Data Mining

Project 1 – Group 4

Joshua Ball R11330455 | Dipendra Yadav R11538518

GitHub link: <https://github.com/dipenpratap/Data-Mining/tree/master/Project%201>

# **Project Report (Model Deployment)**

## **Problem Objective**

Traveling is expensive overall, but one of the biggest costs associated with a trip is often the lodging.

Studies have suggested that the best time to buy a plane ticket for lower-price options is 70 days in advance.

But does the same reasoning apply to hotel rooms?

### **Can this problem be solved using Data Science?**

We are using data from https://www.kaggle.com/ekretsch/hotel-booking-dataset to develop a clustering algorithm

that can find out the best time of the year to book a hotel room in order to get best prices in the market.

## **Setup Phase:**

* 1. In order to setup the data for the modeling phase, we must first partition the data in a balanced manner. In order to ensure the data is balanced we must perform cross-validation. To do this, Joshua utilized the sklearn library which provides a function ‘train\_test\_split’. We pass our data set into this function, setting our test partition size to 25% of the data set and the random state to 7.
  2. Validating the partitions in a crucial step in order to ensure we do not generate models in the modeling phase that have some inherent bias due to an unbalanced data set.
  3. First, we validated the data partitions sizes, getting 56374 and 18792 respectively. Calculating: we can see that the data partitions are indeed balanced size wise.
  4. Next, we checked the mean of various values which are important to our data modeling, such as variables ‘adr’ and ‘rate\_per\_stay’.

|  |  |  |
| --- | --- | --- |
|  | Data Type | Mean |
| 0 | Train\_rate\_per\_stay | 345.238976 |
| 1 | Test\_rate\_per\_stay | 347.688292 |
| 2 | Train\_adr | 99.926590 |
| 3 | Test\_adr | 100.170994 |

* 1. As we can see from the table above, all the variables in question contain very similar means with low deviation from their respective counterparts. Statically, everything about the partitions looks good thus far.
  2. Finally, Dipendra also visualized the data split for further cross validation. For both the test and train partitions, we calculated the number of guests staying in each hotel per month and graphed this. From the graphs we can further confirm the balance of the data partitions.

## **Modeling Phase:**

1. The first model we attempted was done by Joshua. He tried to utilize K-means clustering provided by the sklearn library.
2. The first clustering generated utilized the variables ‘rate\_per\_stay’ and ‘arrival\_date\_month’. From the model descriptions, Joshua found a very similar set of values with respect to cluster 1 of the train set and cluster 1 of the test set, as well as with their respective 2nd clusters. In order to validate the models however, Joshua ran the T-test on each of the clusters to compare them to their test or train counterpart. In all, our P values are about 0.11 and our Z values below the threshold of ±3. Additionally, Joshua ran clustering on the variable ‘adr’ to compare its performance against ‘rate\_per\_stay’, and we found that it gave us a P value outside of the optimal range.
3. The second model was created by Dipendra. He created a model with Linear Regression that predicts the prices of hotels. Statsmodels library has been utilized to train the model.
4. After brainstorming, Dipendra made a selection of possible predictors that included lead\_time, arrival\_date\_year, arrival\_date\_month, arrival\_date\_week\_number, stays\_in\_weekend\_nights, and days\_in\_waiting\_list as predictors that would contribute to the regression. The target variable is average\_daily\_rates.
5. The result of OLS regression is shown below. As we can see from the figure, predictors lead\_time, arrival\_date\_year, arrival\_date\_month, arrival\_date\_week\_number, stays\_in\_weekend\_nights, and days\_in\_waiting\_list are in the coef columns. P > |t| is between -0.05 and 0.05 for all attributes in the training set referring that all of these predictors belong in the model. Whereas, arriva\_date\_week\_number does not belong in the model since it does not fall in the threshold in the test set. Hence, we had to fine tune the model by removing arrival\_date\_week\_number variable from the predictors that results in decrement in the model complexity.

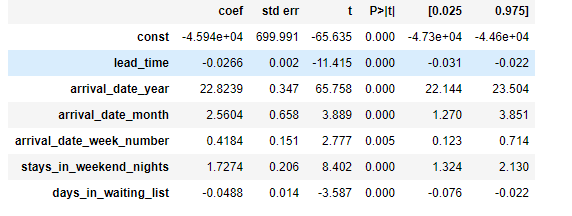


Figure 1: Train\_data OLS Regression Result

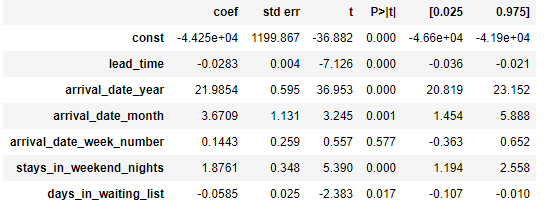


Figure 2: Test\_data OLS Regression Result

1. After fine-tuning the model algorithm, the following change was obtained in the result. There was a relevant change in the P-value. All predictors other than days\_in\_waiting\_list have a P-value of 0.000. Days\_in\_waiting\_list in the test set has a P-value of 0.018 which is perfectly acceptable.

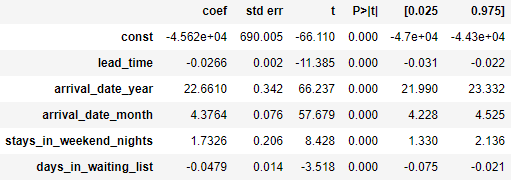


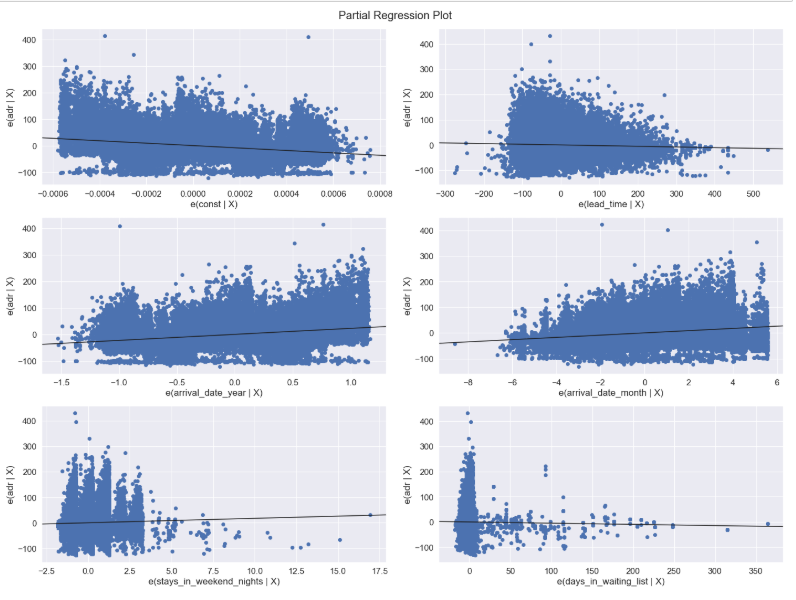
Figure 3: Train\_data OLS Regression Result (After fine-tune)



Figure 4:Test\_data OLS Regression Result (After fine-tune)

## **Evaluation Phase:**

1. For K-means clustering, Joshua found that the algorithm ran on the variables of ‘rate\_per\_stay’ and ‘arrival\_date\_month’ performed the best. This model helped us determine that the months of the summer contain the highest average rates of stay for guests in both types of hotels, exactly our objective with this data set.
2. Predictors lead\_time, arrival\_date\_year, arrival\_date\_month, stays\_in\_weekend\_nights, and days\_in\_waiting\_list are in the coef columns for this model. P >|t| is between -0.05 and 0.05 for all the variables in the training set as well as test set. This refers that all the predictors belong to the model.
3. Since all the predictors have significant P-value, our training model is:
4. The graphs below show how the model fits with all the predictor variables.



1. Example Calculation: Constraints:
   * + - 1. Lead time = 2 days
         2. Arrival date year = 2020
         3. Arrival date month = 3 (march)
         4. Stays in weekend nights = 0 (No)
         5. Days in waiting list = 5 days

Calculation: -45620 -0.0266 \* 2 + 22.6610 \* 2020 + 4.3764 \* 3 + 1.7326 \* 0 – 0.0479 \* 5 = $168.0565/day