Understanding A Traveler's Hotel Booking Pattern

Project Team Members: Monika Baloda, Charu Joshi, Sakshee Vaidya

A. Gary Anderson Graduate School of Management, University of California, Riverside

STATISTICAL COMPUTING STAT 206

Business Report on Final Project

Winter Quarter 2023

CONTRIBUTION OF TEAM MEMBERS

- **Monika Baloda:**
  1. Implemented pattern recognition methodologies using PCA and Cluster Analysis.
  2. Theory, Coding, and Inferences of variable selection techniques (LASSO and Random Forest)
- **Charu Joshi :**
  1. Data acquisition & management 2. Introduction, Project Evaluation & Conclusion
- **Sakshee Vaidya :**
  1. Exploratory Data Analysis (EDA) using python in Google collab
  2. Data visualization & management

Disclaimer: This report is the final project report required in STAT206 : Statistical Computing, a graduate level course offered at A. Gary Anderson Graduate School of Management, University of California, Riverside. This report uses the Expedia company's public data to research some questions. The Expedia logo used here is just to make this report look like a business report, this doesn't represent any association with Expedia company. This report is purely prepared for educational purposes and no commercial purpose is served through it.

EXECUTIVE SUMMARY

Title: - Understanding A Traveler's Hotel Booking Pattern

About the Company: -

Expedia Inc. is the world's largest online travel company owned by the Expedia Group, based in Seattle. Expedia today has Accomodation which includes over 350k Hotels and 1.2M Vacation rentals.

Expedia Dataset: -

Number of Rows - 1048575

Number of Columns - 27+146(from dest data)

Proposal: - Trying to predict a booking pattern for hotel rooms depending on other factors mentioned in the dataset.

Problem Statement: - Understanding a traveler's journey of selecting & booking a hotel.

- Who books more and how?
- How do different traveler segments behave?
- What determines hotel booking?

Methodologies: -

- *Principal Component Analysis (PCA)* - To maximize variation within a few independent variables.
- *Cluster Analysis* - To find homogeneous subgroups among the observations
- *LASSO Regression* - Variable selection technique.
- *Random Forest*-To find importance of variables and accuracy of final model

Conclusion & Recommendations- Distance of destination, hotel location, hotel rating, and user history are the most important factors determining hotel booking on Expedia. Understanding the main factors can influence bookings in a particular context, can help businesses and organizations to tailor their offerings and marketing strategies to better meet the needs.

INDEX

1.INTRODUCTION

1.1 ABOUT THE COMPANY (Expedia Inc.)

- Expedia Inc. is the world's largest online travel company owned by the Expedia Group, based in Seattle.
- Expedia today has Accomodation which includes over 350k Hotels and 1.2M Vacation rentals.
- They use machine learning techniques to build hotel recommendation and ranking systems and hotel revenue management algorithms.

1.2 ABOUT THE DATASET (Expedia Dataset)

Expedia (World's largest online travel agency) was the data sponsor of 'American Statistical Association Datafest' for the year 2017.

● Expedia today has Accomodation which includes over 350k Hotels and 1.2M Vacation rentals, from which the dataset we'll be using for the project comes.

● The Analytic team which generates the dataset is based in Geneva, Switzerland where the organization that manages the relationship and business with hotels is based.

● They use machine learning techniques to build hotel recommendation and ranking systems and hotel revenue management algorithms.

● Expedia consists a number of leading travel brands which includes:

➔ Hotels.com - Hotel specialist site

➔ Homeaway - Leader in vacation rentals

1.3 PROJECT OBJECTIVES

- PROPOSAL - Trying to predict a booking pattern for hotel rooms depending on the other factors mentioned in the dataset.
- PROBLEM STATEMENT - Understanding a traveler's journey of selecting & booking a hotel.
- METHODOLOGY -

★ **Principal Component Analysis (PCA)** - To emphasize variation and bring out strong patterns in a dataset.

★ **Cluster Analysis** - To find homogeneous subgroups among the observations

★ **LASSO Regression -** Encourages simple, sparse model, with few parameters, where data values are shrunk towards a central point as mean.

1.4 WHY EXPLORING THE DATA IS IMPORTANT?

- Data exploration is important because it allows us a deeper understanding of the dataset, making it easier to understand, navigate and use the data further for analysis.
- We will be using both R and Python for the project work, where the former will be best suited for statistical learning and the latter will be used for complete analysis.
- Our data exploration would typically include the following three steps: -

1. Understand the Variables
2. Detect the Outliers
3. Examine patterns and relationships

2. EXPLORATORY DATA ANALYSIS

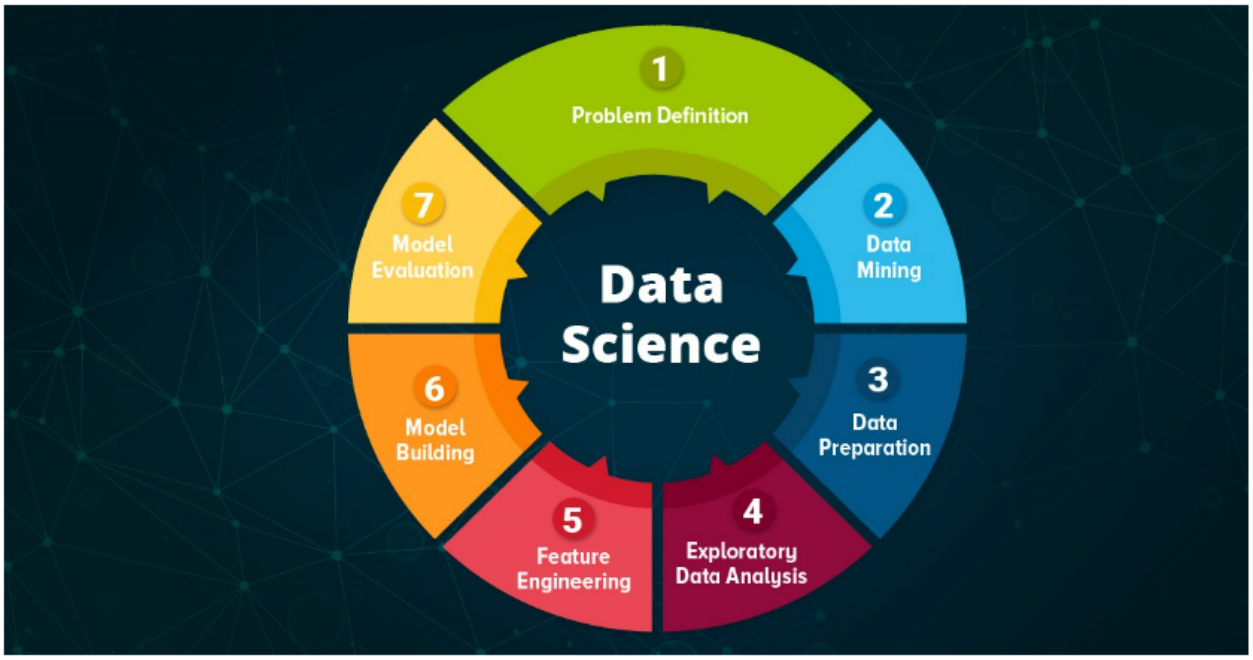
Objective

Exploratory Data Analysis (EDA) is used to analyze, investigate datasets and summarize their main characteristics, often employing data visualization methods.

2.1 The Data Preparation

Data preparation is the process of the gathering, combining, structuring and organizing of raw data which then can be used for processing and analysis.
Some of the steps involved in the data preparation are:

1. *Gather data*- this step involves the collection of the appropriate data from the source.
2. *Clean the data set*- Cleaning of the data set corrects errors and fills in the missing data as a step to ensure data quality. This also involves transforming the data into consistent and readable format.
3. *Data structuring*- this involves modeling the data set and organizing it to meet the analytics requirements. Like converting the text file into csv format.
4. *Store data*- the data set once cleaned can be stored in desired format and used for any analytical application.

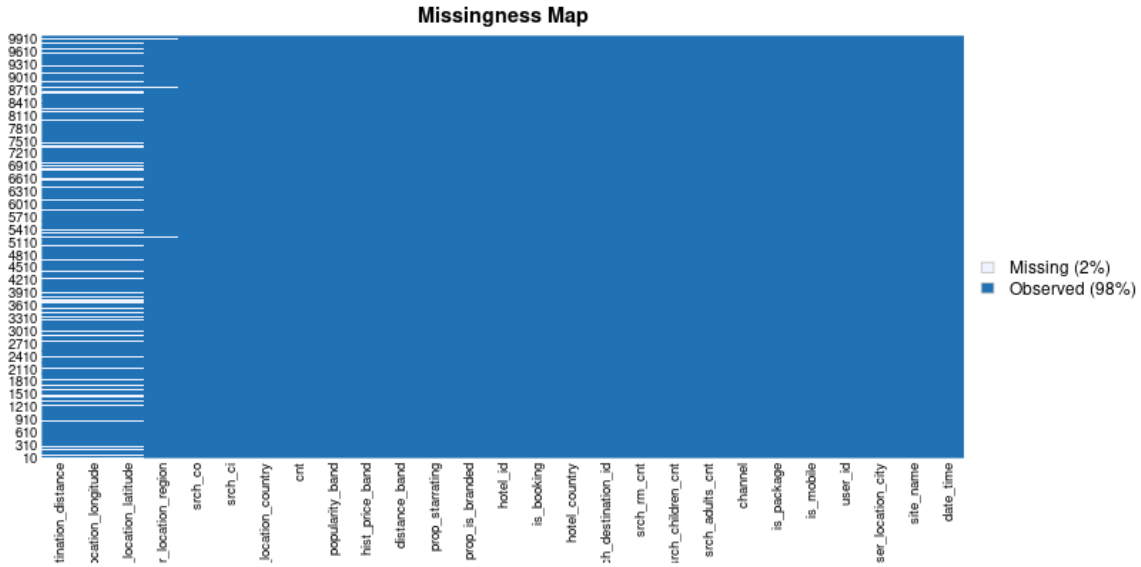


2.1.1 Summary tables

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 27 columns):
 #   Column                     Non-Null Count    Dtype  
---  ------                     --------------    -----  
 0   date_time                  1048575 non-null  object 
 1   site_name                  1048575 non-null  object 
 2   user_location_country      1048556 non-null  object 
 3   user_location_region       1041443 non-null  object 
 4   user_location_city         1048575 non-null  object 
 5   user_location_latitude     830326 non-null   float64
 6   user_location_longitude    830326 non-null   float64
 7   orig_destination_distance  830326 non-null   float64
 8   user_id                    1048575 non-null  int64  
 9   is_mobile                  1048575 non-null  int64  
 10  is_package                 1048575 non-null  int64  
 11  channel                    1048575 non-null  int64  
 12  srch_ci                    1048082 non-null  object 
 13  srch_co                    1048081 non-null  object 
 14  srch_adults_cnt            1048575 non-null  int64  
 15  srch_children_cnt          1048575 non-null  int64  
 16  srch_rm_cnt                1048575 non-null  int64  
 17  srch_destination_id        1048575 non-null  int64  
 18  hotel_country              1048575 non-null  object 
 19  is_booking                 1048575 non-null  int64  
 20  hotel_id                   1048575 non-null  int64  
 21  prop_is_branded            1048575 non-null  int64  
 22  prop_starrating            1048575 non-null  int64  
 23  distance_band              1048575 non-null  object 
 24  hist_price_band            1048575 non-null  object 
 25  popularity_band            1048575 non-null  object 
 26  cnt                        1048575 non-null  int64  
dtypes: float64(3), int64(13), object(11)
```

Figure 1. The given Expedia data set with 27 columns its data types



Overall Missing Values	6225
Percentage of Missing Values	2.31 %
Rows with Missing Values	2052
Columns With Missing Values	7

Figure 2. Missing values are represented by the given plot.

2.2 Sampling

Total number of datapoints is 10 millions. To handle it with ease, we draw a random sample of 10000 data points.

Sampling is performed on large datasets for several reasons:

- *Efficiency*: By taking a smaller representative sample, we can save time and resources while still obtaining useful insights.
- *Cost-effectiveness*: we can reduce costs while still obtaining useful insights by taking smaller sections of data.
- *Feasibility*: we can obtain useful insights while still working within the constraints of our resources.
- *Accuracy*: In some cases, sampling can actually lead to more accurate results than collecting data from the entire population. This is because a well-designed sample can avoid biases that might be present in the full population, such as selection bias or non-response bias.

NOTE- In the original given dataset we have 27 column with title and one column that is untitled, therefore after the sampling that column is also counted as 1 column and titled as #, that is why we have 28 columns after the sampling.

```
data_sampled = pd.read_csv('data.sample.csv')
data_sampled.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28706 entries, 0 to 28705
Data columns (total 28 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Unnamed: 0                 28706 non-null  int64
 1   user_id                    28706 non-null  int64
 2   date_time                  28706 non-null  object
 3   site_name                  28706 non-null  object
 4   user_location_country      28706 non-null  object
 5   user_location_region       28541 non-null  object
 6   user_location_city         28706 non-null  object
 7   user_location_latitude     23034 non-null  float64
 8   user_location_longitude    23034 non-null  float64
 9   orig_destination_distance  23034 non-null  float64
 10  is_mobile                  28706 non-null  int64
 11  is_package                 28706 non-null  int64
 12  channel                    28706 non-null  int64
 13  srch_ci                    28688 non-null  object
 14  srch_co                    28688 non-null  object
 15  srch_adults_cnt            28706 non-null  int64
 16  srch_children_cnt          28706 non-null  int64
 17  srch_rm_cnt                28706 non-null  int64
 18  srch_destination_id        28706 non-null  int64
 19  hotel_country              28706 non-null  object
 20  is_booking                 28706 non-null  int64
 21  hotel_id                   28706 non-null  int64
 22  prop_is_branded            28706 non-null  int64
 23  prop_starrating            28706 non-null  int64
 24  distance_band              28706 non-null  object
 25  hist_price_band            28706 non-null  object
 26  popularity_band            28706 non-null  object
 27  cnt                        28706 non-null  int64
dtypes: float64(3), int64(14), object(11)
```

Figure 3. Sampled Data set

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28706 entries, 0 to 28705
Data columns (total 28 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Unnamed: 0                  28706 non-null  int64
 1   user_id                     28706 non-null  int64
 2   date_time                   28706 non-null  object
 3   site_name                   28706 non-null  object
 4   user_location_country       28706 non-null  object
 5   user_location_region        28541 non-null  object
 6   user_location_city          28706 non-null  object
 7   user_location_latitude      23034 non-null  float64
 8   user_location_longitude     23034 non-null  float64
 9   orig_destination_distance   23034 non-null  float64
 10  is_mobile                   28706 non-null  int64
 11  is_package                  28706 non-null  int64
 12  channel                     28706 non-null  int64
 13  srch_ci                     28688 non-null  object
 14  srch_co                     28688 non-null  object
 15  srch_adults_cnt             28706 non-null  int64
 16  srch_children_cnt           28706 non-null  int64
 17  srch_rm_cnt                 28706 non-null  int64
 18  srch_destination_id         28706 non-null  int64
 19  hotel_country               28706 non-null  object
 20  is_booking                  28706 non-null  int64
 21  hotel_id                    28706 non-null  int64
 22  prop_is_branded             28706 non-null  int64
 23  prop_starrating             28706 non-null  int64
 24  distance_band               28706 non-null  object
 25  hist_price_band             28706 non-null  object
 26  popularity_band             28706 non-null  object
 27  cnt                         28706 non-null  int64
dtypes: float64(3), int64(14), object(11)
```

Figure 4. Sampled and cleaned dataset.

2.3 Summary Statistics

2.3.1 Data Exploration and Visualization

Boxplot of Standardized Variables

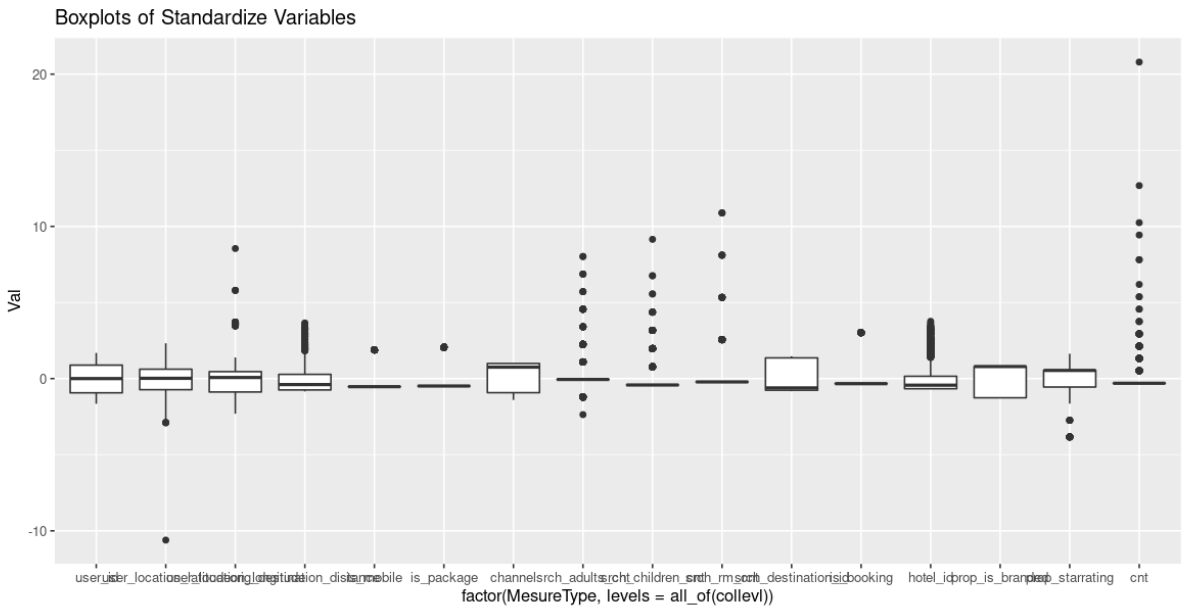


Figure5. Boxplot

We visualized the expedia data set using the boxplots. Here the box plot helps us to represent the interquartile range, or the middle half of the values in each group. We can see that we have outliers in almost all the variables of our data set. The boxplots which have short boxes indicate that their data points consistently hover around the center values and taller boxes indicate we have more variable data, especially when we have similar median values.

Violin Plot of Standardized Input Data (Mean = 0, Variance = 1)

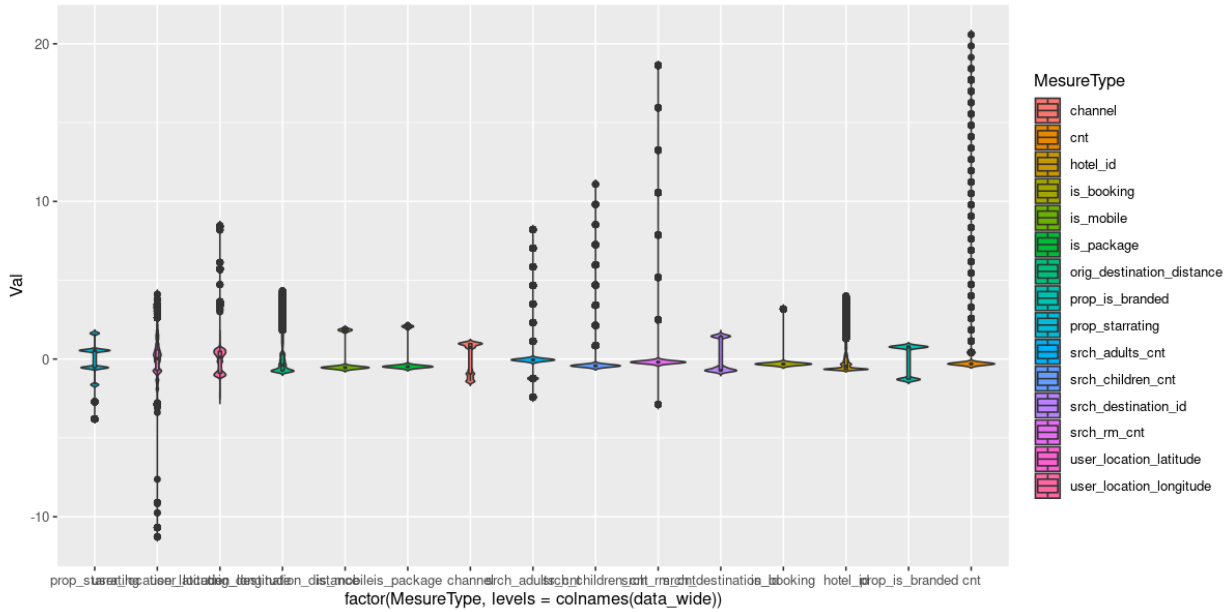
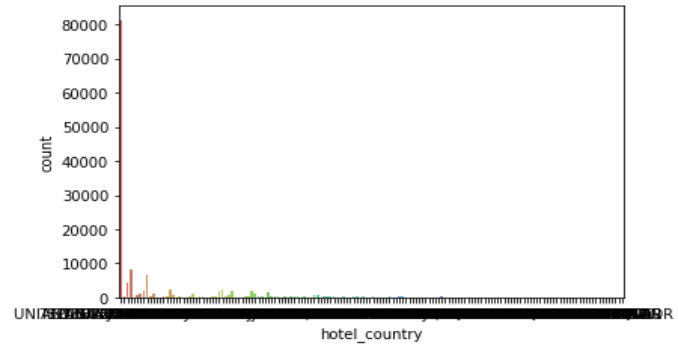


Figure6. Violin Plot

From the above violin plots, we are able to represent the peaks in the expedia data. We have used this to visualize the distribution of the numerical variables.




```
UNITED STATES OF AMERICA    81341
CANADA                       8266
MEXICO                       6492
ITALY                        4089
FRANCE                       2500
                            ...
NIGER                           2
GIBRALTAR                       1
MOZAMBIQUE                      1
KAZAKHSTAN                      1
DJIBOUTI                        1
Name: hotel_country, Length: 154, dtype: int64
```

Figure7. Number of bookings done in different countries before sampling.

We sampled the data from here and cleaned it so that we can have better visualizations so the outliers and missing values won't affect our results.

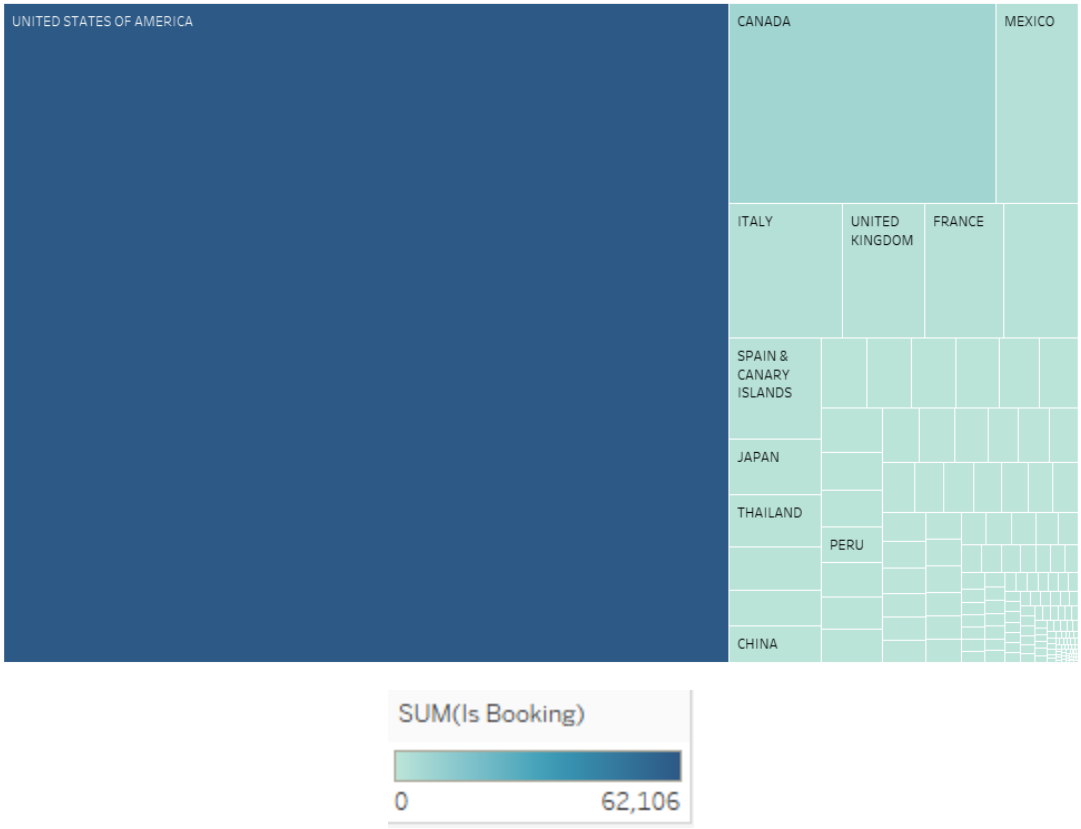
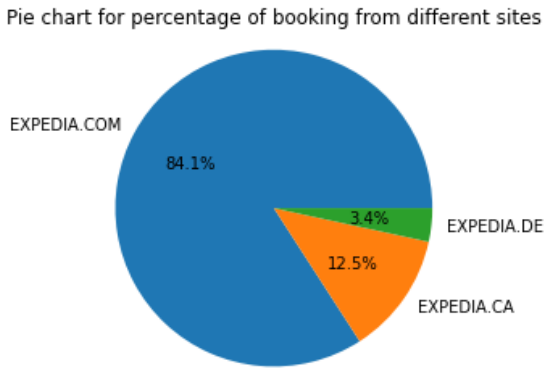
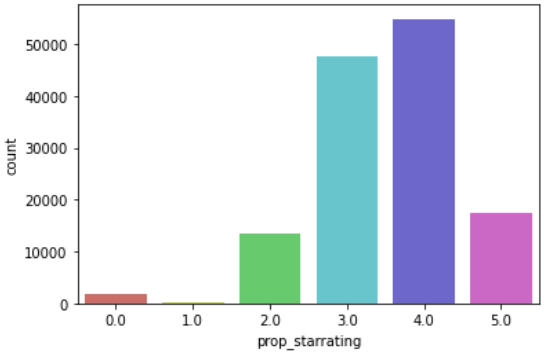


Figure 8. Booking density in different countries.

We can see from the above figure that we have the maximum number of bookings in the United States of America followed by Canada & Mexico.

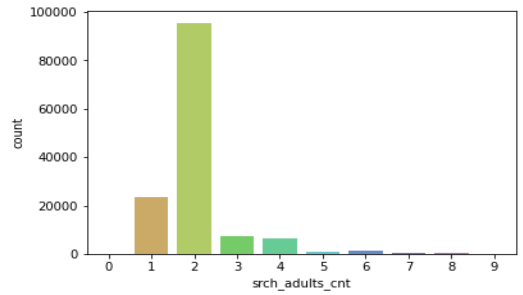




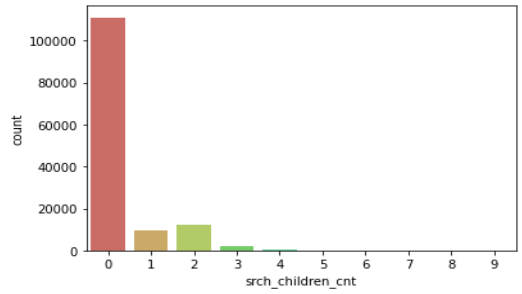

```
4.0    54818
3.0    47685
5.0    17528
2.0    13455
0.0     1886
1.0      283
Name: prop_starrating, dtype: int64
```

Fig9. Expedia point of sale, i.e., the site (Expedia.com is the US, Expedia.ca is Canada, Expedia.de is Germany, though Expedia.com is often used by people outside the US)

Fig10. The star rating of the hotel, from 1 to 5, in increments of 1. A 0 indicates the property has no stars, the star rating is not known or cannot be publicized



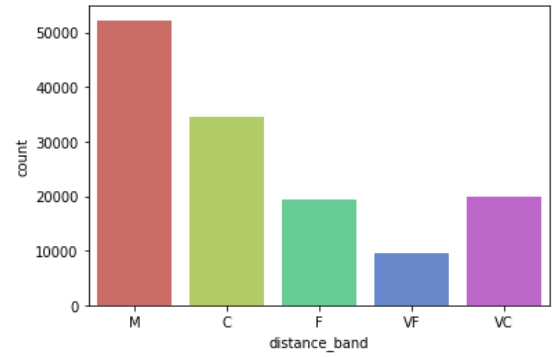

```
2    95496
1    23647
3     7371
4     6548
6     1167
5      746
8      287
7      245
0       75
9       73
Name: srch_adults_cnt, dtype: int64
```




```
0    111060
2     12111
1      9431
3      2282
4       536
5       107
6        91
8        19
7        14
9         4
Name: srch_children_cnt, dtype: int64
```

Fig11. The number of adults specified to occupy the hotel room

Fig12. The number of (optional) children specified to occupy the hotel room




```
M     52220
C     34580
VC    19923
F     19284
VF     9648
Name: distance_band, dtype: int64
```

Fig13. Banded distance of a hotel from the search destination center relative to other hotels in the same destination
VC = very close, C = close,
M = medium close, F = far,
VF = very far

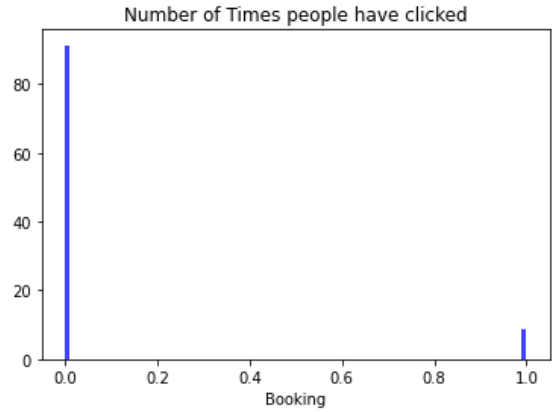


Fig14. 1 if the click/booking was generated as a part of a package search (i.e. a hotel combined with a flight and/or car rental), 0 otherwise

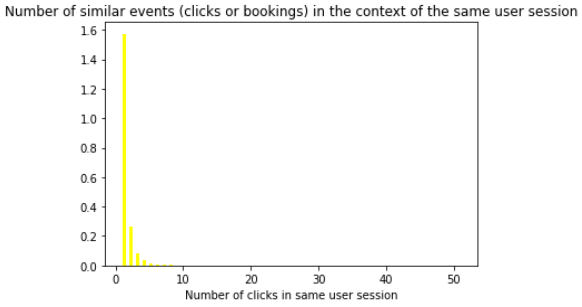


Figure15. Shows how many times similar events have taken place in which there are clicks or bookings in the same user session.

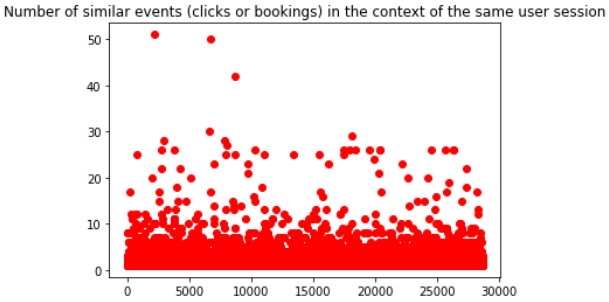


Figure16. Scatter plot showing the Number of similar events (clicks or bookings) in the context of the same user session

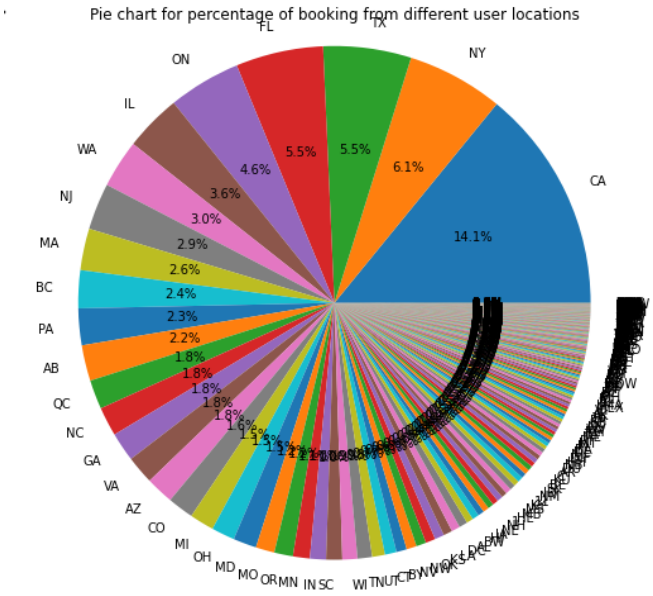


Figure17. Pie chart showing the user location region from where bookings are done

We have used Bar plots, pie charts, scatter plot and mosaic plot for the visualization purpose. From the above findings we can see that our main variable is_booking depends on a lot of the other variables.

2.3.2 Correlation Plots

To find a pattern & correlation coefficient values of 'is_booking' with the other variables, we plot a correlation plot so that we can predict which variables have the highest positive correlation, highest negative correlation and lowest positive, lowest negative correlation with 'is_booking'.

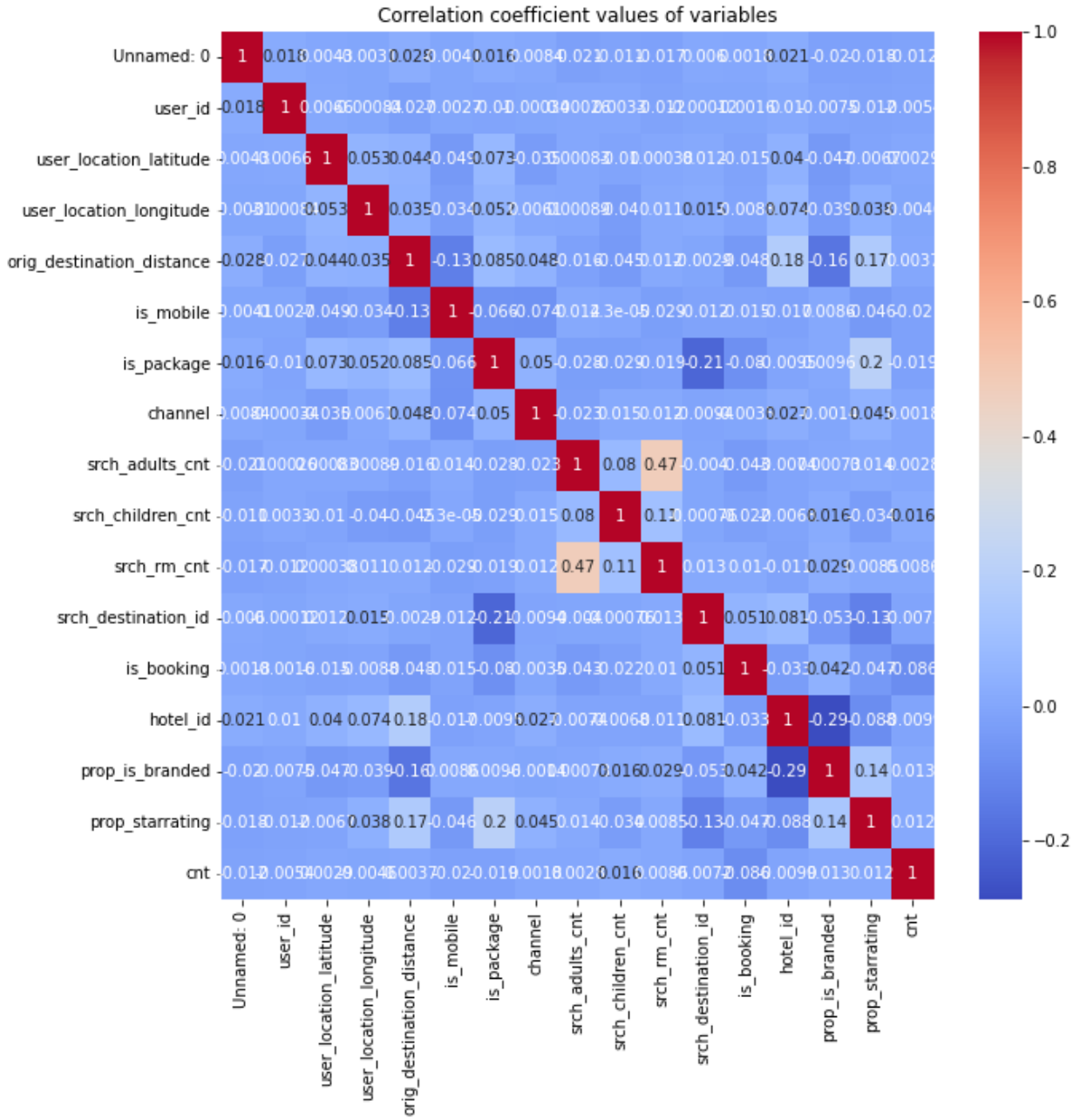


Figure18. Correlation Plot for all the variables of Expedia Data set.

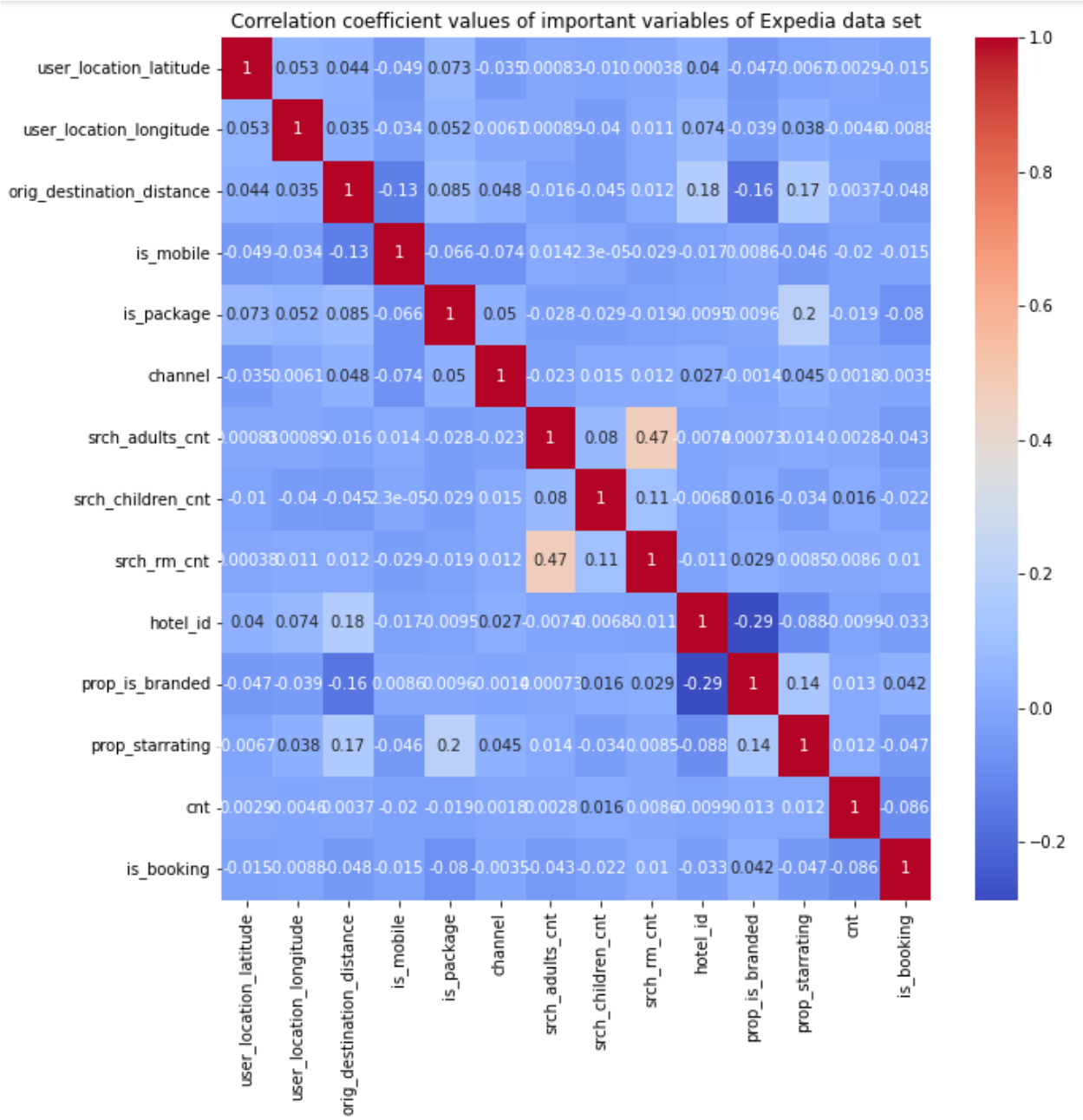


Figure19. Correlation coefficient values of important variables of Expedia data set

From the above mentioned Correlation Plot, we can see there are many correlation values in the plot. Darker shades represent higher correlation and lower represent low value of correlation coefficient irrespective of the directions. We couldn't find any positive or negative strong correlation in the Expedia data set. Therefore, now we will be merging the destination data set to

our expedia data set to apply the Principle Component Analysis (PCA) to find patterns and relationships.

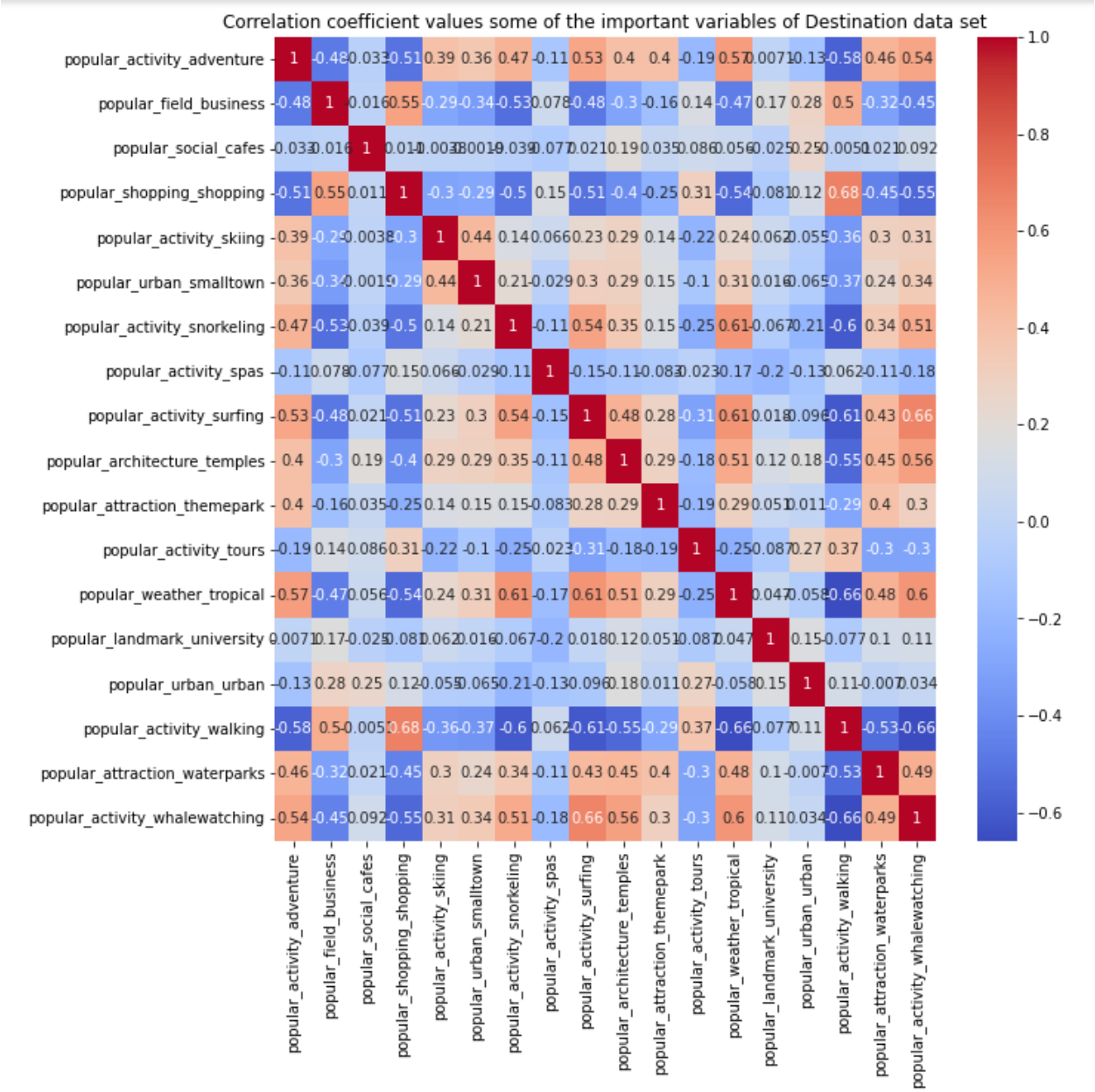


Figure20. Correlation coefficient values some of the important variables of Destination data set

3. MODEL EVALUATION & VALIDATION

3.1 Cluster Analysis

3.1.1. Objective : We want to explore groups-level patterns.

Exploring group-level patterns typically involves analyzing data from multiple individuals or entities that share similar characteristics or belong to the same group.

Our goal is to use the maximum amount of information to uncover natural grouping structure in the data. In the context of data analysis, this implies that we need to identify inherent patterns that occur within the data, instead of forcing preconceived ideas or assumptions about what those structures should be.

3.1.2 Narrowing Down Our Problem:

After merging the dataset we got 171 variables and to achieve this goal, we use statistical methods such as cluster analysis, dimensionality reduction, or factor analysis to identify underlying structures in the data. We can find the relationships between variables and identify groups or clusters of data points that share the similar characteristics. This approach may result in novel understandings of the data, which can assist us in making improved decisions.

The variables should be independent of each other, so that the distance calculation is based solely on their individual values and not on any relationship or interaction between them. This is important because if the variables are not independent, the distance measurement may not accurately reflect the true distance or dissimilarity between the data points being compared.

We can use PCA:

Through the use of principal component analysis (PCA), it is possible to identify a set of principal components that are independent of each other and capture the maximum amount of information from a given set of variables.

The goal of PCA is to transform a set of correlated variables into a set of uncorrelated variables, known as principal components, which explain the maximum amount of variance in the original data. The principal components are sorted in descending order of the amount of variance they explain, and a threshold of an eigenvalue greater than 1 is often used to select the most important principal components.

By selecting the principal components with eigenvalues greater than 1, we can focus on the components that explain the most variation in the data, while ignoring those that explain less.

This can be useful for reducing the dimensionality of the data and simplifying complex relationships among variables, while preserving the most important information in the data.
So in order to find group level patterns we do PCA first and then clustering.

PCA works by calculating the covariance matrix of the original variables and then finding the eigenvectors and eigenvalues of this matrix. The eigenvectors represent the directions of maximum variation in the data, and the eigenvalues represent the amount of variance that is explained by each eigenvector. The eigenvectors are then used to transform the data into a new set of variables, which are the principal components.

By selecting only the principal components that explain a high percentage of the total variance in the data, PCA can reduce the dimensionality of the dataset without losing too much information. This can be useful for data visualization, as it allows us to plot the data in a lower-dimensional space while still capturing the important patterns and relationships in the data. PCA is also commonly used in machine learning for feature extraction, as it can help to reduce the number of features in a dataset and improve the performance of models by removing redundant or irrelevant information.

These variables or features represent the maximum amount of information possible in a given number of variables. The larger the eigenvalue, the more variability is captured by the corresponding PC.
Principal Component Analysis (PCA):
PCs: are simply combination of all variables:
$PC1=a_1x_1 + a_2x_2 +.....+a_{170}x_{170}$
$PC2=b_1x_1 + b_2x_2 +.....+b_{170}x_{170}$
…….
$PC4=d_1x_1 + d_2x_2 +.....+d_{170}x_{170}$
Main Property : $PC_i \perp PC$□ for all $i \neq j$

Implementation(PCA): We need to normalize all variables on the same scale before finding coefficients. X_new=(X-mean(X))/SD(X)

1. The code first imports the PCA class from the Scikit-learn decomposition module.
2. Next, it initializes an instance of the PCA class with n_components=4. This means that the transformed dataset will have only 4 principal components.
3. The code then fits the PCA model to the input data, df, using the fit() method of the PCA instance. This step computes the principal components and determines their weights in the original dataset.

4. After fitting the PCA model, the code transforms the original dataset, df, into a new dataset, df_transformed, with only 4 principal components using the previously fitted PCA model. This is done using the transform() method of the PCA instance.
5. Finally, the code prints the transformed dataset, df_transformed, using the print() statement. Overall, this code performs PCA on the input dataset, df, and transforms it into a new dataset with only many principal components with decreasing importance. The transformed dataset can be used for further analysis or modeling.

Expected Results:

We have found out the loadings ($a_1,...,a_{170}$, $b_1,...,b_{170}$,....., $d_1,...,d_{170}$) for our PCs.

We selected principal components based on the eigenvalue. The standard practice in the literature is that the principal components with eigenvalues greater than unity should be selected. Though it is an arbitrary choice, we follow it, because we need to draw a line somewhere. For exposition, we draw the cumulative variance explained by principal components in the following graph.

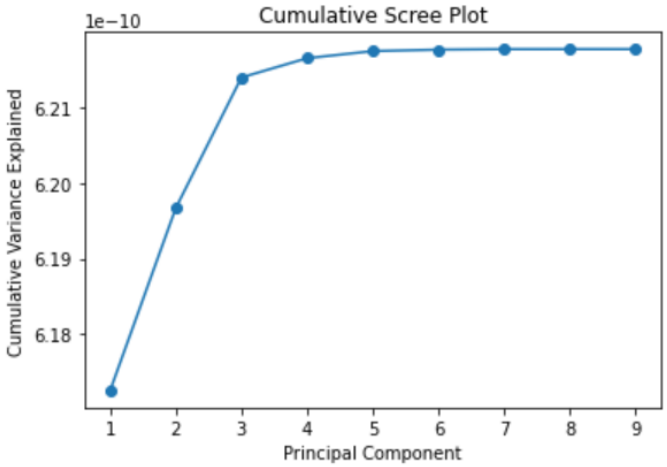


Figure 21.

Now we can define the distance (between point-1 and point-2):

$$d_{12}=\sqrt{[(PC_{11}-PC_{12})^2+(PC_{21}-PC_{22})^2+\ldots..+(PC_{41}-PC_{42})^2]}$$

Based on the minimum distance, we group points with each other. Now we can do clustering using the principal components. Following steps talk about it in detail.

3.1.3 Methodology using K-means clustering

K-means clustering is a powerful unsupervised machine learning algorithm that can be used to partition data points into meaningful clusters based on similarity or distance. The value of k is an important parameter that must be chosen carefully to obtain the best results for a given problem.

1. Choose k data points from the dataset randomly to be used as the initial centroids for the k clusters.
2. Allocate each data point to the closest centroid using the Euclidean distance or another distance measurement technique.
3. Recalculate the centroids for each cluster by averaging all data points that belong to that cluster.
4. Do repetition of steps 2 and 3 until convergence is reached, that is the centroids no longer change or a maximum number of iterations is reached.
5. The last centroids indicate the middle of each cluster, and the data points are partitioned into k clusters based on the nearest centroid.

Implementation of methodology (Clustering):

1. The code we used is performing KMeans clustering on a dataset that has been transformed using principal component analysis (PCA). The goal is to group similar data points into clusters.
2. The KMeans object is created with 5 clusters, and the fit method is called on the PCA transformed data. The labels for each data point are then obtained using the labels_ attribute of the KMeans object.
3. Finally, the cluster labels are assigned to a new column named "cluster_number" in the original dataset using the assign method of pandas DataFrame.
4. The resulting output is the array of cluster labels assigned to each data point. The length of this array should be the same as the number of rows in the original dataset, and each element represents the cluster number assigned to the corresponding data point.

3.1.4 Results

The result "Cluster Means" shows the mean values of various features for each of the four clusters that were created based on some clustering algorithm. The features include latitude and longitude of user locations, distance to the destination, whether the user is using a mobile device, whether the booking includes a package, the booking channel, number of adults and children, number of rooms, destination ID, and many other features (157 in total).

Each row in the result represents one cluster (identified by the "cluster_number" index). The values in each column represent the average value of that feature for all users in that cluster.

For example, we can see that users in cluster 0 have an average latitude of 40.20 and an average longitude of -86.00, while users in cluster 3 have an average latitude of 39.30 and an average longitude of -88.77. We can also see that users in cluster 1 are more likely to book a package than users in the other clusters, with an average value of 0.24 compared to the other clusters' values of around 0.08-0.09.

The "Cluster Counts" result shows how many users were assigned to each cluster. In this case, the largest cluster is cluster 1 with 62,124 users, followed by cluster 2 with 27,229 users, cluster 3 with 10,955 users, and cluster 0 with 6,539 users.

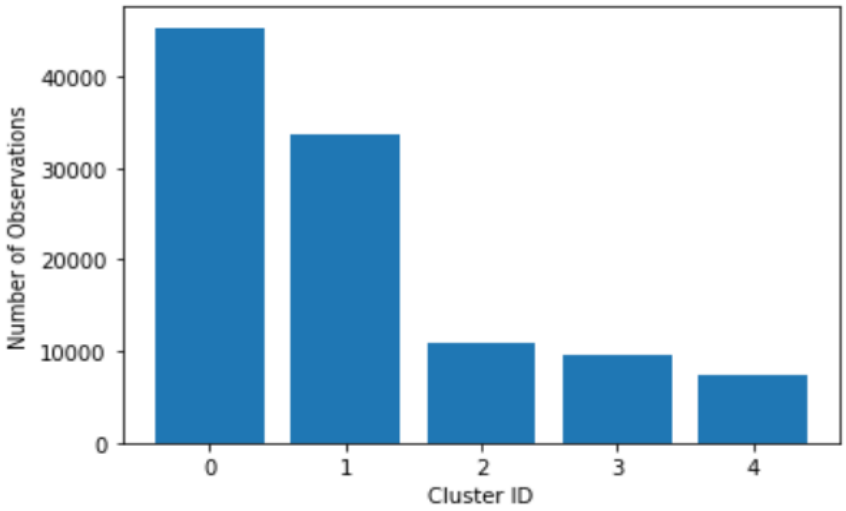


Figure 22. 5 clusters: G0 contains 52% observations, rest G1 25%, G3 11%, G4 & G5 6% each.

We have analyzed many variables at group level. We do two main things: see correlations among related variables e.g. water related travel activities, sports activities at group level. In another exercise, we see the group level means of various variables. We draw comparative inferences from it. Lots of patterns are possible and discussion of all of them is beyond the limit of this project. We therefore, briefly write some of them.

Some of the interesting underlying patterns in the data extracted by cluster analysis:

1. Most Frequent Cities in groups: Group 0: LA, G1: Stuttgart, G2: Miami, G3:SF, G4:NY
2. Booking: Group 2 booked the most hotels. Then G1,G0,G4, and G3 in decreasing order.
3. Package: Three groups #0,2,4 (70% of total) care about packages.
4. Branded Hotel: group #0,2,4 prefers branded hotels.
5. Hotel Popularity: Customers from LA and Miami prefer popular hotels.
6. Mobile: All the groups are alike in connecting from mobile.
7. Price and distance: Groups don't differ in their choice of hotel's price bands and hotel's distance from destination.
8. Tourist activities at destinations;
9. In G0,G3, and G4, if someone goes for a waterfall, she also goes for whale-watching, zoos, and wildlife. But other groups don't.
10. G0 and G4 are party people : like casinos, nightlife.
11. G4 most and G1 is least interested in sports.

3.2 What Matters the Most When One Books the Hotel?

Objective- To find means to identify the key factors or variables that have a significant impact on the decision-making process when booking a hotel.

3.2.1 Methodology

Why variable selection- If we Include irrelevant or redundant features, it can negatively impact the performance of a predictive model. We select the most important variables that can improve model accuracy and reduce errors.

Why not Linear regression?

Linear regression is a type of supervised learning algorithm that helps to model the relationship between an independent variable (also called as the predictor variable) and a dependent variable (also known as the response variable). In linear regression, our goal is to find the linear function that best fits the data.

The equation for linear regression can be written as:

$$\min_{\{\beta_0,\beta_1,\ldots,\beta_p\}} \quad \sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

This equation best fits the data but doesn't provide most variables. To do so, we use LASSO, where use of a penalty term in the loss function has the effect of shrinking the coefficients of less important variables towards zero, effectively removing them from the model. This results in a sparse model with only the most important variables included.

Our optimization problem can be written as:

$$\min_{\{\beta_0,\beta_1,\ldots,\beta_p\}} \quad \sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|$$

Here, the second term containing the | β□| the penalty will shrink all of the coefficients towards zero. Lasso shrinks the coefficient estimates towards zero. In the case of the lasso, the penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter **λ** is sufficiently large. Hence, much like best subset selection, the lasso performs variable selection.

When **λ** =0, then the lasso simply gives the least squares fit, and when **λ** becomes sufficiently large, the lasso gives the null model in which all coefficient estimates equal zero.

After using LASSO, we use a random forest classifier instead of random forest because the target variable is binary, classification algorithms are often preferred over regression algorithms because they are specifically designed to handle categorical data.
The importance score is a measure of the relevance or importance of each input feature (i.e., predictor or independent variable) in a predictive model.

The importance score is calculated by examining how much the model's performance changes when a particular feature is removed or altered. Specifically, it measures the impact of each feature on the model's output, typically expressed as a number between 0 and 1. Features that have a higher importance score are considered to be more influential in determining the model's predictions.
We used a cross-validation method to find out the error of the given model, we chose a model with lower cross-validated error given two models. Cross-validation is a popular method for estimating accuracy because it provides a more reliable estimate of the model's performance than simply using a single training and testing set.

3.2.2 Implementation

1. The code is performing Lasso regression on a dataset with the target variable 'is_booking'. The dataset gets split into training and testing sets by the train_test_split() function from scikit-learn.
2. The predictor variables in the dataset are then standardized using the StandardScaler() function. We do so because standardizing the data can improve the performance of the models by bringing all the variables to a similar scale, which prevents the variables with larger values from dominating the algorithm.
3. Next, the LassoCV model is created using the LassoCV() function from scikit-learn with cv=5, which performs a 5-fold cross-validation to select the optimal value of the regularization parameter alpha.
4. The fit() method is then called on the LassoCV model to fit it to the data.
5. The code then prints out the number of important and not-important variables in the model using the coefficients obtained from LassoCV. Specifically, it counts the number of non-zero coefficients (which correspond to important variables) and the number of zero coefficients (which correspond to not-important variables).
6. Fit the model to the training data using the fit() method.

7. Using the feature_importances_ attribute of the model in order to get the importance scores of each feature, higher importance score are considered to be more influential in determining the model's predictions

3.2.3 Results and Inferences

Since we do the variable importance selection into two parts: first is to screen out the important variable through LASSO. Using this technique, we found a total of 68 important variables and remaining not important.

The output we get shows that there are 68 important variables and 88 not-important variables. This means that Lasso regression has selected 68 variables as important predictors for the target variable *'is_booking*', and has effectively eliminated the remaining 88 variables by setting their coefficients to zero. Some of the important selected variables are: '*prop_is_branded*', '*popular_food_coffee*', '*user_location_latitude', 'user_location_longitude','is_package','srch_adults_cnt','srch_children_cnt', 'srch_rm_cnt', 'srch_destination_type_id',* etc. This demonstrates the feature selection capability of the Lasso regression.

Further, we checked how our model behaved on the prediction performance. The model containing 68 predictors from LASSO has an accuracy of 86.5% in predicting the target variable. Therefore, an accuracy of 86.5% suggests that the Random Forest Classifier model with 68 important input variables is performing well on the given dataset, and is able to make accurate predictions of the target variable with a high degree of confidence.

Going one step further, we want to know from these 68 variables, which are the most important and the next important variables. Some kind of importance ranking will help us to recommend top five or top ten variables to be focused upon. One such method is to give an importance score using a random forest classifier (RCF). The RCF gives an importance score based on the number of times a variable occurring on the node split in assembled trees. We found that the most important variable is *orig_destination_distance* that is, the input feature "orig_destination_distance" had the highest importance score among all the input features used in the model.

The four next most important variables are: *location of the hotel, star-rating, channel,* and *cnt,* which means that these four input features had the highest importance scores after the *"orig_destination_distance"* variable. These variables are in the top five that contributed significantly to the prediction performance of the model.

Using a random forest classifier importance score, we are able to rank all the important variables obtained from LASSO.

4. CONCLUSION & RECOMMENDATIONS

CONCLUSION-

1. **Main Variable:**

   Is_booking: The main variable would likely be used to determine whether or not a booking has been made.

2. **Main Factors that impact Booking pattern:**

   Distance of destination, **hotel location**, **hotel rating**, and **user history** are the most important factors determining hotel booking on Expedia.
   Understanding the main factors can influence bookings in a particular context, can help businesses and organizations to tailor their offerings and marketing strategies to better meet the needs.

3. **How does each cluster behave as per analysis?**

   Some **groups like nightlife more** and **others like sports & leisure activities**. Hotels could potentially benefit from targeted advertising aimed at specific groups.

4. **Do travelers book more individual rooms or in a package ?**

   As per analysis, **people with children book a package more** while **maximum people book individual rooms**.

RECOMMENDATIONS-

- We can focus on Distance of destination, hotel location, hotel rating, and user history variables to maximize the hotel booking strategy.
- Expedia should adopt Targeted Advertising Strategy for different customer segments.
- Efforts should be made on building different vacation packages for a variant cluster of customers.

5. REFERENCES

1. Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). “The elements of statistical learning: data mining, inference, and prediction”. *2nd ed. New York, Springer*.
2. Tibshirani, R. (1996). “Regression shrinkage and selection via the Lasso”. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288.
3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, Olivier Grisel, Mathieu Blondel & Vanderplas, J. (2011). “Scikit-learn: Machine learning in Python”. *Journal of Machine Learning Research*, 12(Oct), 2825-2830
4. Abdi, H., & Williams, L. J. (2010). “Principal component analysis”. *Wiley interdisciplinary reviews: computational statistics*, 2(4), 433-459.

6. CONTRIBUTION OF MEMBERS

CHARU JOSHI -

- Data acquisition & management
- Introduction, Project Evaluation & Conclusion
- Project Notebook, Report writing and presentation

SAKSHEE VAIDYA -

- Data acquisition & management
- Exploratory Data analysis using python in Google collab
- Data visualization & management
- Report writing and presentation

MONIKA BALODA -

- Pattern recognition methodologies.
- Theoretical justification and inferences
- Variable selection techniques
- Report writing and presentation

7. DATA/CODE AVAILABILITY

1. **Exploratory Data Analysis**

   Link for Google Collab:
   https://colab.research.google.com/drive/17qJ96d78Om9KUDmc1FVTrJDliKZa0Umh?usp=sharing

2. **Model Evaluation & Validation**

   Link for Google Collab:
   https://colab.research.google.com/drive/14HDl4TxjBMtaFzvMTdSdWVyd3kBh21mz