Project of System and Methods for Big and Unstructured Data (SMBUD Course)

Professor Marco Brambilla – Year 2023/2024

Davide Ettori – Personal Code: 10707168 – Matricola: 244696

POLITECNICO
MILANO 1863

Project introduction

The dataset I have chosen is on Hotel Booking Cancellations, in .csv form, and it presents a multifaceted problem that's intriguing to tackle using MongoDB as the database technology.
My aim is to both analyze and predict trends in hotel booking cancellations, a critical challenge in the hospitality industry. With MongoDB's flexibility and scalability, I intend to delve into the dataset's diverse attributes (ranging from guest demographics to booking details) to uncover patterns influencing cancellations. MongoDB's document-oriented nature aligns well with the dataset's varied information, enabling seamless storage and retrieval of complex, unstructured data. Leveraging MongoDB's aggregation framework, I aspire to gain insights into factors affecting cancellations, and guest behavior trends based on various attributes like room types, meal preferences, and booking patterns. MongoDB's capacity to handle large volumes of data and perform complex query makes it an ideal choice for this extensive dataset.
I am focusing on 2 main types of information, during all the process of data analysis. The first objective is statistical analysis of the guest behavior and patterns, which is very important for business reason and advertisement choices. The second objective is predictive modeling, in order to try inferring the possibility of cancelation from the raw data points. The second objective will be also part of the bonus section, where I will utilize an Explainable Neural Network (SHAP method) to derive which features are correlated to the cancelation and which one don’t carry any informative content.
I mainly used aggregations query because they allow for the examination of data from multiple perspectives, facilitating the identification of correlations, trends, and anomalies within the dataset. The final objective of my project is to extract valuable information from the dataset, which can be used from the hospitality business to optimize their marketing decisions. I am also proposing some statistics to try predicting the likelihood of a booking cancellation, as well as a machine learning model. I exploited the knowledge acquired during the course, especially about Big Data Analysis, in order to provide meaningful information for the Hotels.

Data Wrangling

I performed some data wrangling before starting to write queries, manly because I wanted to have my dataset as complete and efficient as possible and I preferred to spend some time at the beginning cleaning the raw data. All the processing has been done using the Pandas library, in Python and Jupyter Notebook. First of all I noticed that some attributes names where not very clear, so I renamed a few features, such as “market cluster”, “car parking” and “prev canc”. Then I also decided to add new columns that I tough could be useful for future analysis. I added a column “number of people” as the sum of “number of adults” and “number of children“ and the column “number of nights” as the sum of “number of week nights” and “number of weekend nights”. Afterward I tried importing the .csv dataset in the MongoDB console, but I noticed that some datatypes where wrongly interpreted. I made “date of reservation” as a Date instead of String, “price” as a Double instead of Int32 and “car parking” as a Boolean instead of Int32. Moreover, during the explorative analysis, I noticed that there where a lot of outliers in the fields “price” and “lead time” which must be dealt with, avoiding them to influence too much the average statistic, I also suspect that some of those numbers were reported wrongly. I decided to drop all that datapoints that where too much over the 0.75 percentile (3000 rows), basing on the interquartile range of the data distribution. Finally I also realized that the time period of the booking could be interesting to include separately, so I extracted 2 integer values for month and year from the “date of reservation” attribute. I did not include a day attribute because I consider it not very useful for my analysis. Before loading the dataset I also looked for missing values or null values, in order to drop or impute those attributes, but I found out that this dataset contained essentially no missing values so it was not necessary to worry about this possibility. At this point I was satisfied with my dataset, so I started querying the database. The code is in the file “preprocessing.ipynb”

Dataset Description - data taken from Kaggle website

https://www.kaggle.com/datasets/youssefaboelwafa/hotel-booking-cancellation-prediction/data

The dataset that I loaded locally on MongoDB Compass, had 22 attributes and 33.000 samples. The non-relational schema implemented in MongoDB is fundamentally simple and that's part of the reason I have chosen this technology. The main focus is on the booking information, the main entity, and there are no particularly important relationship to be modeled for the database structure. I created a database named Hotel_Booking_Cancelations_DB and inside it a single collection named Hotel_Booking_Informations. This collection will contain all the documents with information on various hotel booking. I am providing the final version, after preprocessing.

This is the attribute's structure of the Booking entity:

- **Booking_ID** (String): Unique identifier for each booking.
- **number of adults** (Int): Count of adults in the booking.
- **number of children** (Int): Count of children in the booking.
- **number of people** (Int): Total count of people (adults + children).
- **number of weekend nights** (Int): Number of nights reserved for weekends.
- **number of week nights** (Int): Number of nights reserved for weekdays.
- **number of nights** (Int): Total number of nights in the booking.
- **meal type** (String): Type of meal selected for the booking.
- **car parking** (Boolean): Indicates if car parking was booked (True/False).
- **room type** (String): Type of room booked.
- **lead time** (Int): Lead time between booking and check-in.
- **market cluster** (String): Cluster categorization for market segments.
- **repeated** (Boolean): Indicates if the booking is repeated (True/False).
- **prev canc** (Int): Count of previous cancellations by this customer.
- **not prev canc** (Int): Count of non-previous cancellations by this customer.
- **price** (Float): Price of the booking.
- **special requests** (Int): Count of special requests made.
- **date of reservation** (Date): Date of reservation.
- **month** (Int): Month of the reservation date.
- **year** (Int): Year of the reservation date.
- **status** (String): Booking status (Canceled / Not Canceled).

This non-relational schema captures various aspects of hotel reservations, encompassing booking details, guest demographics and temporal attributes, leveraging MongoDB's document-based structure to efficiently store and manage this diverse dataset.

Queries Analysis

I'm reporting 10 queries on this database, choosing the ones which are more informative for the Hotels and the ones more insightful for extracting knowledge from this dataset. The queries range in different aspects of the database, from Booking Patterns and Guest Demographics to Seasonal Analysis and Cancellation Probabilities.

1) Average number of weekend nights for each room type

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([
                                 { $match: { "number of weekend nights": { $gt: 0 } } },
                                 { $group: { _id: "$room type", avg_weekend_nights: { $avg: "$number of weekend nights" }, avg_weekdays_nights: { $avg: "$number of week nights" } } },
                                 {$sort: {"_id": 1}}
                               ])
```

Computes the average number of weekend nights booked for each room type, revealing preferences for weekend stays across different room categories. The data are then sorted in ascending order based on the average number of weekend nights. This is helpful for understanding which room are preferred for weekend and for weekdays


```
  _id: 'Room_Type 1',
  avg_weekend_nights: 1.5084843671023178,
  avg_weekdays_nights: 2.1360387031554624
}
{
  _id: 'Room_Type 2',
  avg_weekend_nights: 1.5797872340425532,
  avg_weekdays_nights: 2.351063829787234
}
{
  _id: 'Room_Type 3',
  avg_weekend_nights: 1,
  avg_weekdays_nights: 2.5
}
{
  _id: 'Room_Type 4',
  avg_weekend_nights: 1.5767670915411356,
  avg_weekdays_nights: 2.641946697566628
}
{
  _id: 'Room_Type 5',
  avg_weekend_nights: 1.5087719298245614,
  avg_weekdays_nights: 2.5350877192982457
}
{
  _id: 'Room_Type 6',
  avg_weekend_nights: 1.6101083032490975,
  avg_weekdays_nights: 2.472924187725632
}
{
  _id: 'Room_Type 7',
  avg_weekend_nights: 1.4285714285714286,
  avg_weekdays_nights: 2.9285714285714284
```

2) Bookings from market clusters with the highest price average

```
Hotel_Booking_Cancelations_DB > db.Hotel_Booking_Informations.aggregate([
                                  { $group: { _id: "$market cluster", avg_price: { $avg: "$price" }, max_price: { $max: "$price" }, min_price: { $min: "$price" } } },
                                  { $sort: { avg_price: -1 } }
                                ])
```

Tries to identify clusters (market segments) with the highest average booking prices, trying to understand revenue generation from different market segments. Then I also visualized the maximum and minimum expenditure in order to see which type of clients are more inclined to spend a lot or to save a lot. The results are ordered from highest to lowest average price.


```
{
  _id: 'Online',
  avg_price: 109.75663013950918,
  max_price: 182.34,
  min_price: 23
}
{
  _id: 'Aviation',
  avg_price: 100.704,
  max_price: 110,
  min_price: 79
}
{
  _id: 'Offline',
  avg_price: 90.4126529100529,
  max_price: 182.53,
  min_price: 26.35
}
{
  _id: 'Corporate',
  avg_price: 82.47888055972014,
  max_price: 181.79,
  min_price: 31
}
{
  _id: 'Complementary',
  avg_price: 65.99,
  max_price: 170,
  min_price: 20
}
```

3) Analysis of Repeated Bookings with Special Requests to Total Repeated Bookings Ratio

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([
                                {$match: {repeated: true}},
                                {$group: {_id: "$repeated",bookings_with_special_request: {$sum: {$cond: [{ $gt: ["$special requests", 0] }, 1, 0]}},
                                    total_repeated_bookings: { $sum: 1 }}},
                                {$project: {
                                    _id: 0,
                                    ratio_with_special_request: {$cond: [
                                        { $gt: ["$total_repeated_bookings", 0] },
                                        { $divide: ["$bookings_with_special_request", "$total_repeated_bookings"] },
                                        0]}}}])
```

The resulting ratio offers insights into the proportion of repeated bookings that include special requests compared to the total repeated bookings. Understanding this ratio aids in assessing the prevalence of special requests among repeat guests, providing valuable insights on their preferences and enhancing guest satisfaction strategies in the hospitality domain. It is very high, since special request indicate some kind of affection or trust for the hotel

4) Determining the trend of lead time for different market clusters over years:

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([{
                                 $group: {
                                   _id: { market_cluster: "$market cluster", year: "$year" }, avg_lead_time: { $avg: "$lead time" } }},
                               {$sort: { "_id.year": 1 }}
                             ])
```

This query examines the average lead time variation across different market clusters over multiple years, highlighting any trends or shifts in lead time preferences. If we looked at the full query we would see that the previous year (2017) the lead time was significantly smaller

```
{
  _id: {
    market_cluster: 'Corporate',
    year: 2018
  },
  avg_lead_time: 23.652925531914892
}
{
  _id: {
    market_cluster: 'Complementary',
    year: 2018
  },
  avg_lead_time: 17.307692307692307
}
{
  _id: {
    market_cluster: 'Offline',
    year: 2018
  },
  avg_lead_time: 109.37776766757659
}
{
  _id: {
    market_cluster: 'Online',
    year: 2018
  },
  avg_lead_time: 77.30867522044458
}
```

5) Analysis of Seasonal Special Requests Variation by Market Cluster

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([
                               {$match: { "special requests": { $gt: 0 } }},
                               {$group: {
                                   _id: { market_cluster: "$market cluster", month: { $month: "$date of reservation" } },
                                   total_special_requests: { $sum: "$special requests" }}},
                               {$sort: { "_id.market_cluster": 1, "_id.month": 1 }}])
```

This query examines the variation in seasonal special requests across different market clusters, highlighting changes in guest preferences for additional services over months. It is useful to identify repeated needs of customers and be prepared to all the possibilities that can arise


```
{
  _id: {
    market_cluster: 'Corporate',
    month: 11
  },
  total_special_requests: 76
}
{
  _id: {
    market_cluster: 'Corporate',
    month: 12
  },
  total_special_requests: 73
}
{
  _id: {
    market_cluster: 'Offline',
    month: 1
  },
  total_special_requests: 53
}
{
  _id: {
    market_cluster: 'Offline',
    month: 2
  },
  total_special_requests: 74
}
```

6) Analysis of Special Requests for Bookings with Previous Cancellations

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([
                                {$group: {_id: "$special requests",
                                    "previoulsly_canceled": { $sum: "$prev canc" },
                                    "not_previoulsly_canceled": { $sum: "$not prev canc" },}},
                                {$sort: {previoulsly_canceled: -1}}])
```

This query analyze the number of previously canceled and not previously canceled booking, dividing the data based on the number of spacial requests. As we can see, those values are descending and ascending in opposition. It means that those attributes have a strong correlation, which must be kept in mind when handling the booking informations


```
{
  _id: 0,
  previoulsly_canceled: 436,
  not_previoulsly_canceled: 2301
}
{
  _id: 1,
  previoulsly_canceled: 257,
  not_previoulsly_canceled: 1608
}
{
  _id: 2,
  previoulsly_canceled: 62,
  not_previoulsly_canceled: 643
}
{
  _id: 3,
  previoulsly_canceled: 1,
  not_previoulsly_canceled: 16
}
{
  _id: 4,
  previoulsly_canceled: 0,
  not_previoulsly_canceled: 1
}
{
  _id: 5,
  previoulsly_canceled: 0,
  not_previoulsly_canceled: 0
}
```

7) Analysis of Guest Count by Lead Time Range

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([
                                {$addFields: {
                                    lead_time_range: {
                                      $concat: [
                                        { $cond: [{ $lte: ["$lead time", 30] }, "0-30 days", "31+ days"] },
                                        " (",
                                        {$dateToString: {format: "%Y",date: "$date of reservation"}},")"]}}},
                                {$group: {
                                    _id: "$lead_time_range",
                                    total_guests: { $sum: "$number of people" }}},
                                {$sort: { total_guests: -1 }}])
```

This query categorizes the guest count based on lead time ranges (30 days or less, more than 30 days) per year, showcasing guest counts for different lead time segments across years. It is useful for understanding the general trend of hotel's booking, as well as for understanding the booking habits of customers at different time periods


```
{
  _id: '31+ days (2018)',
  total_guests: 37457
}
{
  _id: '0-30 days (2018)',
  total_guests: 15366
}
{
  _id: '31+ days (2017)',
  total_guests: 6233
}
{
  _id: '0-30 days (2017)',
  total_guests: 5041
}
{
  _id: '31+ days (2015)',
  total_guests: 2
}
```

8) Analysis of Child Count Distribution by Market Cluster

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([
                                {$group: {
                                    _id: { market_cluster: "$market cluster", children_count: "$number of children" },
                                    total_bookings: { $sum: 1 }}},
                                {$sort: { "_id.children_count": -1 }}])
```

This query examines the distribution of child counts across different market clusters, displaying the number of bookings for each child count in various market segments. I think it is valuable because children are a particular type of customers, which need special preparation. I am sorting the values by the number of children because that is the attribute I am focusing on


```
{
  _id: {
    market_cluster: 'Online',
    children_count: 3
  },
  total_bookings: 7
}
{
  _id: {
    market_cluster: 'Complementary',
    children_count: 2
  },
  total_bookings: 2
}
{
  _id: {
    market_cluster: 'Offline',
    children_count: 2
  },
  total_bookings: 10
}
{
  _id: {
    market_cluster: 'Online',
    children_count: 2
  },
  total_bookings: 547
}
```

9) Probability of canceled bookings for each meal type

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([
                                 {$group: {
                                     _id: "$meal type",
                                     total_bookings: { $sum: 1 },
                                     canceled_bookings: {
                                       $sum: { $cond: [{ $eq: ["$status", "Canceled"] }, 1, 0] }}}},
                                 {$project: {
                                     _id: 1,
                                     probability_cancelation: { $divide: ["$canceled_bookings", "$total_bookings"] }}},
                                 {$sort: {probability_cancelation: -1}}
                               ])
```

Calculates the probability that a booking will be canceled for each meal type. It offers insights into cancellation trends across customers which, for health, religion or culture, might have particular meal preference. In this case, we don't see any particular correlation

```
{
  _id: 'Meal Plan 2',
  probability_cancelation: 0.3685275080906149
}
{
  _id: 'Not Selected',
  probability_cancelation: 0.3312943962115233
}
{
  _id: 'Meal Plan 1',
  probability_cancelation: 0.3037016917584976
}
```

10) Analyzing the cancellation rate for bookings with and without car parking:

```
Hotel_Booking_Cancelations_DB> db.Hotel_Booking_Informations.aggregate([
                                {$group: {
                                    _id: "$car parking",
                                    total_bookings: { $sum: 1 },
                                    canceled_bookings: { $sum: { $cond: [{ $eq: ["$status", "Canceled"] }, 1, 0] } }}},
                                {$project: {
                                    car_parking: "$_id",
                                    cancellation_rate: { $divide: ["$canceled_bookings", "$total_bookings"] }}}
                              ])
```

This query computes the cancellation rates for bookings with and without car parking, allowing an assessment of cancellation tendencies based on this amenity. As expected, the fact of having booked a car parking indicate an lower probability of canceling the booking

```
{
  _id: false,
  car_parking: false,
  cancellation_rate: 0.31929911460590676
}
{
  _id: true,
  car_parking: true,
  cancellation_rate: 0.10198019801980197
}
```

Extra Work

I decided to train a neural network for predicting if a booking will be canceled. After the training process I will extract the features that contributes the most to the prediction and thanks to that I will be able to gain knowledge on the most informative attributes of the dataset, which are convenient to analyze. I used the SHAP method for explainability and Tensorflow for the neural network.
The utilization of SHAP (SHapley Additive exPlanations) in tandem with a trained neural network for hotel booking cancellation prediction unveils a compelling approach in the realm of data analysis. This process initiates with the training of a neural network model, crucially engineered to predict hotel booking cancellations based on a multitude of features within the dataset. Once trained, the SHAP method serves as a pivotal tool, dissecting the model's predictions and attributing importance to each feature's contribution in making cancellation predictions. This interpretability aspect is the cornerstone of effective data analysis, offering insights into which features hold significant meaning in determining the likelihood of a booking cancellation. By discerning and extracting these pivotal features, we not only gain a deeper understanding of the factors influencing cancellations but also identify high-value attributes within the dataset.
For instance, if the analysis reveals that longer lead times or specific room types significantly contribute to higher cancellation probabilities, hotels can devise tailored marketing strategies, incentivizing earlier bookings or optimizing room offerings to align with guest preferences. Moreover, armed with insights from high-impact features hotels can craft personalized guest experiences, promoting loyalty and satisfaction. Ultimately, the integration of these data-driven insights into operational frameworks empowers hotels to proactively address potential cancellations, elevate guest experiences, and optimize business performance, gaining a competitive edge in the dynamic hospitality business.
All the technical details, a clear description of the process, the method and the final results are provided in the python notebook named “SHAP Analysis.ipynb”.

The main result, to summarize, is that “special requests” and “lead time” are the most informative attributes while “price” and “market cluster” are less significant than expected. Moreover, I have found that 15 attributes are almost useless for prediction, which is surprising. Those could be dropped if our only interest is predictive modeling, but they could be very useful for a general analysis of the customer staying in the hotels.

Concluding, I really enjoyed this project since it offered me the opportunity to explore the complete data pipeline, from start to finish. It was really interesting, because I usually take for granted to start with a perfect and clear database, which is almost never the real case. Now I am definitely more conscious of the methods and objectives of Big Data Analysis, since I have seen how to apply the theoretical knowledge of the course to a real business database.