



Big Data Project GirlGenerations

Issue - How to define the “System Loading”

We thought **system loading** is associated with “**The amount of requests in a period of time**”.

- So, we could find how many requests the users made,
- Assume each data represent one ticket, and after every user order the ticket, would be added the data in system dataset.
- We can define **system loading** with the formula below :
 - ◆ $\Delta \text{ amount of tickets addition} / \Delta \text{ hour}$
 - ◆ $\Delta \text{ amount of people addition} / \Delta \text{ hour}$
 - ◆ Because it is difficult to observe the variance in minute, we calculated in hour.



Case 1

Δ amount of tickets addition / Δ hr

Δ hr

Data Preprocessing & Feature Engineering

Data Selection:

- "IDENTITY", "CREATE_DATE", "PRICE", "T_STANDARD_TICKET_TYPE_NAME", "SEAT_REGION_NAME", "SEX"

Feature conversion:

- Convert "CREATE_DATE" to comparable object like panda's DatetimeIndex
- Convert "T_STANDARD_TICKET_TYPE_NAME" to category type

Feature Splition :

Split "SEAT_REGION_NAME" to FloorNo to represent the seat's floor (ex floorB1, Floor2, Floor3)

Feature Combination: **Assume the total tickets is the same as sample size (n=7069)**

Compute the existing tickets that each kind of FloorNo

Compute the existing tickets that each kind of SEAT_REGION_NAME



Read
Feature
Selection

	IDENTITY	CREATE_DATE	PRICE	T_STANDARD_TICKET_TYPE_NAME	SEAT_REGION_NAME	SEX
0	K12197	2010/10/14 p.m. 11:25:58.	1800	non-member	Floor3Sectionyellow3I	male
1	K12197	2010/10/14 p.m. 11:25:58.	1800	non-member	Floor3Sectionyellow3I	male
2	L22289	2010/9/18 p.m. 11:26:37.	1800	member	Floor3Sectionyellow3C	male
3	L22289	2010/9/18 p.m. 11:26:37.	1800	member	Floor3Sectionyellow3C	male
4	C12101	2010/9/18 p.m. 11:28:56.	1800	member	Floor3Sectionyellow3C	male

Convert TimeObject

Split to FloorNo

	IDENTITY	CREATE_DATE	PRICE	T_STANDARD_TICKET_TYPE_NAME	SEAT_REGION_NAME	SEX
0	K12197	2010-10-14 23:25:58	1800	non-member	Floor3Sectionyellow3I	male
1	K12197	2010-10-14 23:25:58	1800	non-member	Floor3Sectionyellow3I	male
2	L22289	2010-09-18 23:26:37	1800	member	Floor3Sectionyellow3C	male
3	L22289	2010-09-18 23:26:37	1800	member	Floor3Sectionyellow3C	male
4	C12101	2010-09-18 23:28:56	1800	member	Floor3Sectionyellow3C	male

	IDENTITY	CREATE_DATE	PRICE	T_STANDARD_TICKET_TYPE_NAME	SEAT_REGION_NAME	SEX	FloorNo
6430	E12347	2010-09-18 13:18:55	3500	member	Floor2Sectionyellow2B	male	Floor2
6652	C12086	2010-09-18 13:22:33	4000	member	FloorB1FloorSectionA	male	FloorB1
6651	C12086	2010-09-18 13:22:33	4000	member	FloorB1FloorSectionA	male	FloorB1
6650	C12086	2010-09-18 13:22:33	4000	member	FloorB1FloorSectionA	male	FloorB1
6653	C12086	2010-09-18 13:22:33	4000	member	FloorB1FloorSectionA	male	FloorB1

Compute existing tickets
on two situations

Data
Analysis

NAME	SEAT_REGION_NAME	SEX	FloorNo	SEAT_REGION_NAMENow_TicketNumber	SEAT_REGION_NAMETotal_TicketNumber	FloorNoNow_TicketNumber	FloorNoTotal_TicketNumber
member	Floor2Sectionyellow2B	male	Floor2	198	198	2580	2580
member	FloorB1FloorSectionA	male	FloorB1	509	509	1668	1668
member	FloorB1FloorSectionA	male	FloorB1	508	509	1667	1668
member	FloorB1FloorSectionA	male	FloorB1	507	509	1666	1668
member	FloorB1FloorSectionA	male	FloorB1	506	509	1665	1668

Analyze Process – 3 Phase

We found the tickets were sell in different period of time, member could buy the ticket previously. So when we analyzed the system loading divide in member and non-member two groups.

There are three phases to discuss **System Loading Performance** :

- Member / non-member average tickets addition
- Average tickets addition of The matrix of 「 (member, non-member) 」 and 「 FloorNo(FloorB1,Floor2,Floor3) 」
- Average tickets addition of The matrix of 「 (member, non-member) 」 and 「 SEAT_REIGON_NAME 」

Phase 1 – Discuss member & non-member

- All Tickets = 7069 tickets
 - Member System Loading: 551.75 (Δ amount of ticket addition / Δ hr)
 - (member | All Tickets) = 6307 tickets
 - Timedelta = 11.43 hr
 - $6307 / 11.43 = 551.75$
 - Member System Loading = 66.66 (Δ amount of ticket addition / Δ hr)
 - (member | All Tickets) = 762 tickets
 - Timedelta = 11.34 hr
 - $762 / 11.34 = 67.18$
- Represent System can afford “551 tickets additions” per hour
- Member System Loading > non-member System Loading

Phase 2 – Discuss matrix of NMe& FloorNo

- **member**

- FloorB1 System Loading = $1460 / 11.37 = 128.4$
- Floor2 System Loading = $2311 / 9.96 = 231.94$
- Floor3 System Loading = $2536 / 11.24 = 225.59$

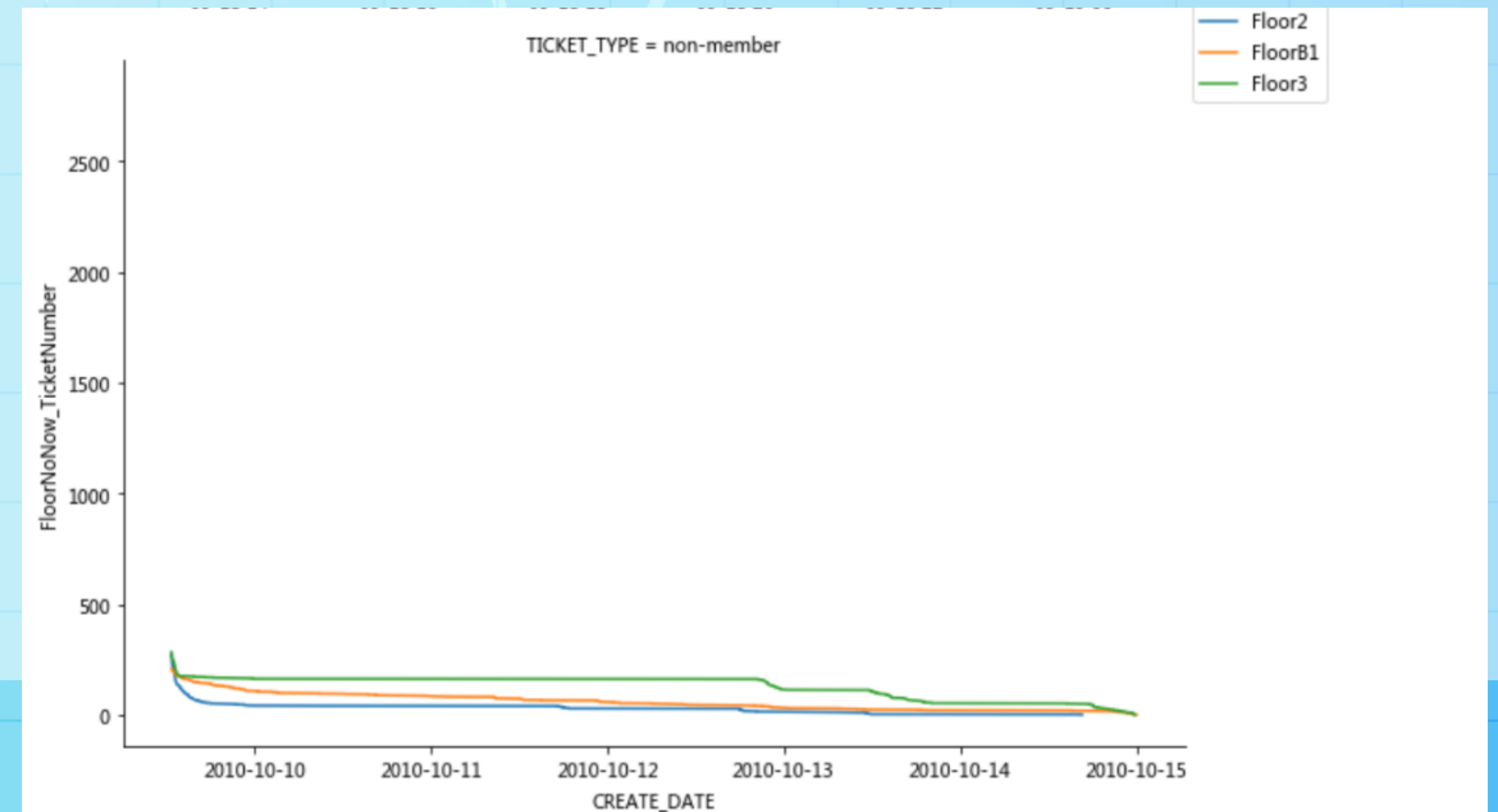
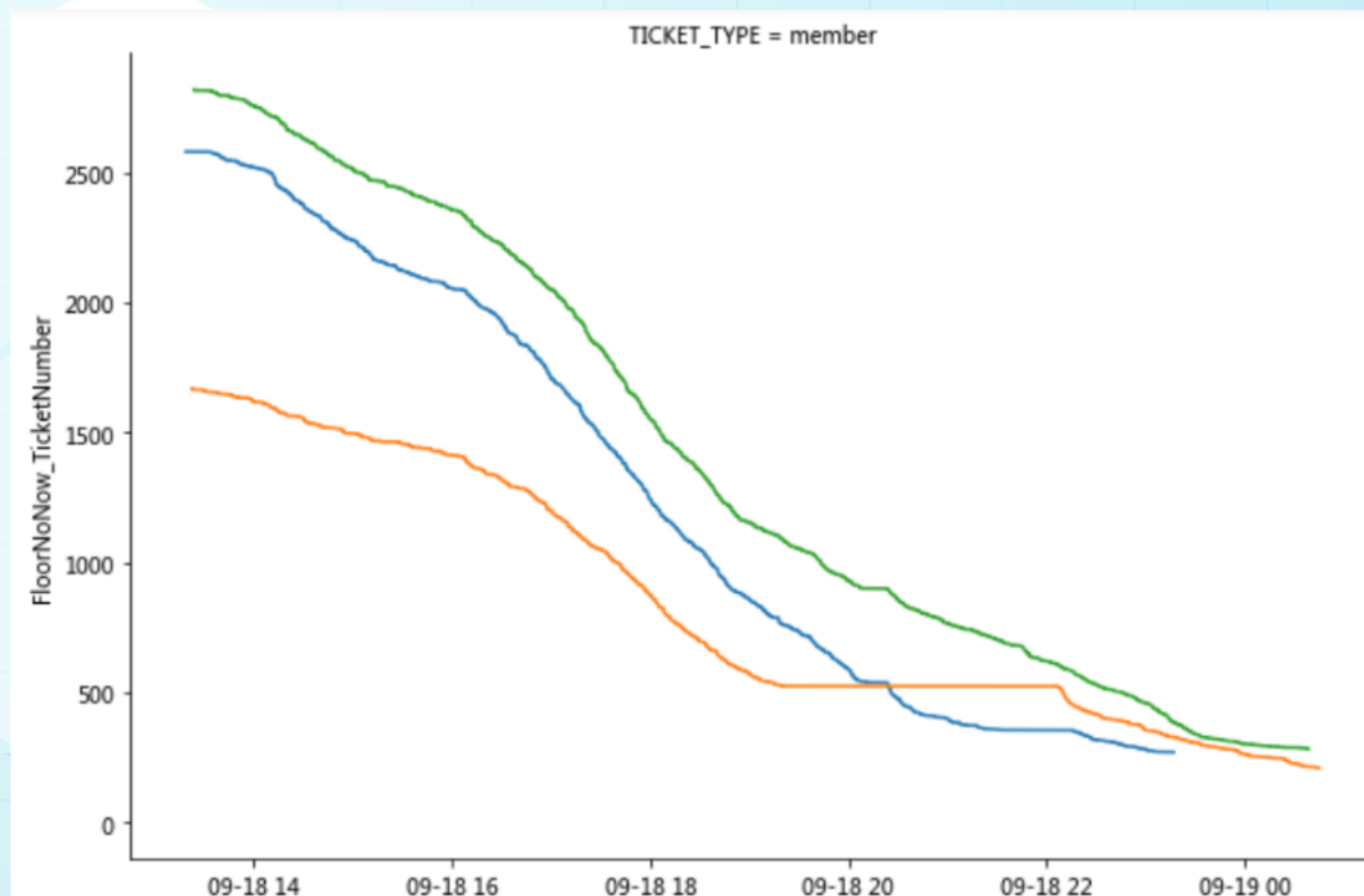
- **non-member**

- FloorB1 System Loading = $209 / 11.29 = 18.51$
- Floor2 System Loading = $270 / 3.94 = 68.56$
- Floor3 System Loading = $283 / 11.08 = 25.55$

- System Loading's Rank of each floor

- 1st : Floor2 (average price of this floor is 3598)
- 2nd : Floor3 (average price of this floor is 1854)
- 3rd : FloorB1 (average price of this floor is 4000)

- Why Floor2 > Floor3 : we guess this is because Floor2 amount of Tickets < Floor3's



Ps. NMe : Non-member, Member | FloorNo : (FloorB1,Floor2,Floor3)

Phase 3 – Discuss matrix of NMe & SEAT_REGION_NAME

This related data output is saved in Result.xlsx.

According to right data we analyze, we can found some characteristics in the data :

Member :

- The Floor3Sectionyellow3G, 3D,3C,3H is the four highest area that system loading perform very well, we can guess because this area's price (800, or 1800) is the cheapest of all tickets. ▶
- After four highest the area, following area is FloorB1's all sections => maybe it mean there always some people who want to experience the "Rock Zone."

Member and Non-member:

- According to the System loading, we find out that people tend to book the "center seat" of the concert, maybe because it has more wide view

Member

		Ticket_SaleNum (Tickets)	Sale_Period(HR)	Loading_Efficiency (Tickets/HR)
FloorNo	Seat_Type			
Floor3	Floor3Sectionyellow3G	454	8.49	53.44
	Floor3Sectionyellow3D	445	8.67	51.34
	Floor3Sectionyellow3C	475	9.99	47.55
	Floor3Sectionyellow3H	440	9.55	46.05
FloorB1	FloorB1FloorSectionA	422	9.74	43.32
	FloorB1FloorSectionC	373	9.95	37.48
	FloorB1FloorSectionD	337	11.22	30.04
	FloorB1FloorSectionB	328	11.16	29.38
Floor2	Floor2Sectionred2E	242	8.95	27.04

After Sorting

Non-member

		Ticket_SaleNum (Tickets)	Sale_Period(HR)	Loading_Efficiency (Tickets/HR)
FloorNo	Seat_Type			
Floor3	Floor3Sectionyellow3D	14	0.41	34.05
Floor2	Floor2Sectionyellow2A	13	0.78	16.58
	Floor2Sectionyellow2D	14	1.04	13.45
Floor3	Floor3Sectionyellow3F	3	0.29	10.31
FloorB1	FloorB1FloorSectionA	88	11.25	7.82
	FloorB1FloorSectionB	76	10.55	7.20
Floor2	Floor2Sectionyellow2E	16	2.35	6.80

Result-1

System can afford “551 tickets additions” per hour in Member

The maximum of affordance is 551

Member System Loading > non-member System Loading

- **System Loading's Rank of each floor :**

- Floor2 > Floor3 > FloorB1

System Loading about each area:

Floor3YellowSection{CenterArea} is the highest, maybe because the price is cheap

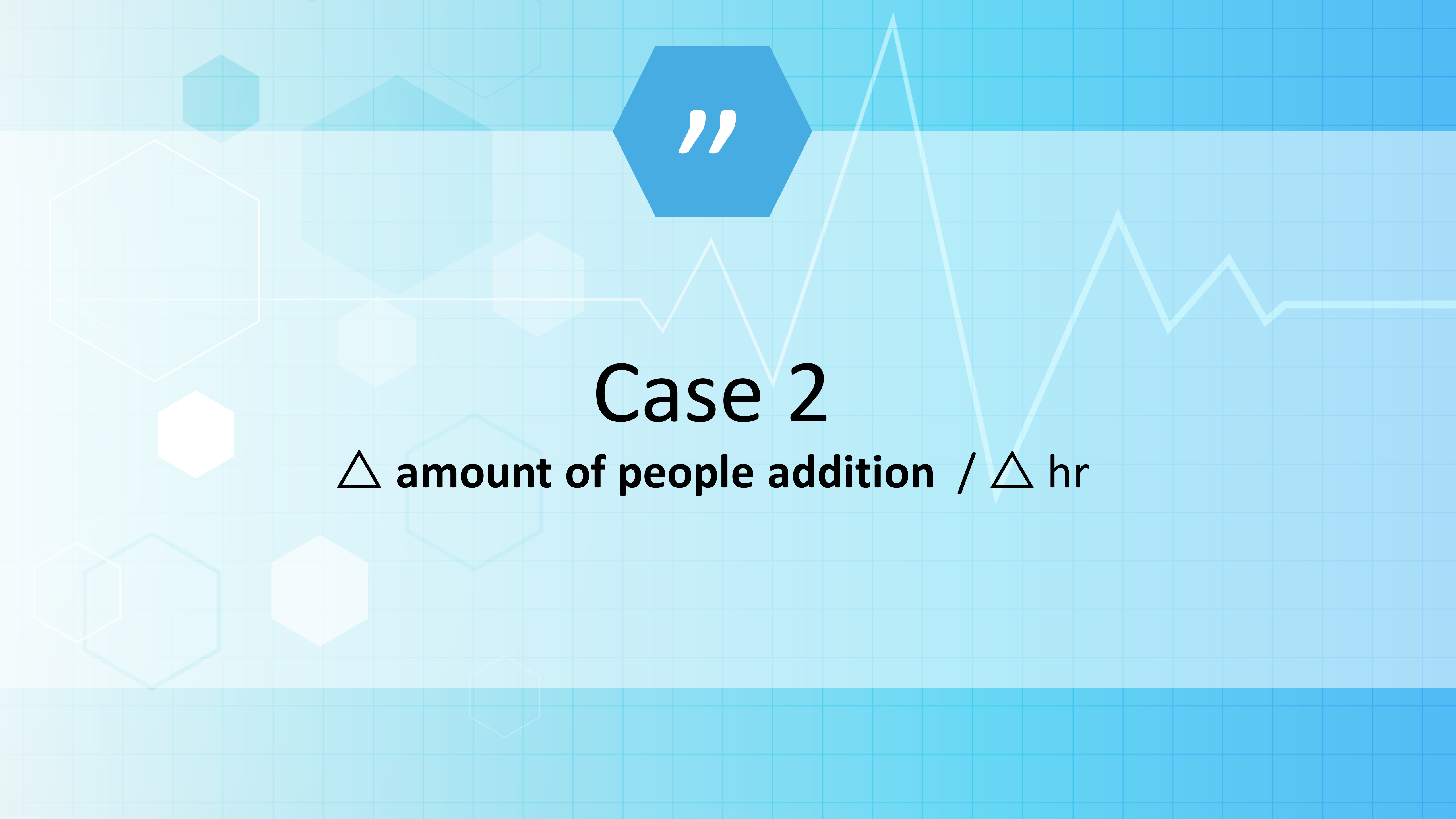
“Rock Zone” is always popular

No matter member or non-member, every body always like cheap ticket



Case 2

Δ amount of people addition / Δ hr



Data Preprocessing & Feature Engineering

Data Selection:

- "IDENTITY", "CREATE_DATE", "T_STANDARD_TICKET_TYPE_NAME"
- conversion:
- convert "CREATE_DATE" to "newDate" & "newTime"
- "newDate" & "newTime" convert to String

Feature Split :

- Split "T_STANDARD_TICKET_TYPE_NAME" to "member" & "non-member"
- Split "CREATE_DATE" to represent the different date(ex 0918, 0918, 1009,1010)

Feature Combination:

Assume the people is the same as sample size (member = 2421 , non-member = 438)

- Compute the at member purchase time & non-member purchase time
- Compute the existing buy request that each kind of different date

Analyze Process

We found every one can order four ticket in one time, but not limit the order time. So there are some people order the ticket in both period of time for member and non-member. In this case, we just wanted to discuss how many people had requests in per hour.

Because the interval of date sold ticket, we split the data in date. We would analyzed with steps below :

- ✓ Split the data in date, there were 2010-09-18 , 2010-09-19 , 2010-10-09~ 2010-10-14.
- ✓ And delete the repeated "IDENTITY" data , and divided into member and non-member.
- ✓ Calculate the total requests per day and average requests per hour.

Result-2

	Member	Non-member
Date	09/18 , 09/19	10/09 , 10/10 , 10/11 10/12 , 10/13 , 10/14
number of people request	2421 people	438 people
Purchase time	12 hr	69 hr
Request per hour	201.75 requests	6.35 requests

Purchase time : If there hour has not any request, it would not calculate in purchase time.