Commercial Online Game Data Analysis Competition

EC 4213 / ET5402 / ET5303
: Machine Learning and Deep Learning

Instructor: Jonghyun Choi

Competition Organizer

School of Integrated Technology

Cognition and Intelligence Lab

Cheong-mok Bae

Index

- Introduction
- Problem: details
- Evaluation Metric

Introduction

Design for Online Game Churn Prediction Model
for considering residual value using the Commercial Online Game Data

Lineage

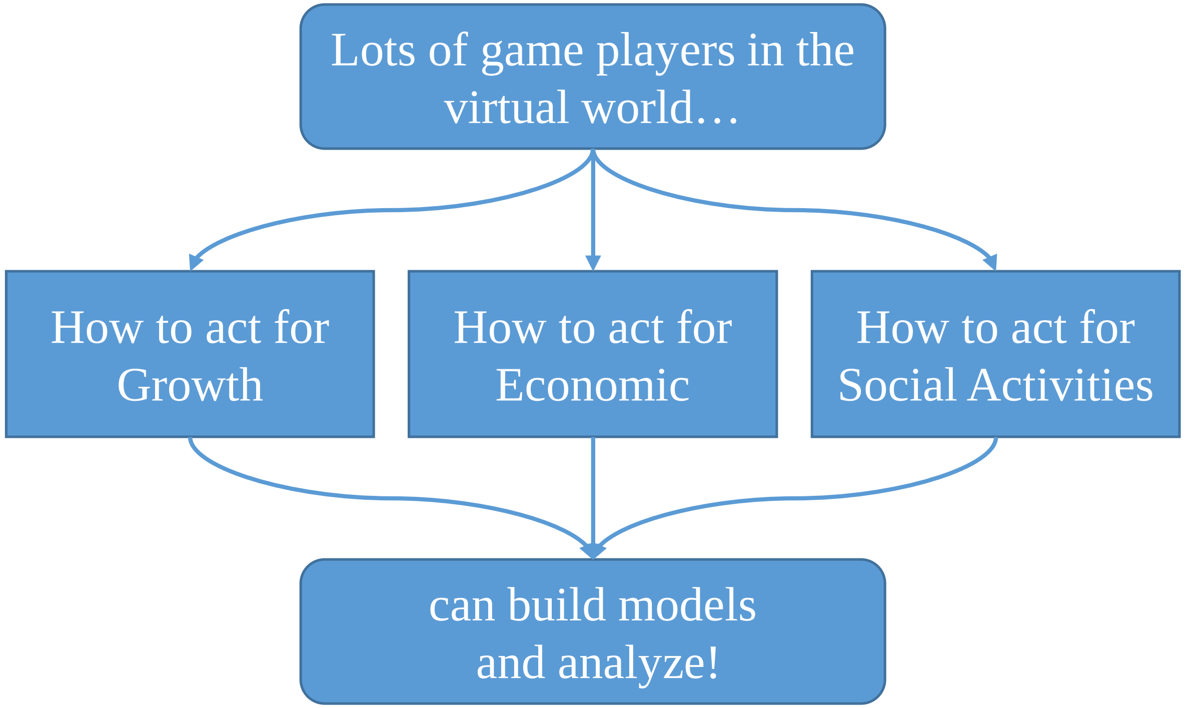
- MMORPG(Massively Multiplayer Online Role-Playing Game)
- Serviced by NCSoft from September 1st in 1998
- Achieved 3.2 trillion KRW for Cumulative Sale in 2016
- Played by 20 million users world-widely
- https://lineage.plaync.com

Lineage (cont’d)

- Can play variety activities based on degrees of freedom
  - ✓ Promoting and economic activities
  - ✓ Social activities
  - ✓ Other variety experience

Attraction of Game Data

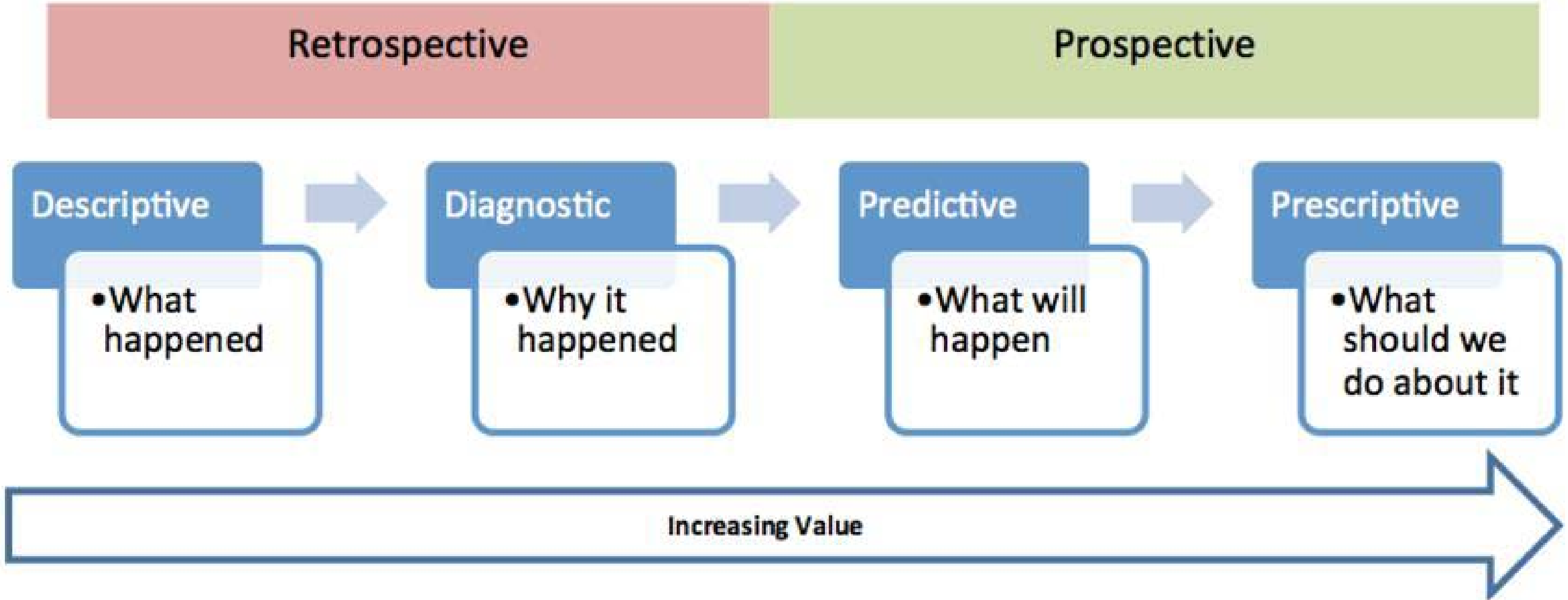
- Record a wide range of activities
  - ✓ Who/When/Where/What/How ➔ everything
  - ✓ Very high-quality data that are hard to access in real-world!



Date	Actor	Action	L...	Location	Entity	Target
19-05-27 00:00:22.157	[1] 데포로쥬,...	1003: 접속	본...	말하는 섬 마을(15850), 3...	From서버:0, 현재:1141/182, Exp:486831099, 인벤A:270049, ...	신규
19-05-27 00:00:22.157	[1] 데포로쥬,...	1005: 맵입장	0	말하는 섬 마을(15850), 3...	남음(초):0, 0	
19-05-27 00:01:09.782	[1] 데포로쥬,...	1005: 맵입장	0	말하는 섬(9), 32671/33249	남음(초):0, 0	
19-05-27 00:01:09.782	[1] 데포로쥬,...	1006: 맵퇴장	0	말하는 섬 마을(15850), 3...	남음(초):0, 0	말하는 섬(9), 3
19-05-27 00:01:35.595	[1] 데포로쥬,...	1202: NPC죽임	0	말하는 섬(9), 32647/33219	소환:오크(0), 0	다이어울프(144
19-05-27 00:01:35.595	[1] 데포로쥬,...	1017: 경험치 획득	기...	말하는 섬(9), 32647/33219	4874, Exp:486835973, 아인감소:2437, 총아인:1157998, 아인%...	다이어울프(144
19-05-27 00:01:40.970	[1] 데포로쥬,...	1202: NPC죽임	0	말하는 섬(9), 32649/33219	소환:오크(0), 0	흑기사(14459),
19-05-27 00:01:40.970	[1] 데포로쥬,...	1017: 경험치 획득	기...	말하는 섬(9), 32649/33219	6052, Exp:486842025, 아인감소:3026, 총아인:1154972, 아인%...	흑기사(14459),
19-05-27 00:01:41.235	[1] 데포로쥬,...	1012: 게임머니 증가	ge...	말하는 섬(9), 32649/33219	A:50, TotA:270099	0, 오크(0)
19-05-27 00:01:47.876	[1] 데포로쥬,...	1202: NPC죽임	0	말하는 섬(9), 32646/33212	소환:오크(0), 0	흑기사(14424),
19-05-27 00:01:47.876	[1] 데포로쥬,...	1017: 경험치 획득	기...	말하는 섬(9), 32646/33212	6052, Exp:486848077, 아인감소:3026, 총아인:1151946, 아인%...	흑기사(14424),
19-05-27 00:01:53.876	[1] 데포로쥬,...	1017: 경험치 획득	기...	말하는 섬(9), 32646/33208	6052, Exp:486854129, 아인감소:3026, 총아인:1148920, 아인%...	다이어울프(144
19-05-27 00:01:53.876	[1] 데포로쥬,...	1202: NPC죽임	0	말하는 섬(9), 32646/33208	소환:오크(0), 0	다이어울프(144
19-05-27 00:01:57.704	[1] 데포로쥬,...	1017: 경험치 획득	기...	말하는 섬(9), 32647/33207	6052, Exp:486860181, 아인감소:3026, 총아인:1145894, 아인%...	흑기사(14424),
19-05-27 00:01:57.704	[1] 데포로쥬,...	1202: NPC죽임	0	말하는 섬(9), 32647/33207	소환:오크(0), 0	흑기사(14424),
19-05-27 00:02:03.392	[1] 데포로쥬,...	1202: NPC죽임	0	말하는 섬(9), 32645/33208	소환:오크(0), 0	흑기사(14424),
19-05-27 00:02:03.392	[1] 데포로쥬,...	1017: 경험치 획득	기...	말하는 섬(9), 32645/33208	6052, Exp:486866233, 아인감소:3026, 총아인:1142868, 아인%...	흑기사(14424),

Purpose of Churn Prediction

- Casual Analytics
  - ✓ Identifying ‘Causation of Churns’ by analyzing churn customers
  - ✓ Actually, hard work to apprehend causation using observed data (**Correalation ≠ Causation**)
- Predictive Analysis 🡐 **Goal of this competition**
  - ✓ Identifying customers who might be churned 🡒 Derive to retention via giving incentives

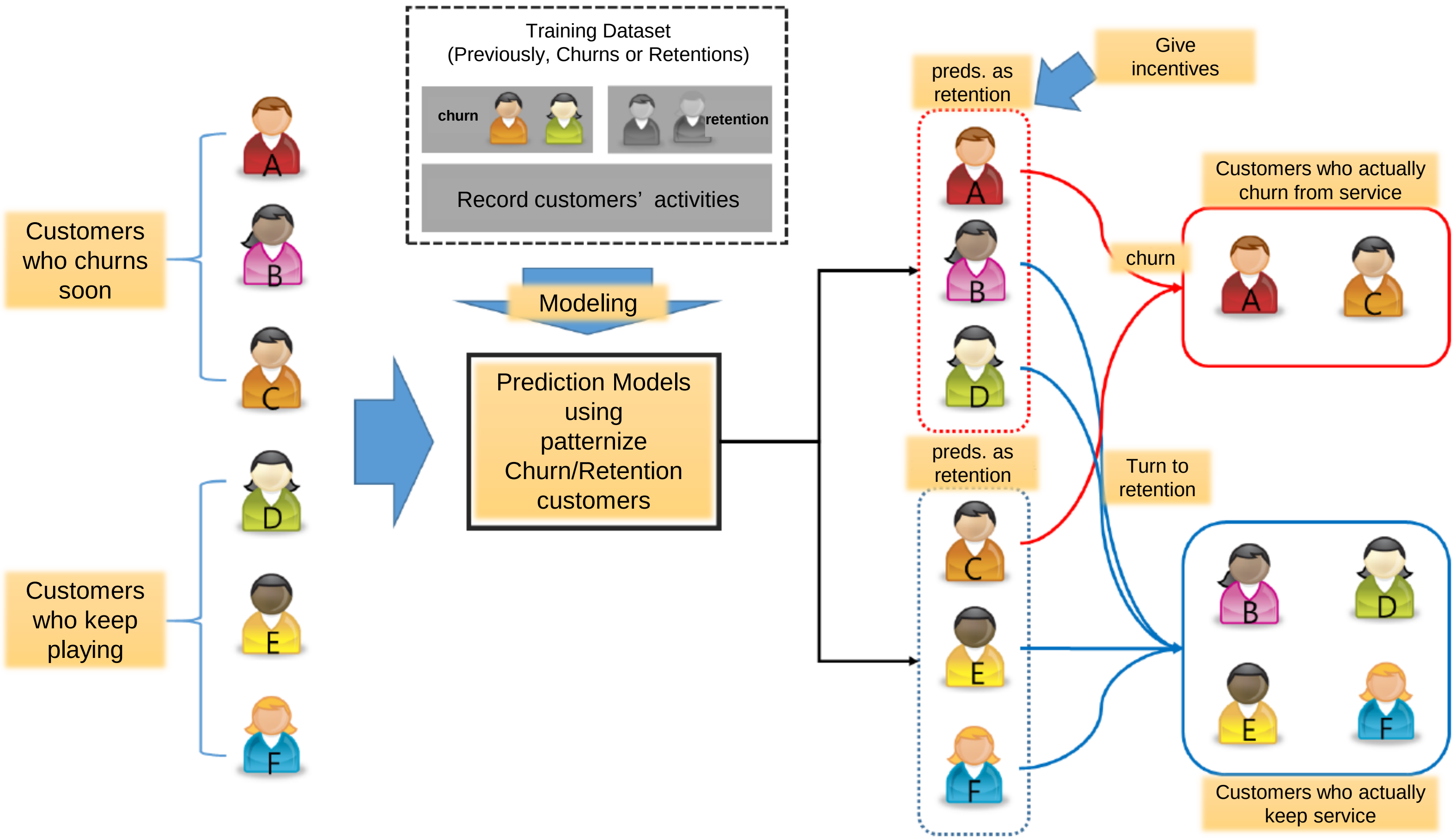


https://dai-global-developments.com/articles/disruption-for-good

Purpose of Churn Prediction (cont'd)

Customer Churn Prediction

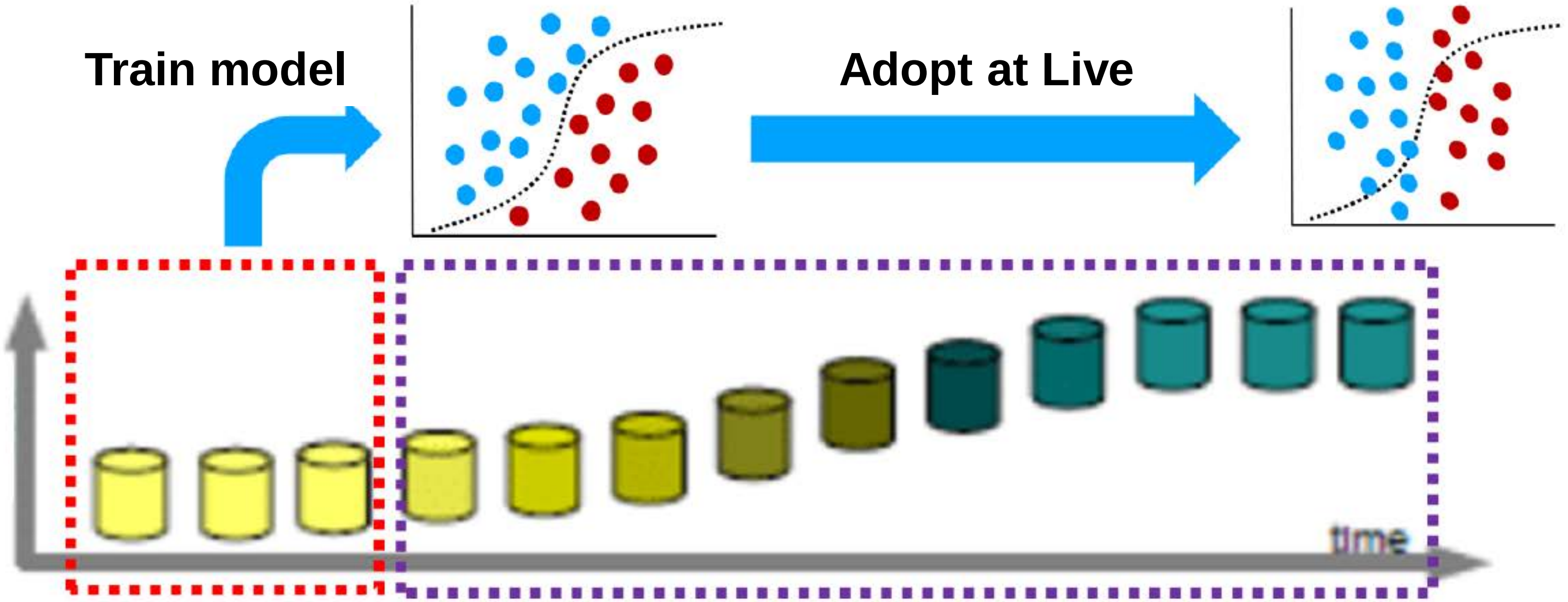
- A general scenario that adapts customers churn prediction model



Problems of general scenario #1

Lack of considering changes over time

- ✓ Changing of statistical characteristics gradually causes deterioration of model that learned from previous data patterns

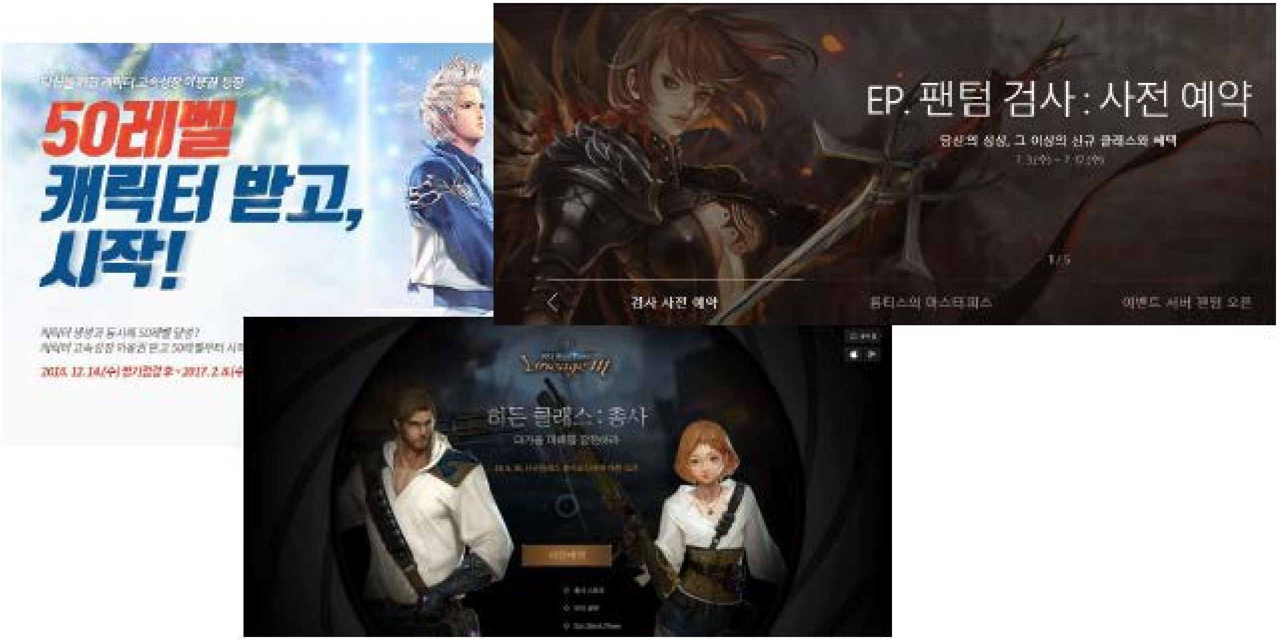


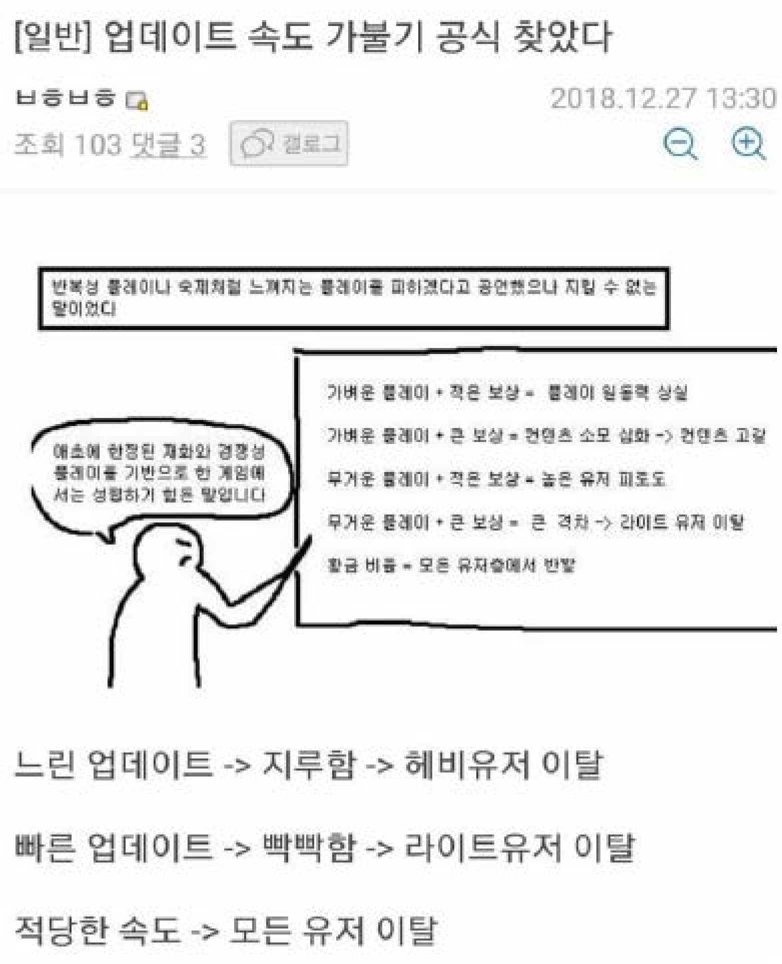
https://www.researchgate.net/publication/270787580_A_Survey_on_Supervised_Classification_on_Data_Streams

Problems of general scenario #1 (cont’d)

Characteristics of Online Game Data

- ✓ Frequent game update and events lead to…
  - Change of game balancing
  - Add or remove game contents
  - Revising business model

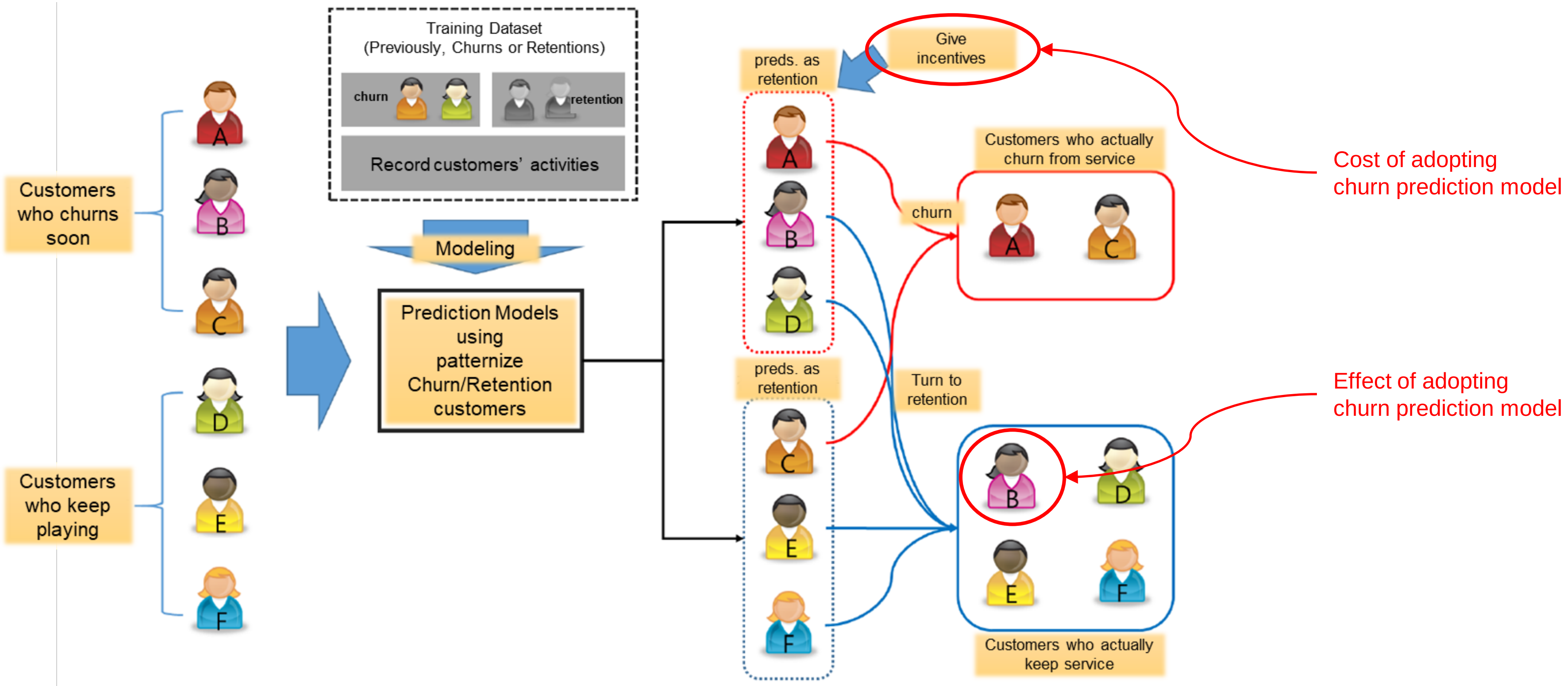




Problems of general scenario #2

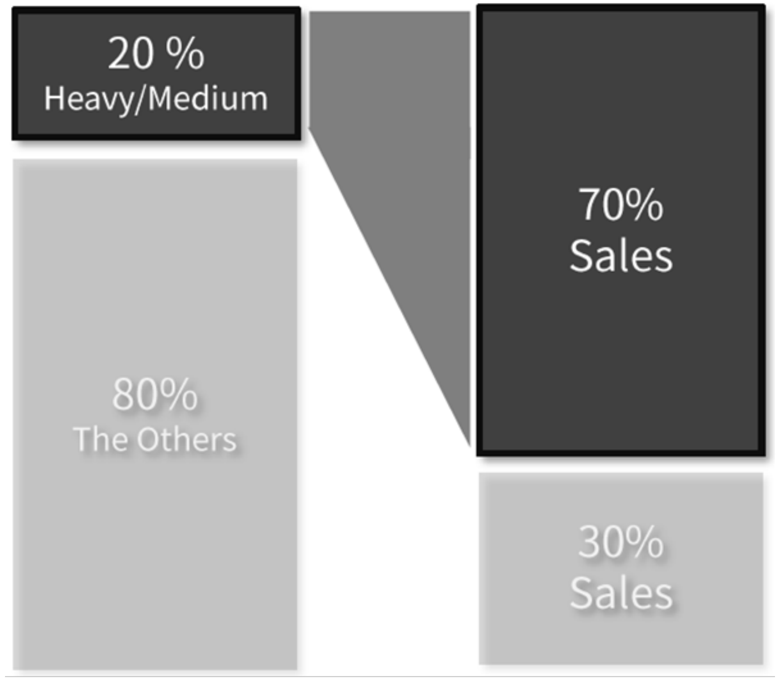
Not consider expectation profits

- The actual goal of churn prediction is not precisely prediction but keeping residual value by preventing churn
- Expectation profits $=$ effect of adopt churn prediction model – cost (accuracy $\neq$ expectation profits)



Problems of general scenario #2 (cont'd)

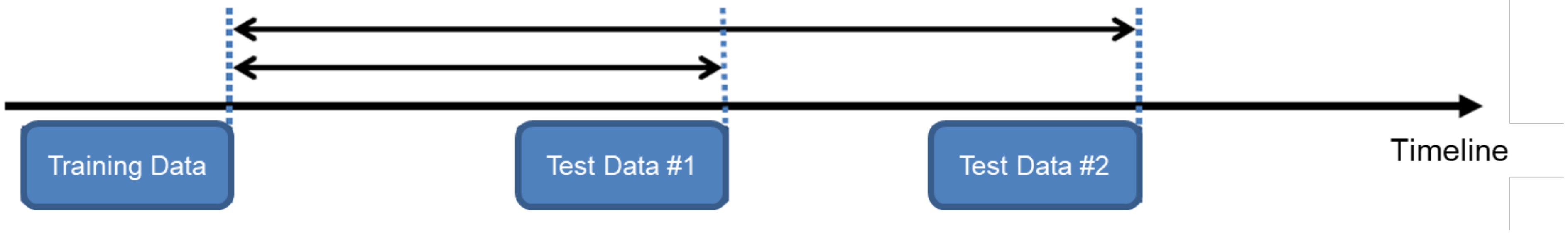
- Important to prevent churn for users who have a high residual value
  - ✓ Is it important to predict churn for malicious users?
  - ✓ How to estimate the residual value?
- Need to set proper incentive
  - ✓ If incentive is high ➔ can attract customers interests
  - ✓ If incentive is low ➔ loss will be higher than beneficial
- Also, important when does give incentive
  - ✓ No effect if miss proper timing that gives incentive



20% of customers contribute to 70% of sales benefits

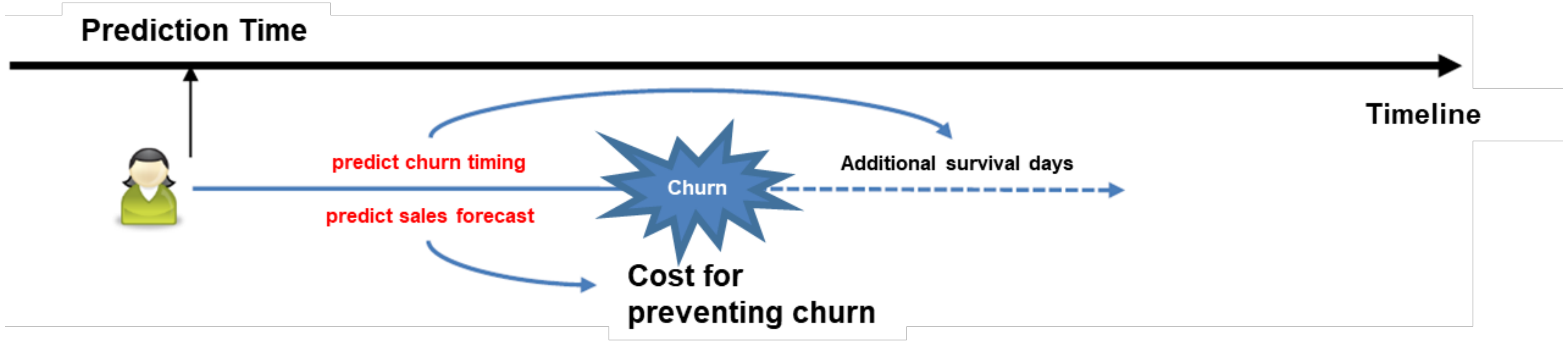
Purpose of the Problem

- Construct models that have robustness for changing time
  - ✓ Give two test dataset that is in a different timeline with train dataset



- Construct models that consider expectation profits
  - ✓ Predict churn timing and sales forecast for each user
  - ✓ Appreciate expectation profits using those two factors

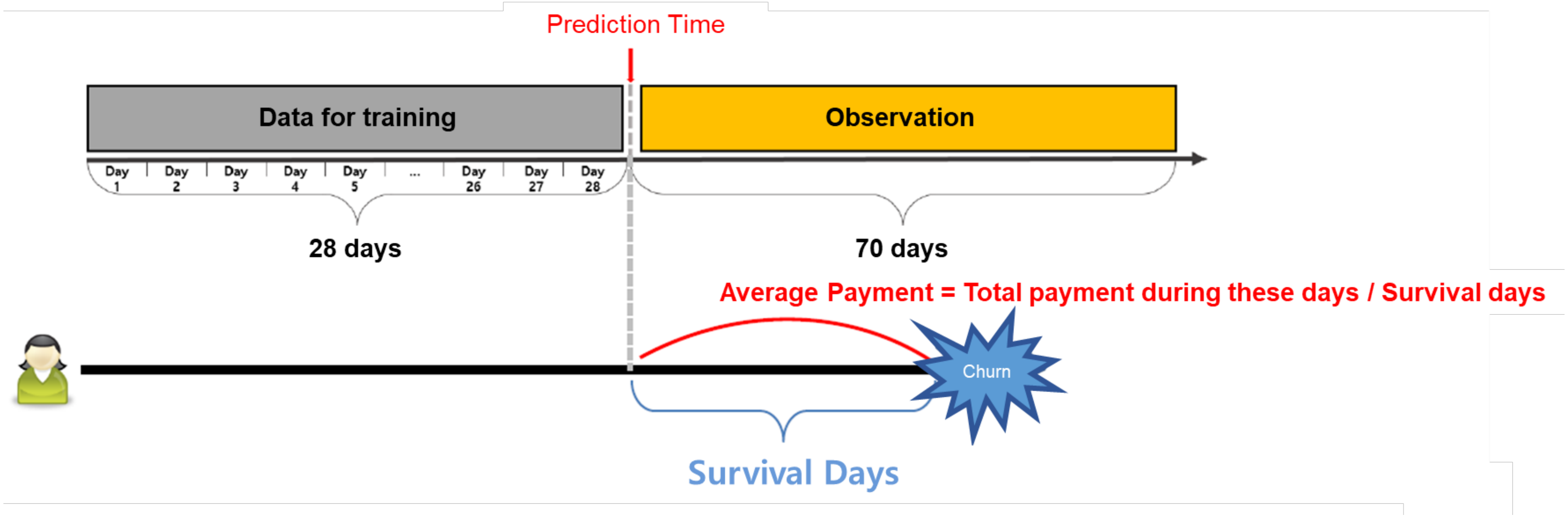
$$\text{Expectation profits} = \text{rate of change} \times (\text{additional survival days} \times \text{sales forcast}) - \text{cost for preventing churn}$$



Problem: details

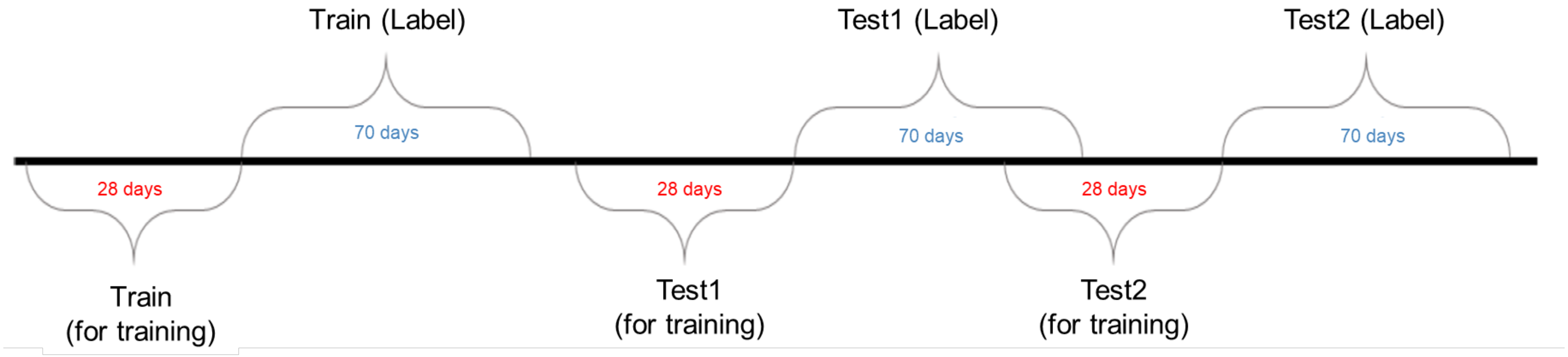
Constitute of Datasets

- Train models using last 28 days data from prediction time
- Predict churn timing(survival days) and average payment with 70 days observed data that are recorded after prediction time
  - ✓ Regard users who are not churn for 64 days as retention (consider whether churn or not with 7 days)
  - ✓ Calculate daily average payment for each user (payment happen after prediction time / survival days)



Constitute of Datasets (cont'd)

- Volume of train and test dataset
  - ✓ Train dataset: include data for 40,000 accounts
  - ✓ Test dataset 1 & 2: include data for 20,000 accounts for each



Datasets: Outline

- Can access 16 .csv files
  - ✓ Predict results for each account ID
  - ✓ Feature data use both account ID and character ID
  - ✓ One account can have one or more character

Dataset			Contents
Train	Test1	Test2	
train_label.csv	-	-	Survival days and average payment for each account
train_activity.csv	test1_activity.csv	test2_activity.csv	Activities logs for each character with account
train_combat.csv	test1_combat.csv	test1_combat.csv	PvP logs for each character with account
train_pledge.csv	test1_pledge.csv	test1_pledge.csv	Pledge combat activity logs for each character with account
train_trade.csv	test1_trade.csv	test1_trade.csv	Trade activity logs for each character with account
train_payment.csv	test1_payment.csv	test1_payment.csv	Daily average payments for each account

Datasets: Label

train_label.csv

- ✓ Survival days and daily average payments for each account
- ✓ Survival days is in between 1 to 64. (64 means retention)

Variable	Contents
acc_id	User account ID
survival_time	Survival days
amount_spent	Daily average payment

Datasets: Activity

train_activity.csv, test1_activity.csv, test2_activity.csv

- ✓ Record for daily activities for each character

Variable	Contents
day	Date
acc_id	User account ID
char_id	Character ID
server	Character server
playtime	Daily playtime
npc_kill	Number of killing Non-Player Character
solo_exp	Obtain experience by solo playing
party_exp	Obtain experience by party playing
quest_exp	Obtain experience by quest clear
rich_monster	Hit boss monster or not (0= not hit, 1= hit)
death	Number of character death
revive	Number of revival character
exp_recovery	Number of recover experience (in church)
fishing	Amount of spending time for fishing (daily)
private_shop	Amount of spending time for private shop (daily)
game_money_change	Daily fluctuation of Adena (currency in Lineage)
enchant_count	Number of Enchant for higher than 7 level items

Datasets: Trade

train_trade.csv, test1_trade.csv, test2_trade.csv

- ✓ Record for daily trading(include private shop) for each character

Variable	Contents
day	Day when happened trade
time	Time when happend trade (00:00:00 ~ 23:59:59)
type	Type of trade (trade window= 1, private shop= 0)
server	Server where happend trade
source_acc_id	Account ID who given items
source_char_id	Character ID who given items
target_acc_id	Account ID who got items
target_char_id	Character ID who got items
item_type	Type of items weapon / armor / accessory / adena (currency) / spell (skill book) / enchant_scroll
item_amount	Quantity of trading item
item_price	Price of trading : NA if trading occurs via trading window (means type=1)

Datasets: PvP

train_combat.csv, test1_combat.csv, test2_combat.csv

✓ Record daily Player vs. Player combat for each character

Variable	Contents
day	Date
acc_id	User account ID
char_id	Character ID
server	Character server
class	class (see the right table)
level	level (see the right table)
pledge_cnt	Number of combat for against with other pledges
random_attacker_cnt	Number of attack for randomly encounter user
random_defender_cnt	Number of defend for randomly encounter user
temp_cnt	Number of temporary combat
same_pledge_cnt	Number of combat for against with same pledge user
etc_cnt	Number of other combat
num_opponent	Number of opponents in combat

Category	Class
0	Monarch
1	Knight
2	Elf
3	Wizard
4	Dark Elf
5	Dragon Warrior
6	Illusioner
7	Warrior

Category	Level	Category	Level
0	1~4	9	45~49
1	5~9	10	50~54
2	10~14	11	55~59
3	15~19	12	60~64
4	20~24	13	65~69
5	25~29	14	70~74
6	30~34	15	75~79
7	35~39	16	80~84
8	40~44	17	higer than 85

Datasets: Pledge

train_pledge.csv, test1_pledge.csv, test2_pledge.csv

- ✓ Record for the pledge, who player character belongs to, members combat activity (daily)

Variable	Contents
day	Date
acc_id	User account ID
char_id	Character ID
server	Character server
pledge_id	Pledge ID
play_char_cnt	Number of pledge members who online currently
combat_char_cnt	Number of pledge members who participate in combat
pledge_combat_cnt	Number of combat for against with other pledges
random_attacker_cnt	Amount of number of attack for randomly encounter user for pledge members
random_defender_cnt	Amount of number of defend for randomly encounter user for pledge members
same_pledge_cnt	Amount of number of combat for against with same pledge user
temp_cnt	Amount of Number of temporary combat for pledge members
etc_cnt	Amount of Number of other combat for pledge members
combat_play_time	Amount of playtime for combat character in pledge
non_combat_play_time	Amount of playtime for non-combat character in pledge

Datasets: Payment

train_payment.csv, test1_payment.csv, test2_payment.csv

- ✓ Record for daily payment for each account

Variable	Contents
day	Date
acc_id	User account ID
amount_spent	Payment

Datasets: Data De-identification

For preventing expose for sensitive information, some features proceeded masking

- Marsking target: Account/Character ID, Server number
- Numerical form data have values that Origin data divided by the standard deviation

i.e)

Type	before transformation	Standard deviation	after Transformation
party_exp	2,235,212	16.4723	135695.1973919853
fishing	21	864.2	0.0242999305716269

Evaluation Metric

Evaluation

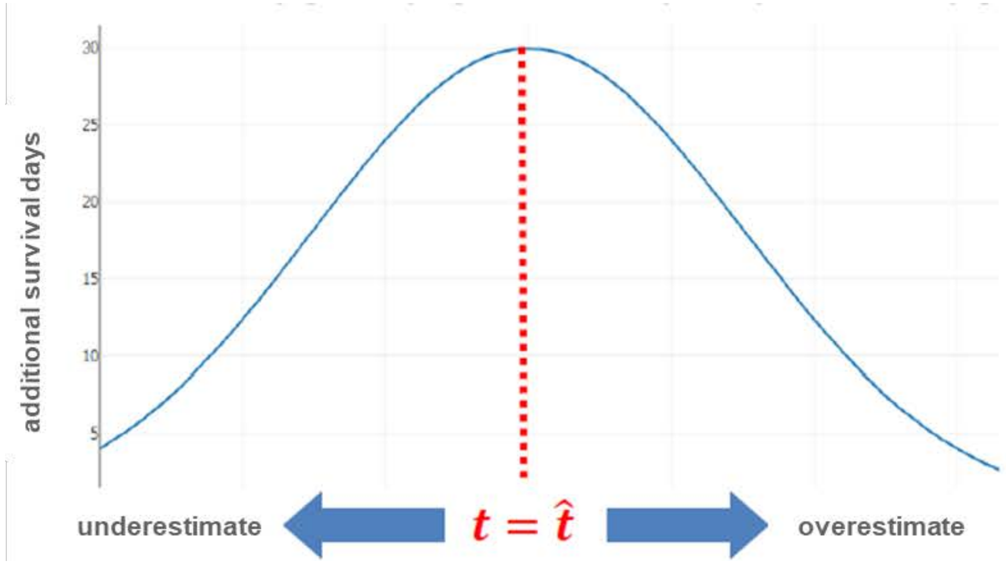
Performance of prediction + Reproducibility + Documents

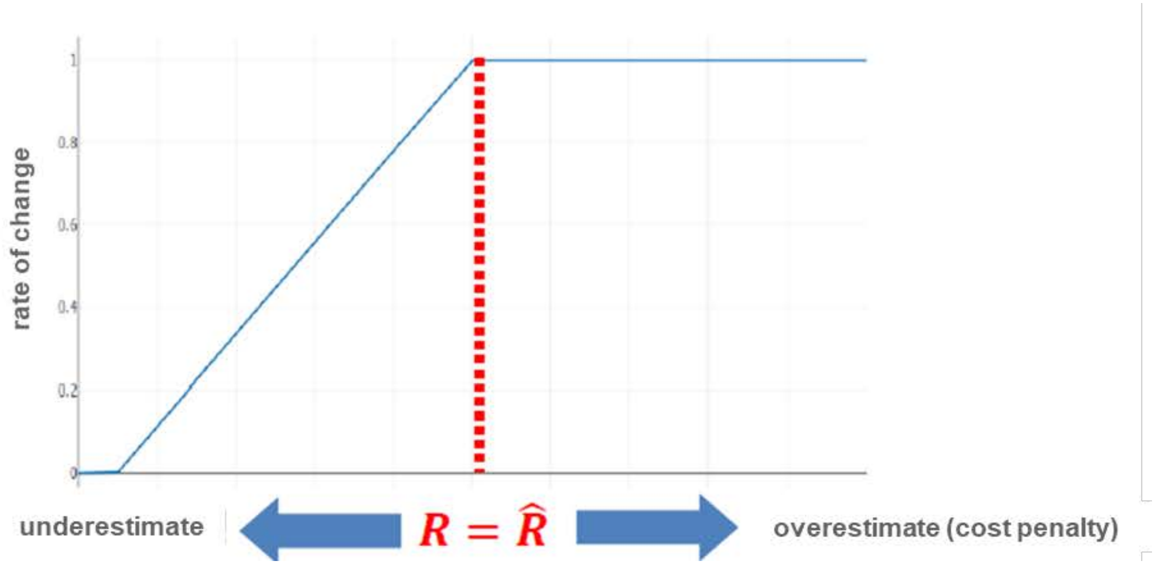
- Performance of prediction
  - ✓ Calculate expectation profits by using predicted ‘Survival day($\hat{t}$)’ and ‘Daily average payment($\hat{R}$)’
  - ✓ Acheive a higher score when the amount of expectation profits for each user bigger and bigger
- Reproducibility
  - ✓ Submit source code for every sub-process and the whole process
  - ✓ Testing how easy, accurately re-produce the result using submitted source code
- Documents
  - ✓ Documents for each subprocess (Exploratory Data Analysis, Pre-processing, Modeling and Tuning)
  - ✓ Describe how systematically and logically approach to solving the problem
    (with proper visualization)

Evaluation: Metric details

Expectation Profits = *residual value* × *rate of change* – *cost for preventing churn*

- *Residual value* = additional survival days($\boldsymbol{T}$) × daily average payment($\boldsymbol{R}$)
  - ✓ Additional Survival days determined by accuracy of prediction of survival days($\hat{t}$) (Residual value = 0, if $\hat{t} \geq 64$ *or* $t = 64$)
- *rate of change* = rate of users who are changed their mind, from churn to retention, due to reacting for incentives
  - ✓ Rate of change determined by accuracy of prediction of daily average payment($\hat{R}$)
- *cost for preventing churn* = given incentives for predicted churn users
  - ✓ cost for preventing churns setted by 1% of predicted *residual value*





Evaluation: Module

Given `score_function.py` can help self-estimate your results

Use `prediction file` and actual `label file` as parameters

Scheme of `prediction file`

Column	Desciption
acc_id	User Account ID
survival_time	predicted survival days
amount_spent	predicted daily average payment

Example of `score_function.py`

```
In [2]: from score_function import score_function
   ...: score_function('predict.csv','true.csv')
56319.66765172657
```

Appendix

Fomulation for estimate *expectation profits*

- expectation profits $=$ residual value $\times$ rate of change $-$ cost for preventing churn
- residual value = additional survival days(T) $\times$ daily average payment(R)
  - ✓ $T = \begin{cases} 0, if\ \hat{t} = 64\ or\ t = 64 \\ 30 \times e^{-\frac{(t-\hat{t})^2}{2\times 15^2}}, otherwise \end{cases}$
  - ✓ $\hat{t}$: predicted survival days, t: actual observed survival days
- cost for preventing churn(C) = given incentives for predicted churn users
  - ✓ $C = \begin{cases} 0,\ \text{if a user predicted as retention or predicted daily average payment} \\ 0.01 \times 30 \times \hat{R}, \text{if a user predicted as churn} \end{cases}$
  - ✓ $\hat{R}$: predicted daily average payment
- rate of change(γ) = rate of users who are changed their mind, from churn to retention, due to reacting for incentives
  - ✓ $\gamma = \begin{cases} 0, if\ \hat{C} < \frac{C_{opt}}{10}\ or\ C_{opt} = 0 \\ \frac{10}{9}\left(\frac{\hat{C}}{C_{opt}} - 0.1\right), otherwise \end{cases}$
  - ✓ $\hat{C}$: predicted cost, C_{opt}: proper cost ($R = \hat{R}$)