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:

a. 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.

b. 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.

c. First, we validated the data partitions sizes, getting 56374 and 18792 respectively. Calculating: $\frac{18792}{75166} = 0.2500$ we can see that the data partitions are indeed balanced size wise.

d. 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

e. 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.

f. 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:

a. The first model we attempted was done by Joshua. He tried to utilize K-means clustering provided by the sklearn library.

b. 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.

c. 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.

d. 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.

e. 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.

	coef	std err	t	P>|t|	[0.025	0.975]
const	-4.594e+04	699.991	-65.635	0.000	-4.73e+04	-4.46e+04
lead_time	-0.0266	0.002	-11.415	0.000	-0.031	-0.022
arrival_date_year	22.8239	0.347	65.758	0.000	22.144	23.504
arrival_date_month	2.5604	0.658	3.889	0.000	1.270	3.851
arrival_date_week_number	0.4184	0.151	2.777	0.005	0.123	0.714
stays_in_weekend_nights	1.7274	0.206	8.402	0.000	1.324	2.130
days_in_waiting_list	-0.0488	0.014	-3.587	0.000	-0.076	-0.022

Figure 1: Train_data OLS Regression Result

	coef	std err	t	P>|t|	[0.025	0.975]
const	-4.425e+04	1199.867	-36.882	0.000	-4.66e+04	-4.19e+04
lead_time	-0.0283	0.004	-7.126	0.000	-0.036	-0.021
arrival_date_year	21.9854	0.595	36.953	0.000	20.819	23.152
arrival_date_month	3.6709	1.131	3.245	0.001	1.454	5.888
arrival_date_week_number	0.1443	0.259	0.557	0.577	-0.363	0.652
stays_in_weekend_nights	1.8761	0.348	5.390	0.000	1.194	2.558
days_in_waiting_list	-0.0585	0.025	-2.383	0.017	-0.107	-0.010

Figure 2: Test_data OLS Regression Result

f. 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.

	coef	std err	t	P>|t|	[0.025	0.975]
const	-4.562e+04	690.005	-66.110	0.000	-4.7e+04	-4.43e+04
lead_time	-0.0266	0.002	-11.385	0.000	-0.031	-0.022
arrival_date_year	22.6610	0.342	66.237	0.000	21.990	23.332
arrival_date_month	4.3764	0.076	57.679	0.000	4.228	4.525
stays_in_weekend_nights	1.7326	0.206	8.428	0.000	1.330	2.136
days_in_waiting_list	-0.0479	0.014	-3.518	0.000	-0.075	-0.021

Figure 3: Train_data OLS Regression Result (After fine-tune)

	coef	std err	t	P>|t|	[0.025	0.975]
const	-4.414e+04	1181.921	-37.344	0.000	-4.65e+04	-4.18e+04
lead_time	-0.0283	0.004	-7.124	0.000	-0.036	-0.021
arrival_date_year	21.9282	0.586	37.418	0.000	20.780	23.077
arrival_date_month	4.2972	0.131	32.900	0.000	4.041	4.553
stays_in_weekend_nights	1.8762	0.348	5.390	0.000	1.194	2.558
days_in_waiting_list	-0.0580	0.025	-2.366	0.018	-0.106	-0.010

Figure 4:Test_data OLS Regression Result (After fine-tune)

Evaluation Phase:

a. 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.

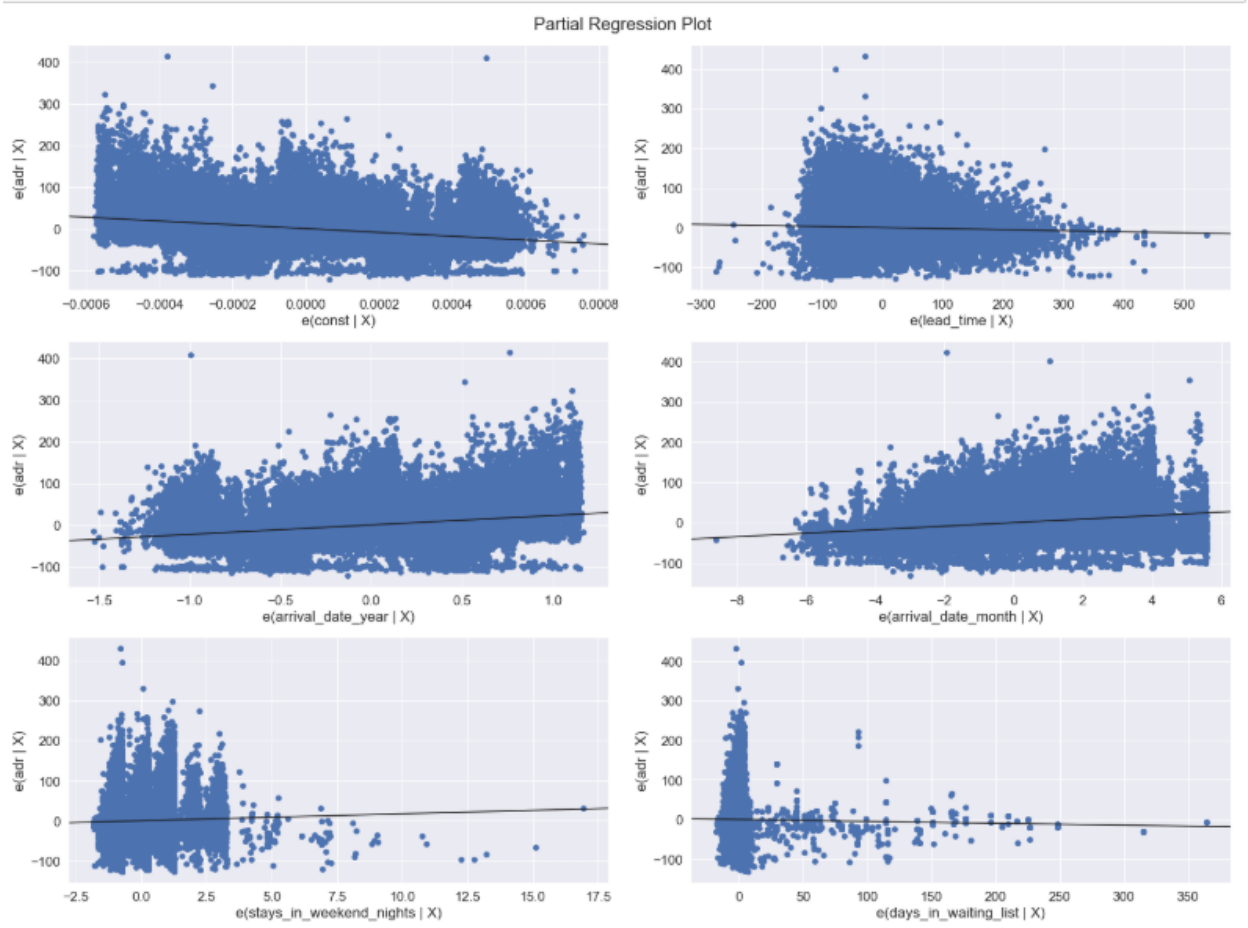
b. Predictors lead_time, arrival_date_year, arrival_date_month, stays_in_weekend_nights, and days_in_waiting_list are in the coef columns for this model. P >|t| is between -0.05 and 0.05 for all the variables in the training set as well as test set. This refers that all the predictors belong to the model.

c. Since all the predictors have significant P-value, our training model is:

$$\textit{Average Daily Rate} = -45620 - 0.0266 * \textit{lead time} + 22.6610 * \textit{arrival date year}$$

$$+4.3764 * \textit{arrival date month} + 1.7326 * \textit{stays in weekend nights} - 0.0479 * \textit{days in waitinglist}$$

d. The graphs below show how the model fits with all the predictor variables.



e. Example Calculation: Constraints:

a. Lead time = 2 days
b. Arrival date year = 2020
c. Arrival date month = 3 (march)
d. Stays in weekend nights = 0 (No)
e. 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