUE,prop=TRUE,
sortVars=TRUE,labels=names(df),ylab=c('histogram of missing
data','pattern'),cex.axis=0.3,gap=0)
md.pattern(df)



#  check whether the missing value is MCAR(missing value completely are ramdom) or
MNAR(missing value are not random)
```

```
marginplot(df[,c('visitor_hist_starrating',"visitor_hist_adr_usd")])# left red boxplot: the distribution of hist_adr when starrating is missing; blue boxplot: the distribution of hist_adr when starrating is not missing
#interesting thing found : ??
# let's check whether when adr_used is missing, his_starrating is missing too!
number=0
for( i in 1:nrow(df)){
  if (is.na(df$visitor_hist_starrating[i])){
    if (is.na(df$visitor_hist_adr_usd[i])){
      number =number +1
    }
  }
}
print(number)
# we found the number is 94542 which is equal to the number of missing in hist_adr

#distribution of each variable which has missing value
#visitor_hist_starrating
ggplot(df,aes(x=visitor_hist_starrating))+geom_bar(color='white',stat = 'count',binwidth = 0.2,aes(y=..density..))+ geom_density(color='red',bw=0.2)+ggtitle('visitor_hist_starrating distribution')+theme(plot.title = element_text(hjust = 0.5))
#visitor_hist_adr_usd
ggplot(df,aes(x=visitor_hist_adr_usd))+ geom_bar(color='white',stat = 'count',binwidth = 50,aes(y=..density..)) +geom_density(color='red',bw=50)+ggtitle('visitor_hist_adr_usd distribution')+theme(plot.title = element_text(hjust = 0.5))
#srch_query_affinity_score
ggplot(df,aes(x=srch_query_affinity_score))+ geom_bar(color='white',stat = 'count',binwidth = 5,aes(y=..density..)) +geom_density(color='red',bw=5)+ggtitle('srch_query_affinity_score distribution')+theme(plot.title = element_text(hjust = 0.5))
#orig_destination_distance
ggplot(df,aes(x=orig_destination_distance))+ geom_bar(color='white',stat = 'count',binwidth = 1000,aes(y=..density..)) +geom_density(color='red',bw=1000)+ggtitle('orig_destination_distance distribution')+theme(plot.title = element_text(hjust = 0.5))
#gross_bookings_usd
ggplot(df,aes(x=gross_bookings_usd))+ geom_bar(color='white',stat = 'count',binwidth = 500,aes(y=..density..)) +geom_density(color='red',bw=1000)+ggtitle('orig_destination_distance distribution')+theme(plot.title = element_text(hjust = 0.5))


# boxplot of categrical data
ggplot(df,aes(x=factor(promotion_flag),y=prop_log_historical_price))+geom_boxplot()+ggtitle('flag vs hist_price')+theme(plot.title = element_text(hjust = 0.5))
```

```
ggplot(df,aes(x=factor(prop_brand_bool),y=prop_log_historical_price))+geom_boxplot()+ggtitle('
flag vs hist_price')+theme(plot.title = element_text(hjust = 0.5))

ggplot(df,aes(x=factor(srch_saturday_night_bool),y=visitor_hist_starrating))+geom_boxplot()+gg
title("saturday vs hist_star")+theme(plot.title = element_text(hjust = 0.5))

ggplot(df,aes(x=factor(srch_saturday_night_bool),y=visitor_hist_starrating))+geom_boxplot()+gg
title('flag vs hist_price')+theme(plot.title = element_text(hjust = 0.5))
# correlation
ggplot(df,aes(x=prop_id,y=visitor_hist_starrating)) + geom_jitter()

ggplot(df,aes(x=site_id,y=visitor_hist_starrating)) + geom_jitter()

ggplot(df,aes(x=prop_country_id,y=visitor_hist_starrating)) + geom_jitter()

df$visitor_hist_starrating =replace(df$visitor_hist_starrating,is.na(df$visitor_hist_starrating),'')

#filling the his_starrating & hist_adr_usd
md.pattern(df)

# dealing with missing value
df <- fread('train.csv',header = TRUE,na.strings = c('NULL'))
drop1 <-
c('srch_id','date_time','position','prop_id','srch_destination_id','visitor_location_country_id','prop_
country_id',"comp1_inv","comp2_inv","comp3_inv","comp4_inv","comp5_inv","comp6_inv","com
p7_inv","comp8_inv")
df_1 <- fread('train.csv',drop =drop1,nrows = 100000,header = TRUE,na.strings = c('NULL'))
summary(df_1)

# For comp1:
for (i in length(df_1$comp1_rate)){
  if (is.na(df_1$comp1_rate[i]) & is.na(df_1$comp1_rate_percent_diff[i])== TRUE){
    df_1$comp1_rate[i] = 0
    df_1$comp1_rate[i] = 0
  }else if (df_1$comp1_rate[i]==1 & is.na(df_1$comp1_rate_percent_diff[i])==FALSE){
    df_1$comp1_rate_percent_diff[i] ==45.57296
  }else if(df_1$comp1_rate[i]==-1 & is.na(df_1$comp1_rate_percent_diff[i])==FALSE){
    df_1$comp1_rate_percent_diff[i] ==14.41436
  }
}

summary(df_1)
# positive agv
```

```
sqldf('select comp1_rate,comp1_rate_percent_diff from df_1 where comp1_rate is not null')
not_null <- sqldf('select comp1_rate,comp1_rate_percent_diff from df_1 where comp1_rate is
not null and comp1_rate_percent_diff is not null')
positive_avg1 <- sqldf('select avg(comp1_rate_percent_diff) from df_1 where comp1_rate ==1
and comp1_rate_percent_diff is not null')
negetive_avg1 <- sqldf('select avg(comp1_rate_percent_diff) from df_1 where comp1_rate ==-1
and comp1_rate_percent_diff is not null')
zero_avg1 <- sqldf('select avg(comp1_rate_percent_diff) from df_1 where comp1_rate ==0 and
comp1_rate_percent_diff is not null')

summary(sqldf('select comp1_rate,comp1_rate_percent_diff from df_1 where comp1_rate==0'))

summary(sqldf('select comp1_rate,comp1_rate_percent_diff from df_1 where comp1_rate is
null'))

sqldf('select comp1_rate,comp1_rate_percent_diff from df_1 where comp1_rate==0')
example <- data.frame(a=c(1,2,3),b=c(NA,1,NA))
example

for (i in 1:length(example$b)){
  if (is.na(example$b[i] & is.na())){
    example$b[i] <- 9
  }
}

for (i in 1:length(df_1$comp1_rate_percent_diff)){
  if(is.na(df_1$comp1_rate[i]) & is.na(df_1$comp1_rate_percent_diff[i])){
    df_1$comp1_rate[i] <- 0
    df_1$comp1_rate_percent_diff[i] <- 0
  }else if (df_1$comp1_rate[i]==0 & is.na(df_1$comp1_rate_percent_diff[i])){
    df_1$comp1_rate_percent_diff[i] <- 19
  }
}



# PCA CFA LOGI LDA
#rename the column and organized the datatype
draft <- read.csv('cleaned.csv',header = TRUE)
draft1 <- draft[,-1]

colnames(draft1) <-c('H_PurStar','H_PurPrice','HotelStar','HotelReviewScore','C?I','Location1',
                     'Location2','LogHotelP','DisplayP','Promotion?','StayLength','HStay','NAdult',
```

```
                'NKid','NRoom','Saturday?','RamdomDisp?','Click?','Y','CompePD')
draft2 <- draft1[draft1$H_PurPrice!= 0 & draft1$HotelStar!=0,]
draft2$H_PurStar[draft2$H_PurStar >=1 & draft2$H_PurStar <2] <- 1
draft2$H_PurStar[draft2$H_PurStar >=2 & draft2$H_PurStar <3] <- 2
draft2$H_PurStar[draft2$H_PurStar >=3 & draft2$H_PurStar <4] <- 3
draft2$H_PurStar[draft2$H_PurStar >=4 & draft2$H_PurStar <5] <- 4
draft2$H_PurStar[draft2$H_PurStar >=5 & draft2$H_PurStar <6] <- 5
head(draft2)
summary(draft2)


numericMatrix <- draft2[,c(2,6:9,11:15,20)]
scaledNumeric <- data.frame(scale(numericMatrix, center = FALSE, scale =
apply(numericMatrix, 2, sd, na.rm = TRUE)))
ordinalMatrix <- draft2[,c(1,3,4,5,10,16,17,18)]
Y <- draft2[,19]

head(scaledNumeric)
head(numericMatrix)
head(ordinalMatrix)

# compute correlation matrix for both numeric and ordinal data
library(corrplot)
c <- cor(numericMatrix)
round(c,2)
Pearson <- hetcor(numericMatrix)
round(Pearson$correlations,2)

cS <- cor(ordinalMatrix,method = 'spearman')
round(cS,2)

ord.f = sapply(ordinalMatrix, as.factor)
Y.f=as.factor(Y)

head(ord.f)
Polychoric <- hetcor(ord.f)$cor
round(Polychoric,2)

mixmatrx <- cbind(scaledNumeric,ord.f)
head(mixmatrx)

Mixed <- hetcor(mixmatrx)
cm <-Mixed$correlations[-7,-7]
```

```
round(cm,2)

# run CFA on mixed correlation matrix
f = factanal(covmat = cm, factors = 10)
print(f$loadings, cutoff = .25)
f$rotmat

# run pca on numeric correlation matrix
pcaN = psych::principal(scaledNumeric,nfactors=4, scores=TRUE)
print(pcaN$loadings, cutoff=.45, sort=T)
head(pcaN$scores)

# run CFA on ordinal variable matrix
pcaO = factanal(covmat = cS, factors = 5)


# commpute logistic regression
install.packages('caTools')
library(caTools)
set.seed(88)
newdata <- cbind(scaledNumeric,ord.f,Y.f)
head(newdata)
# fit in the model based on pca
new <- cbind(data.frame(pcaN$scores),ord.f,Y.f)

Y0 <-new[new$Y.f==0,]
Y1 <-new[new$Y.f==1,]
nY0 <- Y0[sample(nrow(Y0),5292),]

equal <-rbind(Y1,nY0)

split <- sample.split(equal$Y.f, SplitRatio = 0.75)
train <- subset(equal, split == TRUE)
test <- subset(equal, split == FALSE)

model1 <- glm(formula = Y.f ~ RC1 + RC2 + RC3 + RC4 + H_PurStar + HotelStar +
HotelReviewScore+
              `C?I` + `Promotion?` + `Saturday?` + `RamdomDisp?`,
            family = binomial(link = "logit"), data = train)
summary(model1)
#stepwise
stepAIC(model1,direction = 'both')
model2 <- glm(formula = Y.f ~ RC1 + RC3 + RC4 + HotelStar + `C?I` + `Promotion?` +
```

```
        `RamdomDisp?`, family = binomial(link = "logit"), data = train)
summary(model2)
#backward
stepAIC(model1,direction = 'backward') # same as model2

#forward
stepAIC(model1,direction = 'forward') # same as model1

# test on test set
library(aod)
# model1
fitted.results1 <- predict(model1,newdata=subset(test,select=c(1:11)),type='response')
fitted.results1 <- ifelse(fitted.results1 > 0.5,1,0)

misClasificError1 <- mean(fitted.results1 != test$Y.f)
print(paste('Accuracy',1-misClasificError1))
# model2
fitted.results2 <- predict(model2,newdata=subset(test,select=c(2,3,4,6,8,9,11)),type='response')
fitted.results2 <- ifelse(fitted.results2 > 0.5,1,0)

misClasificError2 <- mean(fitted.results2 != test$Y.f)
print(paste('Accuracy',1-misClasificError2))
table(test$Y.f,fitted.results2)

# compare the two model
anova(model2,model1,test='Chisq')

# explain the coefficent
exp(coef(model2))
exp(coef(model1))

confint(model1)
confint(model2)

# test the each values probability
# interpretation for C?I
testdata <- data.frame( `C?I` =c(0,1),RC1=mean(train$RC1),RC3=mean(train$RC3),
            RC4=mean(train$RC4),`Promotion?`=1,`RamdomDisp?`=1,HotelStar=3)
colnames(testdata)<-c('C?I','RC1','RC3','RC4','Promotion?','RamdomDisp?','HotelStar')
testdata$`C?I`<-as.factor(testdata$`C?I`)
testdata$`Promotion?`<-as.factor(testdata$`Promotion?`)
testdata$`RamdomDisp?`<-as.factor(testdata$`RamdomDisp?`)
testdata$HotelStar<-as.factor(testdata$HotelStar)
```

```
testdata$prob <- predict(model2,newdata=testdata,type='response')
head(testdata)

# interpretation for Promotion?
testdata1 <- data.frame( `Promotion?`=c(0,1),RC1=mean(train$RC1),RC3=mean(train$RC3),
                         RC4=mean(train$RC4),`C?I`=1,`RamdomDisp?`=1,HotelStar=3)
colnames(testdata1)<-c('Promotion?','RC1','RC3','RC4','C?I','RamdomDisp?','HotelStar')
testdata1$`C?I`<-as.factor(testdata1$`C?I`)
testdata1$`Promotion?`<-as.factor(testdata1$`Promotion?`)
testdata1$`RamdomDisp?`<-as.factor(testdata1$`RamdomDisp?`)
testdata1$HotelStar<-as.factor(testdata1$HotelStar)
testdata1$prob <- predict(model2,newdata=testdata1,type='response')
head(testdata1)

#interpretation for hotelstar
testdata2 <- data.frame( HotelStar=c(1,2,3,4,5),RC1=mean(train$RC1),RC3=mean(train$RC3),
                         RC4=mean(train$RC4),`C?I`=1,`RamdomDisp?`=1,`Promotion?`=1)
colnames(testdata2)<-c('HotelStar','RC1','RC3','RC4','C?I','RamdomDisp?','Promotion?')
testdata2$`C?I`<-as.factor(testdata2$`C?I`)
testdata2$`Promotion?`<-as.factor(testdata2$`Promotion?`)
testdata2$`RamdomDisp?`<-as.factor(testdata2$`RamdomDisp?`)
testdata2$HotelStar<-as.factor(testdata2$HotelStar)
testdata2$prob <- predict(model2,newdata=testdata2,type='response')
head(testdata2)

# dispersion
deviance(model2)/df.residual(model2)
fit.od <- glm(formula = Y.f ~ RC1 + RC3 + RC4 + HotelStar + `C?I` + `Promotion?` +
                `RamdomDisp?`, family = quasibinomial(link = "logit"), data = train)

pchisq(summary(fit.od)$dispersion * model2$df.residual,model2$df.residual,lower.tail = F )


# descrimata analysis
library(MASS)
LDA =
lda(Y.f~RC1+RC2+RC3+RC4+H_PurStar+HotelStar+HotelReviewScore+`C?I`+`Promotion?`+`
Saturday?`+`RamdomDisp?`+`Click?`, data=train)
print(LDA)
p = predict(LDA, newdata=subset(test,select = c(1:12)))$class
table(test$Y.f,p)

# mahalanobis distance
```

```
head(train)
head(test)

target <- test[,1:11]
G0 <- train[train$Y.f==0,1:11]
G1 <- train[train$Y.f==1,1:11]
G0[5:11] <- apply(G0[5:11],2,as.numeric)
G1[5:11] <- apply(G1[5:11],2,as.numeric)
target[5:11] <- apply(target[5:11],2,as.numeric)

M0 <- colMeans(G0)
M1 <- colMeans(G1)
V0 <- cov(G0)
V1 <- cov(G1)

Dis0 <- mahalanobis(target,M0,V0)
Dis1 <- mahalanobis(target,M1,V1)
diff <- Dis0 - Dis1
result <- ifelse(diff>0,1,0)
table(test$Y.f,result)

# Bayes
V <- exp(-0.5*Dis1 + 0.5*Dis0)
d <- 0.5
result1 <- ifelse(V>d,1,0)
table(test$Y.f,result1)
```

Python code

```
# -*- coding: utf-8 -*-
"""
Created on Mon May 29 16:47:50 2017

@author: Yimei Tang
"""
##Import Libraries
import sklearn
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
```

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.cross_validation import train_test_split,cross_val_score
from sklearn.metrics import classification_report,
precision_score,recall_score,roc_curve,auc,confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.grid_search import GridSearchCV
from sklearn.grid_search import RandomizedSearchCV
from sklearn.ensemble import ExtraTreesClassifier
from sklearn import cross_validation,metrics
from sklearn.feature_selection import SelectFromModel


#Load Data
train_df=pd.read_csv('C:\\Users\\Yimei Tang\\Desktop\\DePaul\\CSC 424 Advanced Data
Analysis\\Homework\\Homework 3\\train_prePCA.csv',
                     index_col=0)

row,column=train_df.shape
##Preview Data
train_df.columns
row,column=train_df.shape
row,column


##Convert categorical varialbes to numerical by using OneHotEncoder
train_df.dtypes
train_df.info()
train_df=pd.get_dummies(train_df, columns=['Month','TofD','TraMon','RG','Condition','Type'])

##Use MinMaxScaler
mm=MinMaxScaler()
train_df['price_usd']=mm.fit_transform(train_df['price_usd'].reshape(-1,1))


##Seperate Click_bool as Y1 and booking_bool as Y2
Y1=train_df['click_bool'].as_matrix()
Y2=train_df['booking_bool'].as_matrix()
train_df=train_df.drop(['click_bool','booking_bool'], 1)
X=train_df.as_matrix()
```

```
##Check any null values
np.isnan(np.sum(X))

###Seperate Train and Test Set
x_train,x_test,y_train,y_test = train_test_split(X,Y1,test_size=0.2,random_state=1)




#####Decision Tree
parameters={'class_weight':['balanced'],
            'max_depth':list(range(3, 15)),
            'min_samples_split' :list(range(3, 15)),
            'max_features': ['auto','sqrt','log2',None]
            }
clf=DecisionTreeClassifier()
gs=GridSearchCV(clf,param_grid=parameters,scoring='f1')
gs.fit(x_train,y_train)
gs.best_score_
gs.best_estimator_
gs.scorer_


###Cross Validation
cvs=cross_val_score(est,x_train,y_train,cv=10,scoring='f1')
cvs


predictions=est.predict(x_test)
confusion_matrix(y_test,predictions)
print(classification_report(y_test,predictions))
dtpfsf1='%0.2f' % sklearn.metrics.f1_score(y_test,predictions)
dtpfsrc='%0.2f' % sklearn.metrics.recall_score(y_test,predictions)


##Feature Importance
imp = est.feature_importances_
names=train_df.columns
imp,names=zip(*sorted(zip(imp,names)))
plt.figure(figsize=(20,20))
plt.barh(range(len(names)),imp,align='center')
plt.yticks(range(len(names)),names)
```

```
###ExtraTreesClassifier
forest= ExtraTreesClassifier(n_estimators=500, class_weight='balanced')
forest.fit(x_train,y_train)
imp2 = forest.feature_importances_
imp2,names=zip(*sorted(zip(imp2,names)))
plt.figure(figsize=(20,20))
plt.barh(range(len(names)),imp2,align='center')
plt.yticks(range(len(names)),names)

###Use the important features
sel = SelectFromModel(forest, prefit=True)
x_new_train = sel.transform(x_train)
x_new_test = sel.transform(x_test)

feature_idx = sel.get_support()
feature_name = train_df.columns[feature_idx]
feature_name


###After selecting the important features:
###Results are much better
gs.fit(x_new_train,y_train)
gs.best_score_
gs.best_estimator_
gs.scorer_
est=gs.best_estimator_

cvs=cross_val_score(est,x_new_train,y_train,cv=10,scoring='f1')


predictions=est.predict(x_new_test)
confusion_matrix(y_test,predictions)
print(classification_report(y_test,predictions))
dtafsf1='%0.2f' % sklearn.metrics.f1_score(y_test,predictions)
dtafsrc='%0.2f' % sklearn.metrics.recall_score(y_test,predictions)



###Random Forest
parameters={'max_depth':[10,20],
            'min_samples_split' :[3,7],
            'class_weight':['balanced'],
            'n_estimators':[300,310]}
```

```
clf=RandomForestClassifier()
rs=RandomizedSearchCV(clf,param_distributions=parameters,scoring='f1',n_iter=4)
rs.fit(x_train,y_train)

rs.best_score_
rs.best_estimator_
rs.scorer_
est=rs.best_estimator_

###Cross Validation
cvs=cross_val_score(est,x_train,y_train,cv=10,scoring='f1')
cvs


predictions=est.predict(x_test)
confusion_matrix(y_test,predictions)
print(classification_report(y_test,predictions))
rfbfsf1='%0.2f' % sklearn.metrics.f1_score(y_test,predictions)
rfbfsrc='%0.2f' % sklearn.metrics.recall_score(y_test,predictions)


###After selecting the important features:
rs.fit(x_new_train,y_train)
rs.best_score_
rs.best_estimator_
rs.scorer_
est=rs.best_estimator_

predictions=est.predict(x_new_test)
confusion_matrix(y_test,predictions)
print(classification_report(y_test,predictions))

rfafsf1='%0.2f' % sklearn.metrics.f1_score(y_test,predictions)
rfafsrc='%0.2f' % sklearn.metrics.recall_score(y_test,predictions)



###KNN
parameters={'n_neighbors':list(range(2, 5)),
        'weights':['uniform','distance'],
        'algorithm' :['auto', 'ball_tree', 'kd_tree', 'brute'],
        }
clf=KNeighborsClassifier()
```

```
gs=GridSearchCV(clf,param_grid=parameters,scoring='f1')
gs.fit(x_train,y_train)
gs.best_score_
gs.best_estimator_
gs.scorer_
est=gs.best_estimator_

###Cross Validation
cvs=cross_val_score(est,x_train,y_train,cv=10,scoring='f1')
cvs

predictions=est.predict(x_test)
confusion_matrix(y_test,predictions)
print(classification_report(y_test,predictions))
knnppcaf1='%0.2f' % sklearn.metrics.f1_score(y_test,predictions)
knnppcarc='%0.2f' % sklearn.metrics.recall_score(y_test,predictions)



###Use AfterPCA
#Load Data
train_df2=pd.read_csv('C:\\Users\\Yimei Tang\\Desktop\\DePaul\\CSC 424 Advanced Data
Analysis\\Homework\\Homework 3\\train_afterPCA.csv',
                  index_col=0)

##Convert categorical varialbes to numerical by using OneHotEncoder
train_df2=pd.get_dummies(train_df2, columns=['Month','TofD','TraMon','RG','Condition','Type'])


##Seperate Click_bool as Y1 and booking_bool as Y2
Y1=train_df2['click_bool'].as_matrix()
Y2=train_df2['booking_bool'].as_matrix()
train_df2=train_df2.drop(['click_bool','booking_bool'], 1)
X=train_df2.as_matrix()

###Seperate Train and Test Set
x_train,x_test,y_train,y_test = train_test_split(X,Y1,test_size=0.2,random_state=1)



clf=KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
          metric_params=None, n_jobs=1, n_neighbors=2, p=2,
          weights='distance')
```

```
###Cross Validation
clf.fit(x_train,y_train)
cvs=cross_val_score(clf,x_train,y_train,cv=10,scoring='f1')
cvs


predictions=clf.predict(x_test)
confusion_matrix(y_test,predictions)
print(classification_report(y_test,predictions))
knnapcaf1='%0.2f' % sklearn.metrics.f1_score(y_test,predictions)
knnapcarc='%0.2f' % sklearn.metrics.recall_score(y_test,predictions)

#####Conclusion
threemodels = pd.DataFrame({
      'Model': ['Decision Tree Before Feature Selection',
            'Decison Tree After Feature Selection',
            'Random Forest Before Feature Selection',
            'Random Forest After Feature Selection',
            'KNN Beore PCA Feature Selection',
            'KNN After PCA Feature Selection'],
      'F1_score': [dtpfsf1,dtafsf1,rfbfsf1,rfafsf1,knnppcaf1,knnapcaf1],
      'Recall_score': [dtpfsrc,dtafsrc,rfbfsrc,rfafsrc,knnppcarc,knnapcarc],
      'Numbers of Features': [30,20,30,20,30,20]})

pd.set_option('display.height', 1000)
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)


threemodels.sort_values(by='F1_score',ascending=False)
```