Data Analysis Report: User Features That Contribute The Most To Their Repurchase Rate

Crimson Kim Minerva Capstone Project X Class101

Table of Contents

I. The WHY: Context

> → The Big Question: Where should Class101 focus their resources in order to increase their customer repurchase rate? (= Which user characteristics and behaviors contribute the most to their likeliness of repurchasing another product?)

▼ Other Relevant Questions

1. What kind of user data is currently accessible (for tracking & aggregation)?

   → First, we need to pre-process various user characteristics & behavior data for each user

2. Can we identify a more direct and useful relationship than just a simple correlation?

   → Let's borrow the power of statistical learning!

▼ (Reference) What are the advantages and disadvantages of general data analysis and data science-based analysis?

- Data Science Main Pros

1. Individual analysis tasks, like in the example below, can be analyzed all at once (individual tasks that would be carried out separately and in a fragmented way by several teams)
    - i) Relationship between the number of posts and repurchase
    - ii) Relationship between web vs app usage and repurchase
    - iii) The relationship between the package type of the first purchased product and repurchase
    - iv) Theoretically, hundreds of other assumptions to test
2. Minimize the influence of confounding variables by analyzing all of them at once
3. Can rank the relative contribution of each individual variables to the goal variable

- Data Science Main Cons
    1. Data Collection & Processing takes a lot of effort/time/labor
    2. Data Science Insights Required (Statistics, Coding, Polishing, etc.)
    3. What is the direct relationship between the variables with high repurchase contribution and the repurchase rate? → A look at the Scatter Plot based on variables with high contribution to repurchase

▼ What is Class101?

- CLASS101 USA Website

- Summary: Online Class Platform, "something like a blend of MasterClass and Coursera, but made more hip and fun"

▼ Size Stats

  - 2.5M Registered Users, 400k Payed Users (each class costs $300 on average)
  - Received $40M funding so far (Series A & B), makes around $100M per year

II. The HOW: Modelling Process

> → 20-30 different characteristics and behavior data of ~50,000 users who made their first purchase experience at the end of last year, were put into various machine learning models, and we looked at which variables contributed (relatively) the most to their likeliness of repurchasing our product

▼ User Group (Dataset Row Count ~= 50,000)

- New Users (First Ever Class Start Date between the two dates below)
  - 2020.09.01 ≤ Class Start Date < 2020.12.01
- There were around 50,000 users after cutting off weird data

▼ X (Features): User Characteristics & Behavior (Dataset Column Count = 84)

- 20-30 user variables (characteristics & behavioral data) were collected
- Additional processing of some variables based on the criteria below:
  - For 1 day / 3 days / 10 days / 30 days / 90 days / 150 days after the Class Start Date
    - As an example, one variable called "completed_electures" was divided into 6 more columns, for each period
- Qualitative data is converted into 0 and 1 variables:

    - For example, a variable "brand" is divided into four variables this way: brand_career, brand_creative, brand_money, brand_others

▼ Reference the full list

```
Data columns (total 84 columns):
 #   Column                                    Non-Null Count  Dtype
---  ------                                    --------------  -----
 0   completed_lectures_ratio_1day             51052 non-null  float64
 1   completed_lectures_ratio_3days            51052 non-null  float64
 2   completed_lectures_ratio_10days           51052 non-null  float64
 3   completed_lectures_ratio_30days           51052 non-null  float64
 4   completed_lectures_ratio_90days           51052 non-null  float64
 5   completed_lectures_ratio_150days          51052 non-null  float64
 6   nonmission_posts_1day                     51052 non-null  int64
 7   nonmission_posts_3days                    51052 non-null  int64
 8   nonmission_posts_10days                   51052 non-null  int64
 9   nonmission_posts_30days                   51052 non-null  int64
 10  nonmission_posts_90days                   51052 non-null  int64
 11  nonmission_posts_150days                  51052 non-null  int64
 12  posts_1day                                51052 non-null  int64
 13  posts_3days                               51052 non-null  int64
 14  posts_10days                              51052 non-null  int64
 15  posts_30days                              51052 non-null  int64
 16  posts_90days                              51052 non-null  int64
 17  posts_150days                             51052 non-null  int64
 18  comments_1day                             51052 non-null  float64
 19  comments_3days                            51052 non-null  int64
 20  comments_10days                           51052 non-null  int64
 21  comments_30days                           51052 non-null  int64
 22  comments_90days                           51052 non-null  int64
 23  comments_150days                          51052 non-null  int64
 24  cheers_before                             51052 non-null  int64
 25  cheers_10days                             51052 non-null  int64
 26  cheers_30days                             51052 non-null  int64
 27  cheers_90days                             51052 non-null  int64
 28  cheers_150days                            51052 non-null  int64
 29  wish_before                               51052 non-null  int64
 30  wish_10days                               51052 non-null  int64
 31  wish_30days                               51052 non-null  int64
 32  wish_90days                               51052 non-null  int64
 33  wish_150days                              51052 non-null  int64
 34  completed_class                           51052 non-null  int32
 35  is_hero                                   51052 non-null  int32
 36  is_branded                                51052 non-null  int32
 37  amount                                    51052 non-null  int32
 38  order_ticket_diff                         51052 non-null  int64
 39  lv_count_1day                             51052 non-null  int64
 40  lv_count_3days                            51052 non-null  int64
 41  lv_count_10days                           51052 non-null  int64
 42  lv_count_30days                           51052 non-null  int64
 43  lv_count_90days                           51052 non-null  int64
 44  lv_count_150days                          51052 non-null  int64
 45  lv_web_ratio_150days                      51052 non-null  float64
 46  lv_ios_ratio_150days                      51052 non-null  float64
 47  lv_android_ratio_150days                  51052 non-null  float64
 48  lv_app_ratio_150days                      51052 non-null  float64
 49  lv_distinct_hours_150days                 51052 non-null  int64
 50  count_distinctdays_3days                  51052 non-null  int64
 51  count_distinctdays_10days                 51052 non-null  int64
 52  count_distinctdays_30days                 51052 non-null  int64
 53  count_distinctdays_90days                 51052 non-null  int64
 54  count_distinctdays_150days                51052 non-null  int64
 55  rebuy_182days                             51052 non-null  float64
 56  discount_ratio                            51052 non-null  float64
 57  brand_career                              51052 non-null  uint8
 58  brand_creative                            51052 non-null  uint8
 59  brand_money                               51052 non-null  uint8
 60  brand_other                               51052 non-null  uint8
 61  boundness_inbound                         51052 non-null  uint8
 62  boundness_outbound                        51052 non-null  uint8
 63  internal_category_art                     51052 non-null  uint8
 64  internal_category_career                  51052 non-null  uint8
 65  internal_category_craft                   51052 non-null  uint8
 66  internal_category_digitalDrawing          51052 non-null  uint8
 67  internal_category_language                51052 non-null  uint8
 68  internal_category_mindAndSelfDevelopment  51052 non-null  uint8
 69  internal_category_money                   51052 non-null  uint8
 70  internal_category_other                   51052 non-null  uint8
 71  package_name_allin                        51052 non-null  uint8
 72  package_name_coaching                     51052 non-null  uint8
 73  package_name_only                         51052 non-null  uint8
 74  package_name_others                       51052 non-null  uint8
 75  is_event_other                            51052 non-null  uint8
 76  is_event_price101                         51052 non-null  uint8
 77  is_event_tutorial101                      51052 non-null  uint8
 78  difficulty_중급                             51052 non-null  uint8
 79  difficulty_상급                             51052 non-null  uint8
 80  difficulty_초급                             51052 non-null  uint8
 81  difficulty_초중급                            51052 non-null  uint8
 82  klass_state_basuga                        51052 non-null  uint8
 83  klass_state_earlybird                     51052 non-null  uint8
dtypes: float64(13), int32(4), int64(40), uint8(27)
```

▼ Y (Goal Variable): Rebuy (Boolean)

- Candidate 1. rebuy_182days → Used as the main Y (Reason: The repurchase rate is so low so priority was put to determining whether a repurchase has even been made or not)
  - Did each user in the above user group repurchase tickets within 182 days after their ticket start date? (1 or 0)
- Candidate 2. sum_revenue_182days
  - Total purchase amount for 182 days after Class Start Date for each user in the above User Group (not net transaction)

▼ Fitted Models

	Test Score	Model
0	80.69	Random Forest
1	76.31	Logistic Regression
2	75.04	Naive Bayes
3	73.79	Perceptron
4	73.10	Stochastic Gradient Decent
5	72.18	Decision Tree
6	71.69	KNN

- Among them, Random Forest had the highest test_accuracy_score, so Random Forest was used as the main model
- None of the score were that high, however perfection was not the main goal for the initial iteration of the project, so proceeded as is

III. The WHAT: Results

> → (In order of contribution) The number of class attendance, progress rate, price, device used, participation in the cheering and wishing function, and community activity were the user behavior & characteristic variables that relatively contributed the most to class

> repurchase. It would be better to check the relationship between these variables and repurchase through the Scatter Plots below.

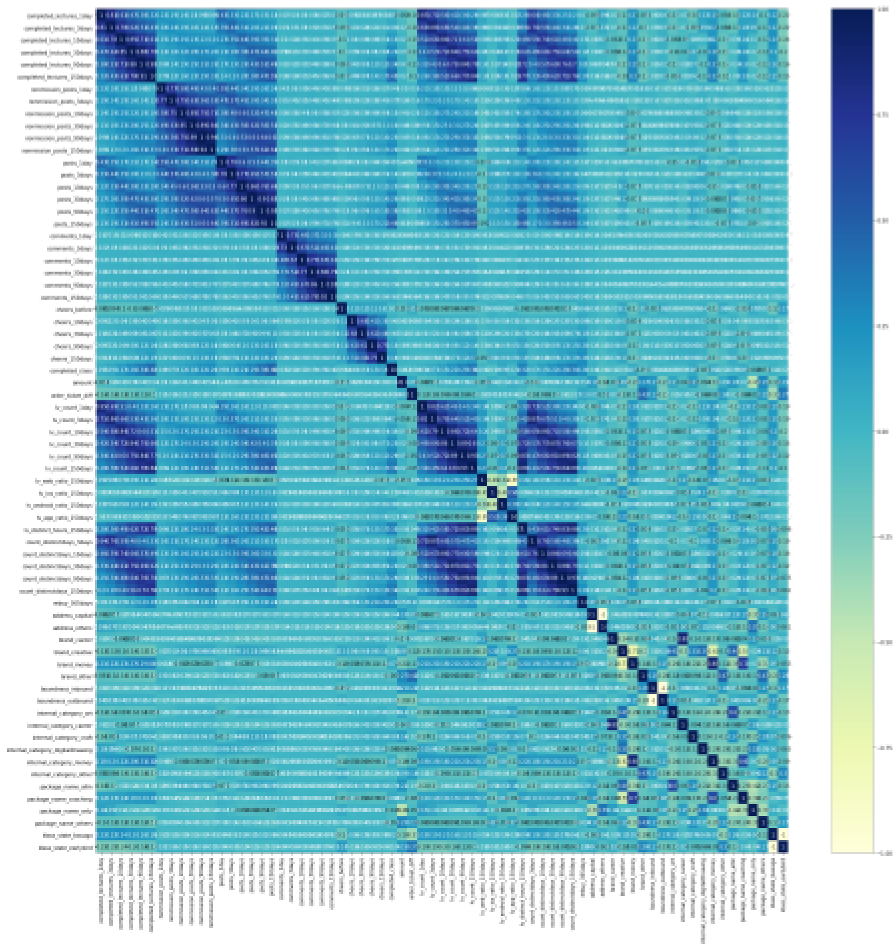
Simple Correlations (Just for reference)

▼ Top 20 Xs with the highest correlation with Y

wish_150days	32.54
wish_90days	28.76
lv_count_150days	25.57
count_distinctdays_150days	25.00
lv_distinct_hours_150days	24.22
cheers_150days	23.24
posts_150days	23.19
wish_30days	21.84
cheers_90days	20.74
count_distinctdays_90days	20.58
lv_count_90days	20.47
posts_90days	18.81
wish_10days	15.80
nonmission_posts_150days	15.55
cheers_30days	15.41
count_distinctdays_30days	13.23
nonmission_posts_90days	12.73
wish_before	12.48
lv_count_30days	12.43
order_ticket_diff	12.40

- Currently, individual correlation coefficients of variables are not very high, so we need to collect more Xs (need creativity/questions from other colleagues, as well as with data processing/engineering)

▼ (Reference) Correlation Matrix: Correlation of all variables (there are too many features to show all the numbers)



Mutual Information Correlations (Just for reference)

▼ *Mutual Information and Entropy*

- To truly understand the relationship between X (features) and Y, we not only have to study the covariance or movement of the data – as Pearson's does by comparing how the value of one variable changes as the other variable changes – we also have to see the *Information* shared between the two variables. This is when I came across a useful concept and tool known as Mutual Information. Let's dissect the term one by one: First, what is Information? There are many colloquial as well as scientific ways of describing this term, but the appropriate one for the context of this paper is that it captures the amount of "surprise" (noted as *surprisal)* contained in an observation of data (McClure, 2020). The more surprisal associated with a variable the more information that variable contains. We can quantify this surprisal through another useful concept and tool known as Entropy. Entropy is the average level of Information or uncertainty inherent to a variable’s possible outcomes.

- Second, what does *Mutual* entail in Mutual Information? It denotes the dependency of two or more variables, which measures the *coincidence* of the configurations between the things that are being measured (McClure, 2020). When we observe something we are being exposed to some source of information, and we can use algorithms or models to exploit that information to explain or predict something more general. Further, we can redefine Mutual Information using Entropy as such:
    - MI(X;Y) = Entropy(X) - Entropy(X|Y)
- Hence, Mutual Information measures the entropy drops under the condition of Y (represented as Entropy(X|Y)) (Zhu, 2021). In conclusion, this means that the higher the Mutual Information, the closer connection between X and Y, which suggests that we should put this feature in the training dataset.

▼ *Concepts Applied In Practice*

- Using this useful tool, I carried out a Feature Selection process through Mutual Information instead of fitting all features into the models. First, if we compute the Mutual
- Information Scores (mi_score below) of the 84 columns. The 15 highest scores are shown here:

feature selection through mutual information, before fitting the models

In [216]:
```
from sklearn.feature_selection import mutual_info_classif as MIC

#compute the mutual info scores
mi_score = MIC(X,Y, discrete_features=False)
```

In [249]:
```
#constructing a dataframe that maps the mi scores with the X_columns
d = {'X_columns':list(X.columns), 'mi_scores': list(mi_score)}
mi_scores_map = pd.DataFrame(d)

mi_scores_map.sort_values('mi_scores',ascending=False).head(15)
```

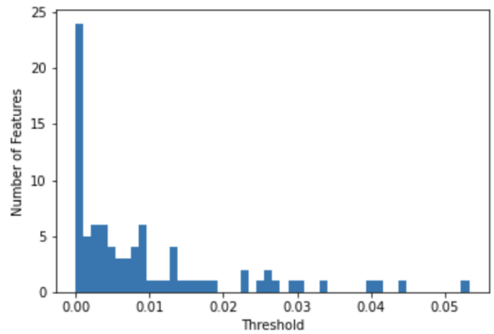
Out[249]:

	X_columns	mi_scores
34	wish_150days	0.053250
33	wish_90days	0.044131
18	posts_150days	0.040500
45	lv_count_150days	0.039410
55	count_distinctdays_150days	0.033279
50	lv_distinct_hours_150days	0.030667
29	cheers_150days	0.029806
32	wish_30days	0.027030
54	count_distinctdays_90days	0.025692
17	posts_90days	0.025618
44	lv_count_90days	0.025107
49	lv_app_ratio_150days	0.023049
46	lv_web_ratio_150days	0.022658
28	cheers_90days	0.018740
39	order_ticket_diff	0.017327

- As you can see, the MI scores are not so high, signifying that the dependency of information between Xs and Y are not that high from the start, which can be problematic.
- This histogram shows the number of features that are higher than a varying level of thresholds for the minimum MI Scores:

In [248]:
```
plt.hist(mi_scores_map['mi_scores'], bins=50)
plt.xlabel('Threshold')
plt.ylabel('Number of Features')
```

Out[248]: Text(0, 0.5, 'Number of Features')



- We can observe that most of the features fall within the thresholds 0<threshold<0.02. I use this as a reference to choose the input threshold values for the feature selection.
- Finally, when I compute the prediction scores for each corresponding threshold value, I can see that the threshold value of 0.005 scores the highest score out of the other options. However, unfortunately, an increase of the prediction score from 80.24% (originally) to 81.01% is not as significant, and thus it is concluded that Feature Selection through Mutual Information this project was not so useful in the end. A further analysis on why this was the case can be helpful in the future. A possible reason for this is because the MI scores were not so high from the start, and diggin deeper into this clue can enlighten some more interesting discoveries of the user data for this project.

```
In [246]:

#minimum threshold for the mutual information scores to be selected
thresholds = [0, 0.001, 0.003, 0.005, 0.007, 0.01, 0.015, 0.02, 0.05]

for i in thresholds:
    mi_scores_map_selected = mi_scores_map[mi_scores_map['mi_scores']>i]

    X_2 = X[mi_scores_map_selected['X_columns']]
    X_train_2, X_test_2, Y_train_2, Y_test_2 = train_test_split(X_2, Y, test_size=0.20,
                                                                random_state=42)

    random_forest = RandomForestClassifier(n_estimators=100)
    random_forest.fit(X_train_2, Y_train_2)
    Y_pred_train_2 = random_forest.predict(X_test_2)
    Y_pred_test_2 = random_forest.predict(X_test_2)
    acc_random_forest = round(random_forest.score(X_test_2, Y_test_2) * 100, 2)


    print("Threshold: ", i, " Accuracy Score: " acc_random_forest)
```

```
(Threshold, Accuracy Score):  0 80.69
(Threshold, Accuracy Score):  0.001 80.75
(Threshold, Accuracy Score):  0.003 80.69
(Threshold, Accuracy Score):  0.005 81.01
(Threshold, Accuracy Score):  0.007 80.46
(Threshold, Accuracy Score):  0.01 80.57
(Threshold, Accuracy Score):  0.015 80.33
(Threshold, Accuracy Score):  0.02 78.6
(Threshold, Accuracy Score):  0.05 75.79
```

Random Forest Model Scores (Just for reference)

▼ Accuracy Scores

```
Train Score 99.95 %
Test Scores Mean: 80.24
Test Scores: [0.79998531 0.80358455 0.80364358]
Test Scores Standard Deviation: 0.0017107795695389992
```

▼ Confusion Matrix

```
[[7132  332]
 [1644 1103]]
              precision    recall  f1-score   support

           0       0.81      0.96      0.88      7464
           1       0.77      0.40      0.53      2747

    accuracy                           0.81     10211
   macro avg       0.79      0.68      0.70     10211
weighted avg       0.80      0.81      0.78     10211

F1 Score 0.5274988043998087
```

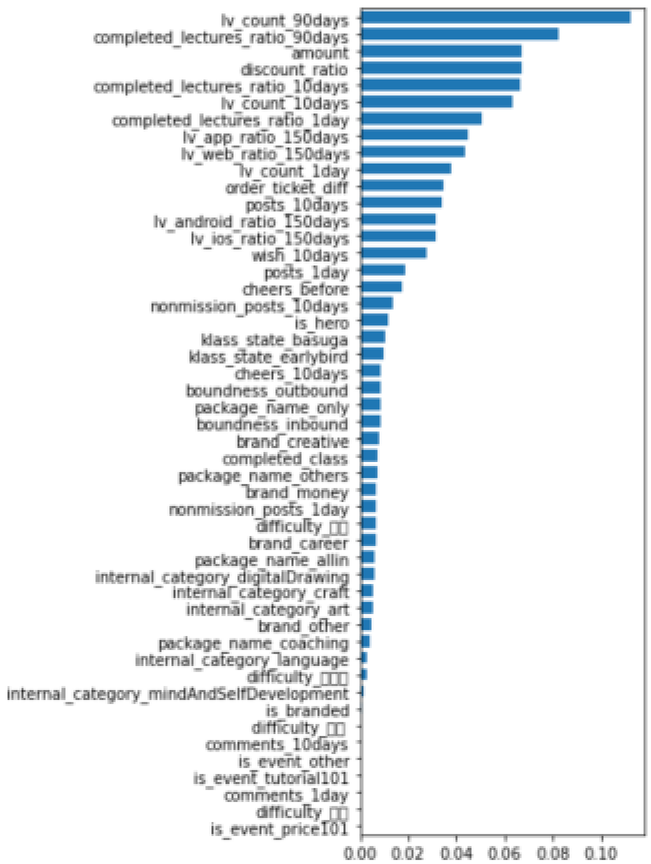
▼ There are definitely room for improvement

- Increase number of rows
- find better Xs

- Come up with derived Xs
- Think of ways to increase f-1 scores
- Further elimination of outlier values
- User segmentation

Random Forest Permutation Feature Importance Scores

- Permutation Feature Importance (Contribution of each of the X variables to Y)

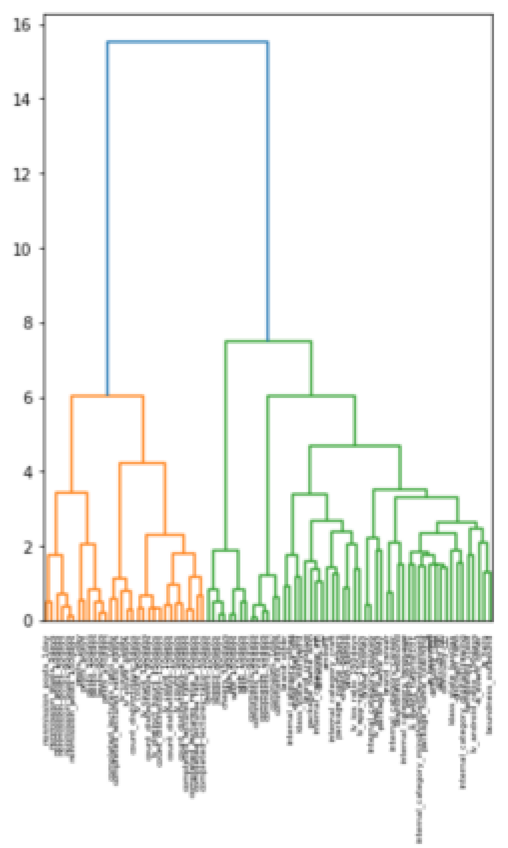


→ **Features such as attendance retention and class progress rate, price (class price and discount rate), device type, number of comments, likes, and support, appear to have an effect on increasing customer repurchase**

→ Conversely, it can also be seen that variables that did not have a high Importance Score did not significantly contribute to the repurchase rate: Example — whether the customer initially bought the class as an early bird or after the launch, the class brand/category, whether the customer initially bought the product through promotion/event

▼ What is a Permutation Feature Importance?

- It is possible to know which variables contributed the most to the Random Forest prediction model, by ranking the variables in terms of Feature Importance.
- However, if Xs are numeric variables, you should not use the basic Feature Importance method, but use Permutation Feature Importance.
- To proceed with this, new variables must be selected by minimizing the correlation between X (draw a dendrogram to identify the correlation hierarchy, then cut off variables with high correlations through this, hierarchy level=1)



Scatter Plots — Showing a more detailed relationship between the High Contribution Xs against the Y

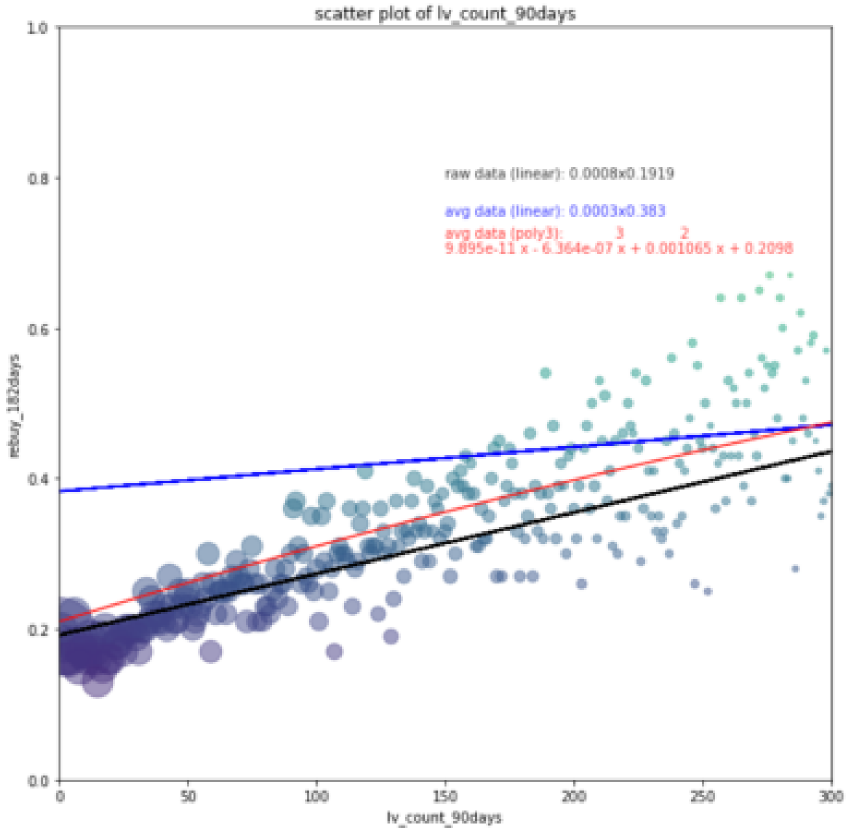
- **FYI**
  - The average repurchase rate of 50,000 users for 182 days is about 26%.
  - Putting "1% increase in repurchase rate" into scale
    - If the repurchase rate of 300,000 payed users rises to 1%, with an average class price at 300,000 won, for example:
      - 300,000 Users * 300,000 KRW * 0.01 Rebuy Rate * 2 (simple conversion based on one year)
      - = 1.8 billion KRW (1.5M USD) per year sales increase!!
      - (Of course, depending on the situation, it is not possible to substitute 100% of the above formula in reality, but it is good to refer to it roughly)
    - In the graphs below, a 1% increase in the repurchase rate does not seem like a big deal, but in terms of the amount, it has a huge impact.
  - Plot Legend
    - Dots
      - Each dots represent average repurchase rate for each X points
      - Color: The higher the repurchase rate, the closer to the bright color (red-orange-yellow) among the rainbow colors
      - Size: User Count
    - Lines
      - Black: The linear relationship between each RAW data points
      - Blue: The linear relationship between each AVERAGED data points
      - Red: The Polynomial line (third degree here) for the AVERAGED data points
      - (polynomial?)
        - Give curves to trendline to reduce generalization and connect data points more detailedly
        - the degrees in polynomial - most number of times a function will cross the x-axis

▼ <A Lecture Attendance Frequency & Retention Related Data>

A-1) Lecture View Count Within 90 days

...

0.0003x0.383

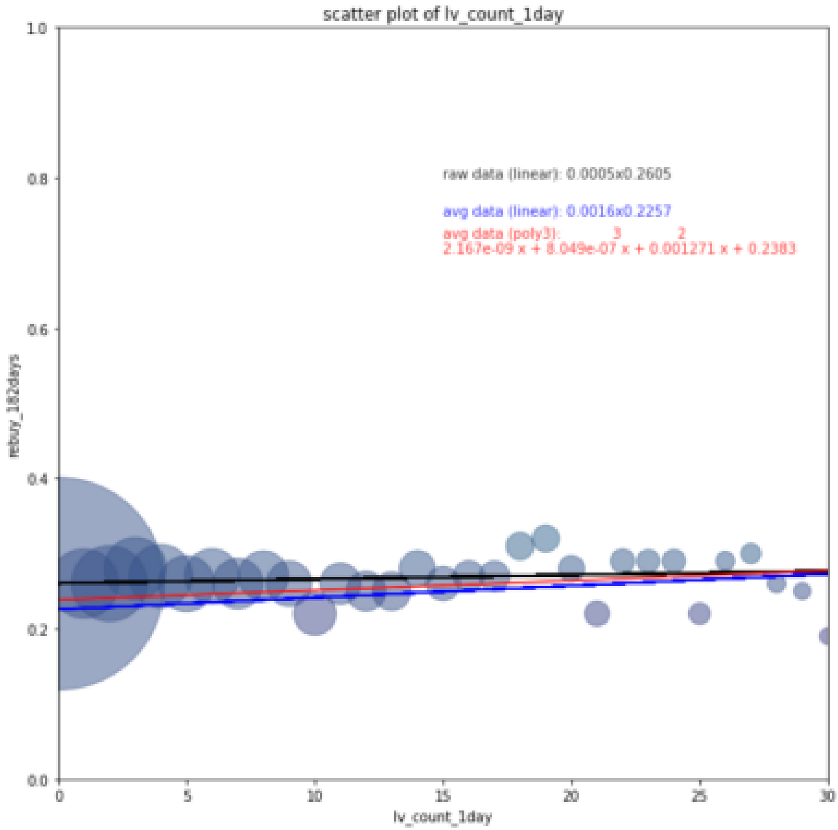


- The Lecture View Count, which contributed the most, showed a fairly linear relationship with the repurchase rate: as the number of class attendance (and revisit count) increases, the repurchase rate also rises steadily.

A-2) Lecture View Count Within 24 hours

...

```
0.0016x0.2257
```

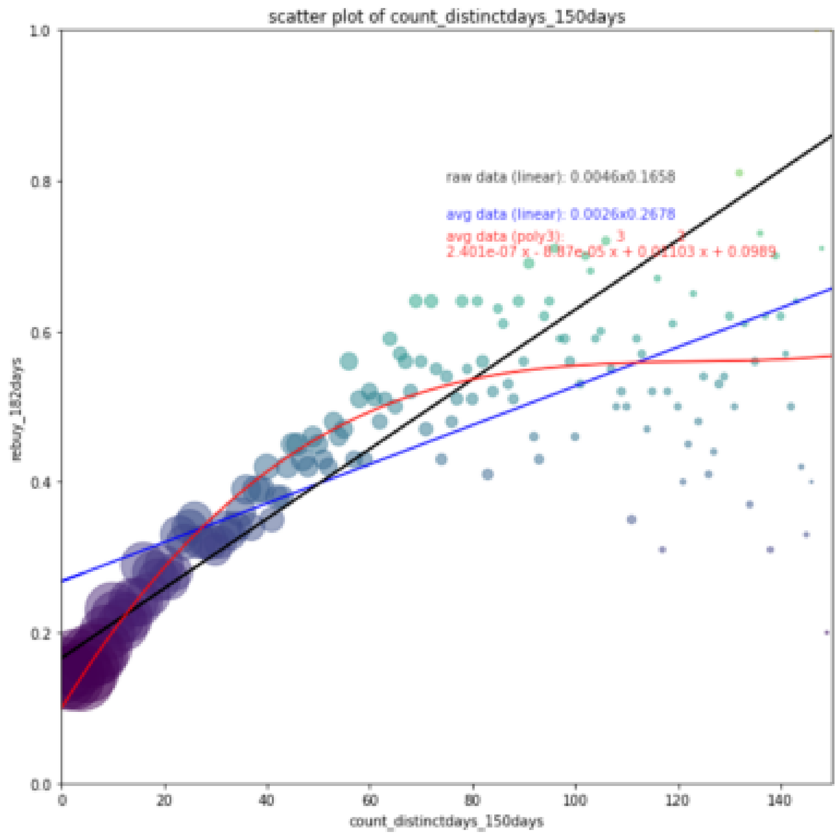


- The number of times users opened the lectures within the first 24 hours does not seem to have a significant effect on the repurchase rate (there is no noticeable pattern)

A-3) Distinct Lecture View Days Within 150 Days

...

```
0.0026x0.2678
```

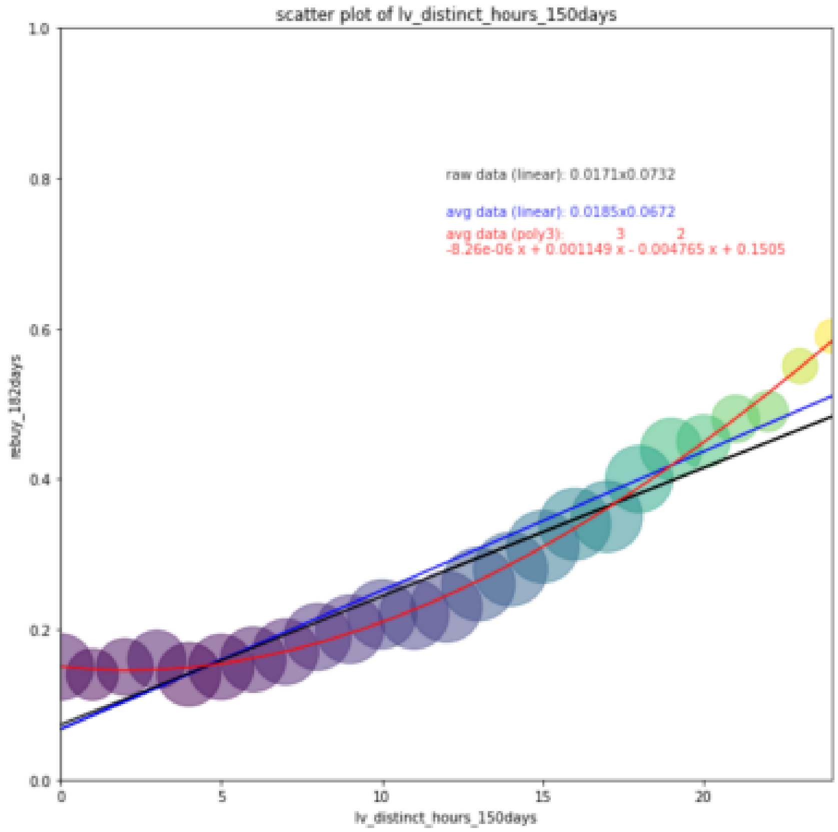


- As Distinct Days visited during 150 days since first purchase increases, the repurchase rate increases linearly
- The higher the retention, the higher the repurchase rate.
- Even if you visit all 150 days unconditionally, the repurchase rate does not increase that much — customers would just have to visit 30-40% of the days (50-60 days out of 150 days) to have a high (50%+) repurchase rate

A-4) Distinct Lecture View Hours Within 150 Days

* * *

0.0185x0.0672



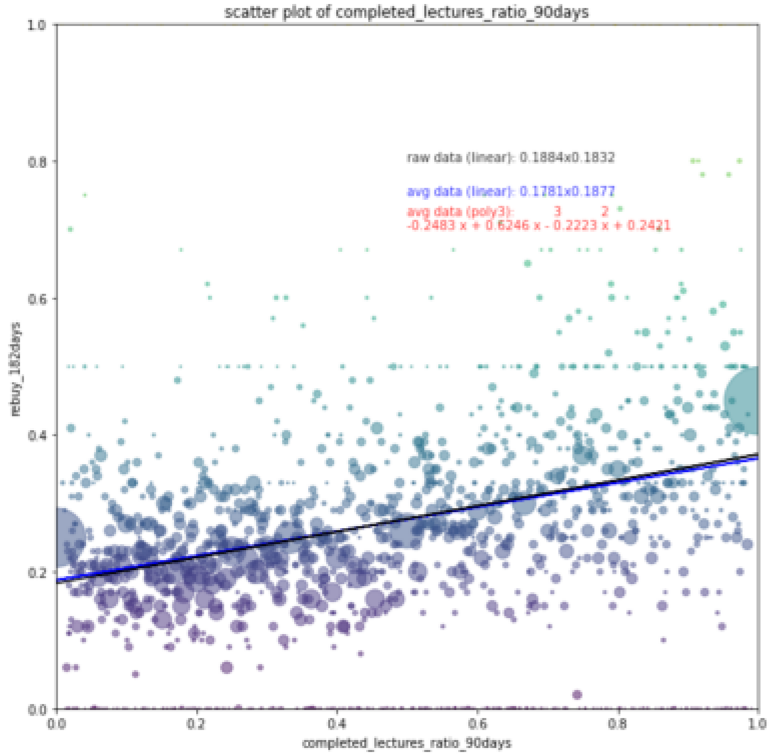
- Reference data: The more varied the Distinct Hours taken for 150 days, the higher the repurchase rate (Perhaps it's due to an already existing correlation of lecture attendance count and diversity of hours customers took class

▼ <B Class Progress Related Data>

B) Completed Lecture Ratio Within 90 Days

...

0.1781x0.1877



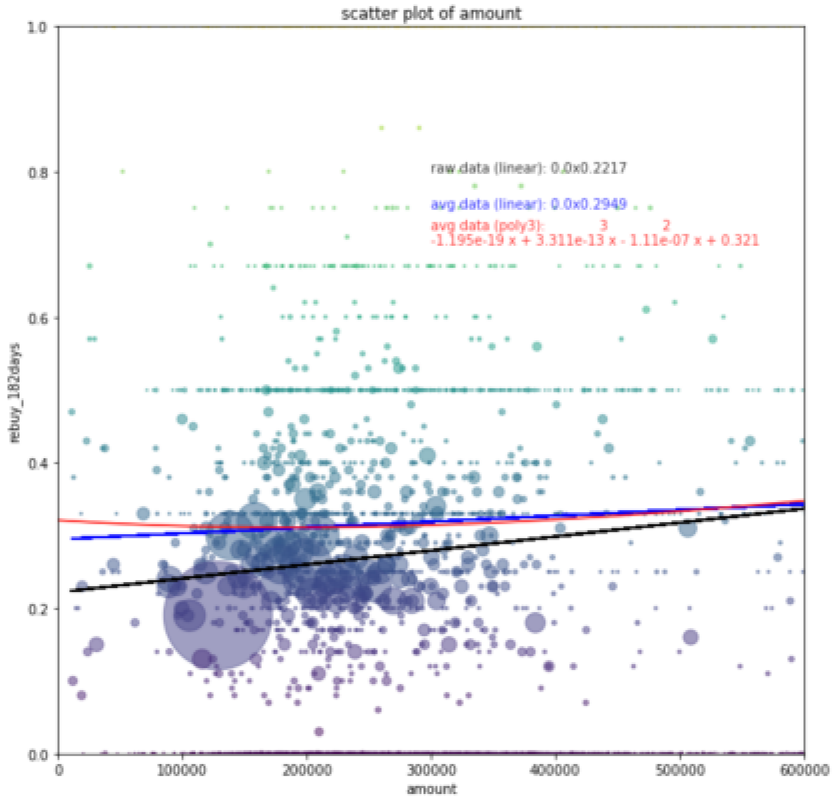
- The higher the class progress rate, the higher the repurchase rate, quite linearly
- **Among them, the repurchase rate (about 45%) of users who achieved progress rate of 1.0 (aka class completed) seems to be a very significant result!!**
    - → **There are a whole lot of users (4000), whilte the average of 45% (almost double of the average repurchase rate of 26%**

▼ **<C Class Price Related Data>**

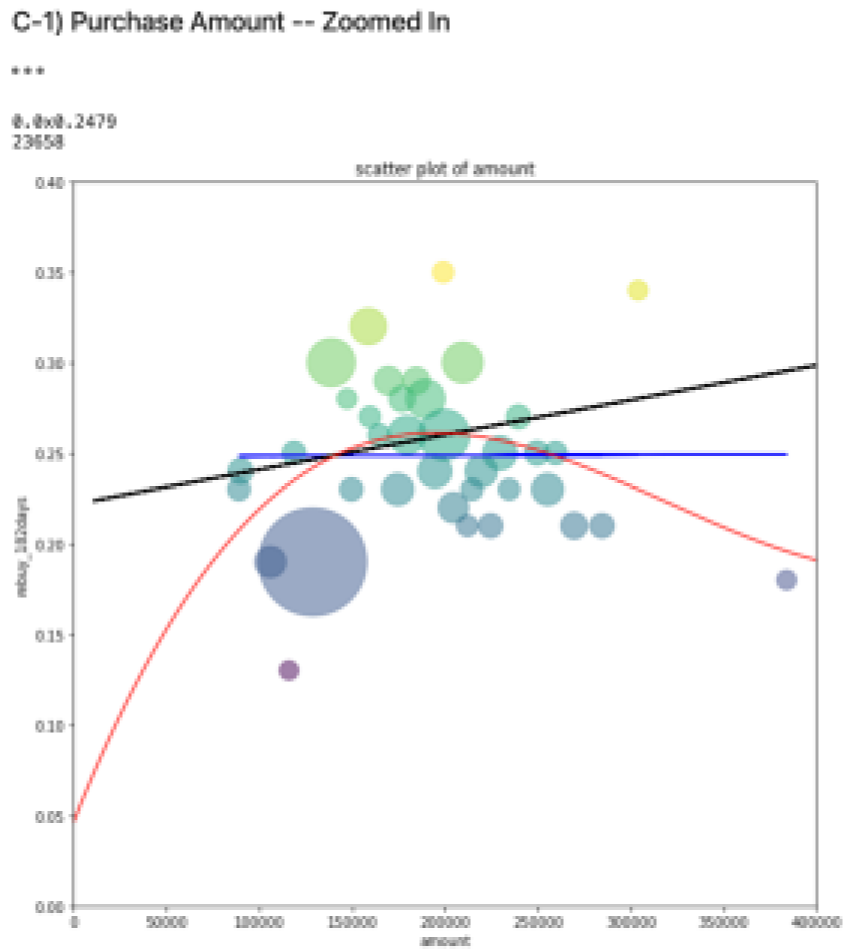
C-1) Purchase Amount -- Zoomed Out

...

```
0.0x0.2949
```



- At first glance, looking at the entire data (with the same reference point as seen in other scatter plots), it can be seen that the higher the class price, the higher the repurchase rate slightly, which is a counter-intuitive finding
    - → Outlier cases (e.g. 1M KRW+ expensive classes) are expected to play a major role (must be checked)
- However, if you zoom in and look at it again, the picture is different as shown below.
    - Zoom In & User Count based on a class of at least 200 people

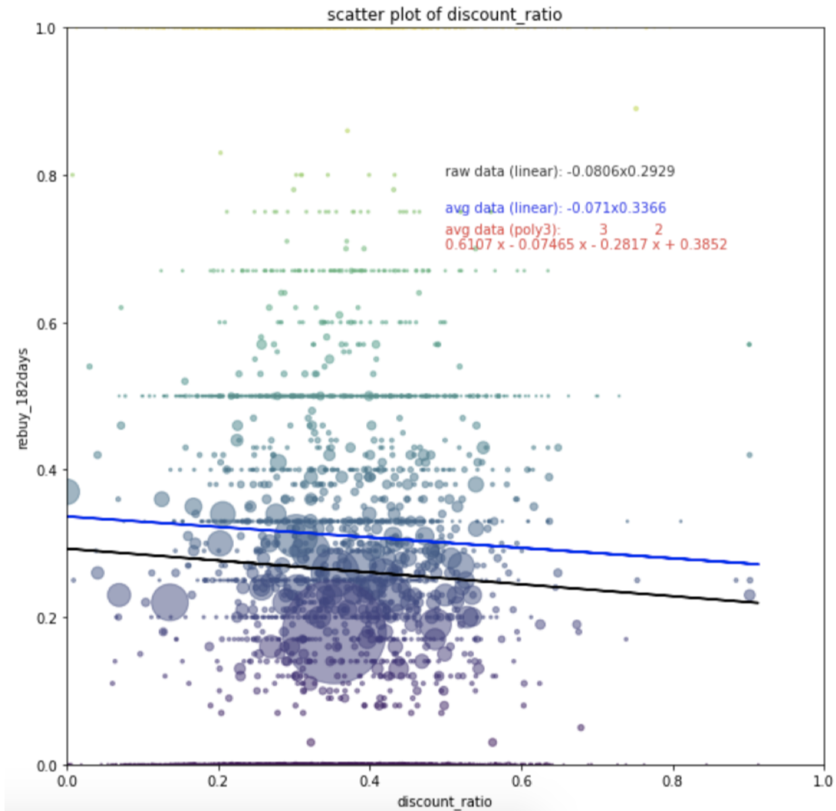


- There seems to be no general trend (blue line)
- If you add a little more curve, the repurchase rate increases as the price rises up to about 200,000 won, but after that, it decreases again.
- It would be interesting to dig a little more in terms of price optimization

C-2) Discount Amount Ratio

•••

```
-0.071x0.3366
```



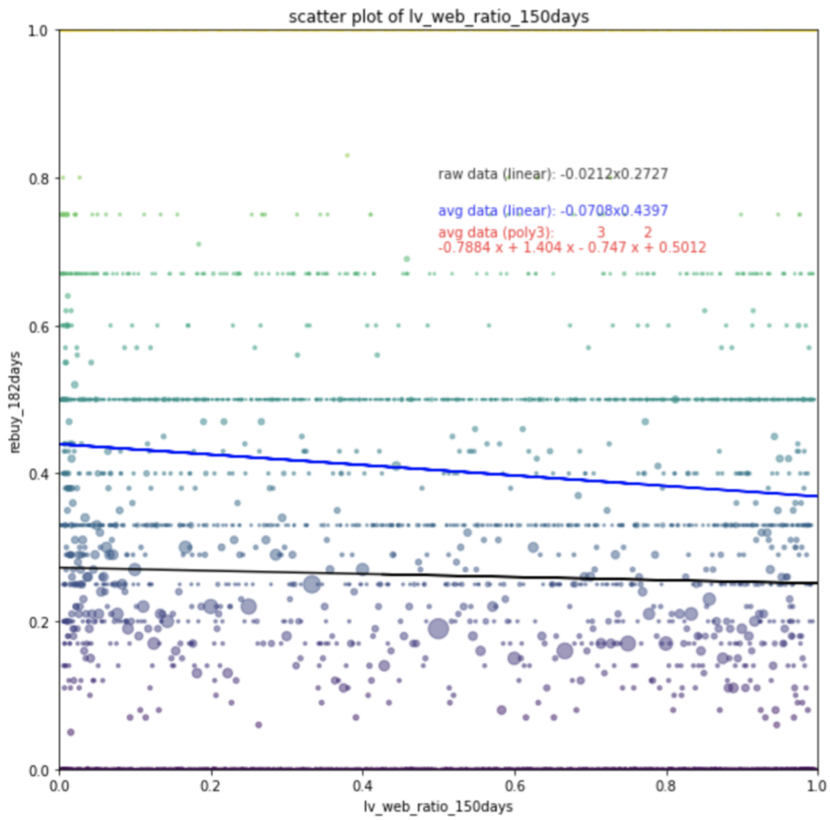
- The higher the discount rate, the lower the repurchase rate.
  - → Since the discount format is not a ratio to the amount, but a coupon with a fixed discount amount, additional analysis is required by dividing the discount amount into sections rather than the ratio.

▼ **<D Device Related Data>**

D) Web Lecture View Ratio Within 150 days -- Without 0 and 1

•••

```
-0.0708x0.4397
```



- App 수강 비중이 높은 유저일수록 재구매율이 높음
  - → App 수강 비중이 높은 유저는 App & Web 두 곳에 클래스101 서비스를 보유하고 있을 확률이 높음. 즉, App 수강 비중이 높을수록 클원의 서비스와 높은 engagement 를 가진 유저일 가능성이 높음.
  - → 위 그래프에서는 제외 되었지만 "App만" 수강한 유저들이 "Web만" 수강한 유저들 보다 재구매율이 높음
- The higher the percentage of users taking the app, the higher the repurchase rate.
  - → Users with a high proportion of App usage (for taking lectures) are more likely to have Class 101 services in both App & Web. In other words, the higher the proportion of app attendance, the higher the probability of being a user with high engagement with our service.

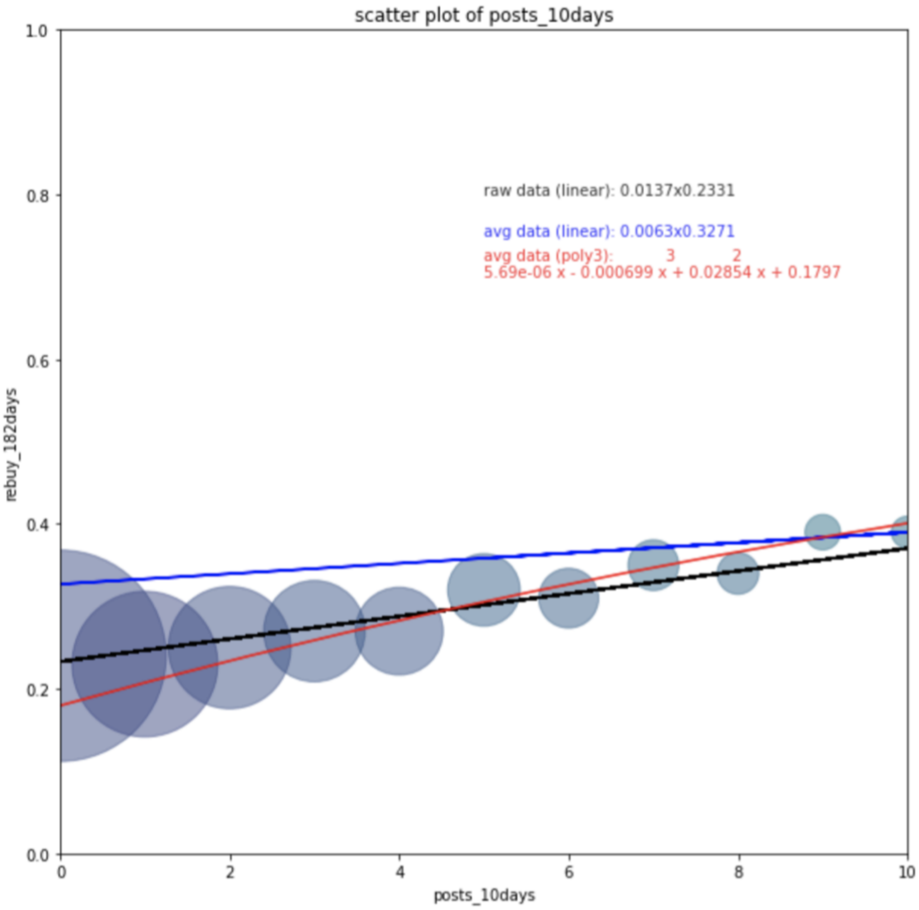
- → Although excluded from the graph above, users who took “App only” had a higher repurchase rate than users who took “Web only”

▼ **<E Other Engagement Related Data>**

E-1) Number of Posts Within 10 Days

•••

0.0063x0.3271

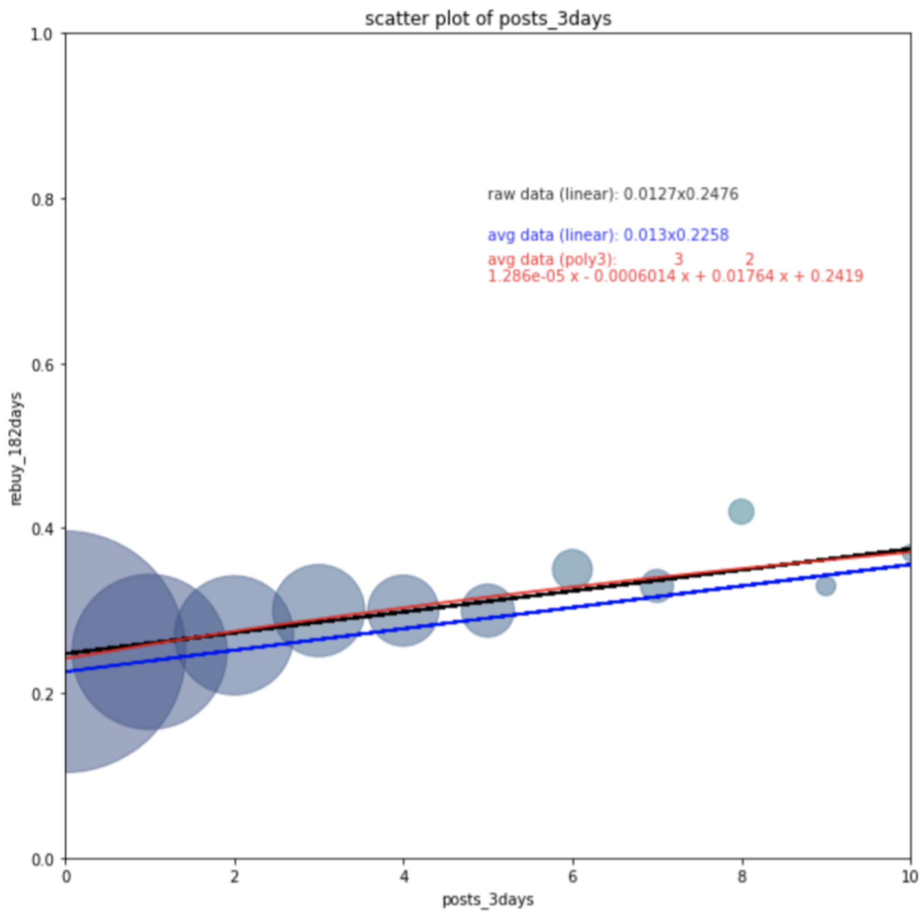


- The higher the number of comments within 10 days, the higher the repurchase rate.

E-2) Number of Posts Within 3 Days

...

0.013x0.2258

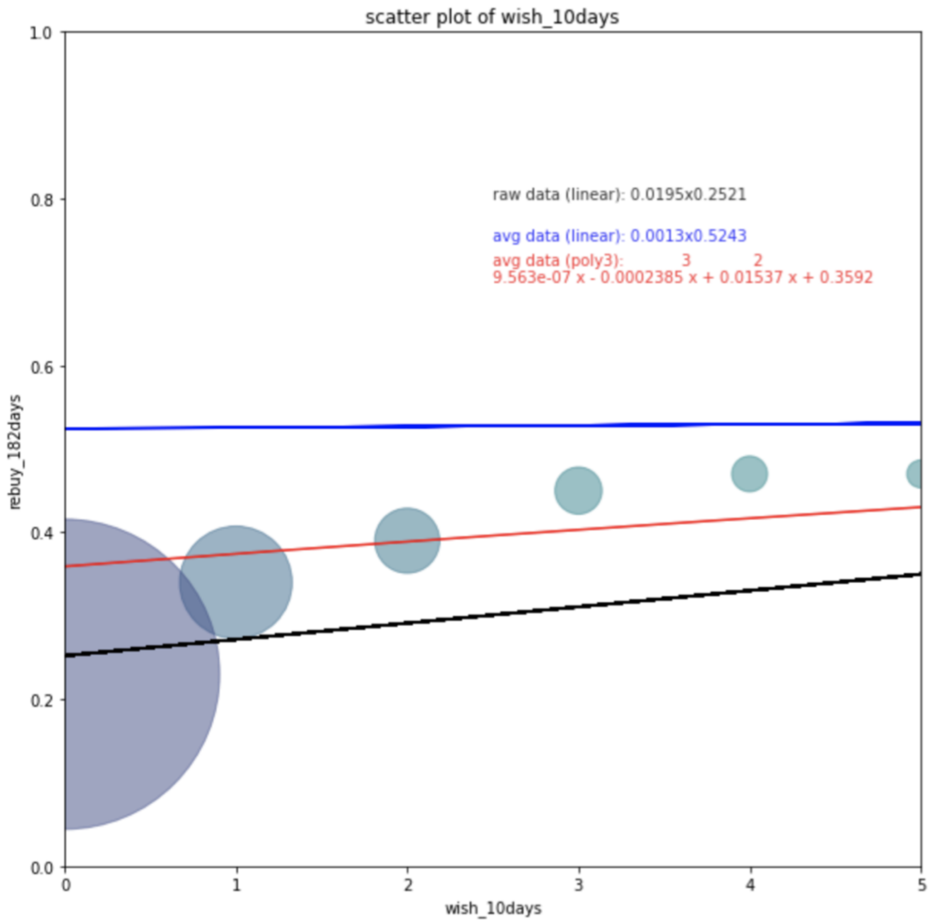


- The data on the number of comments for 3 days after purchase also show a similar trend to the E-1 graph

E-3) Number of Wishlisted Within 10 Days

...

```
0.0013x0.5243
```

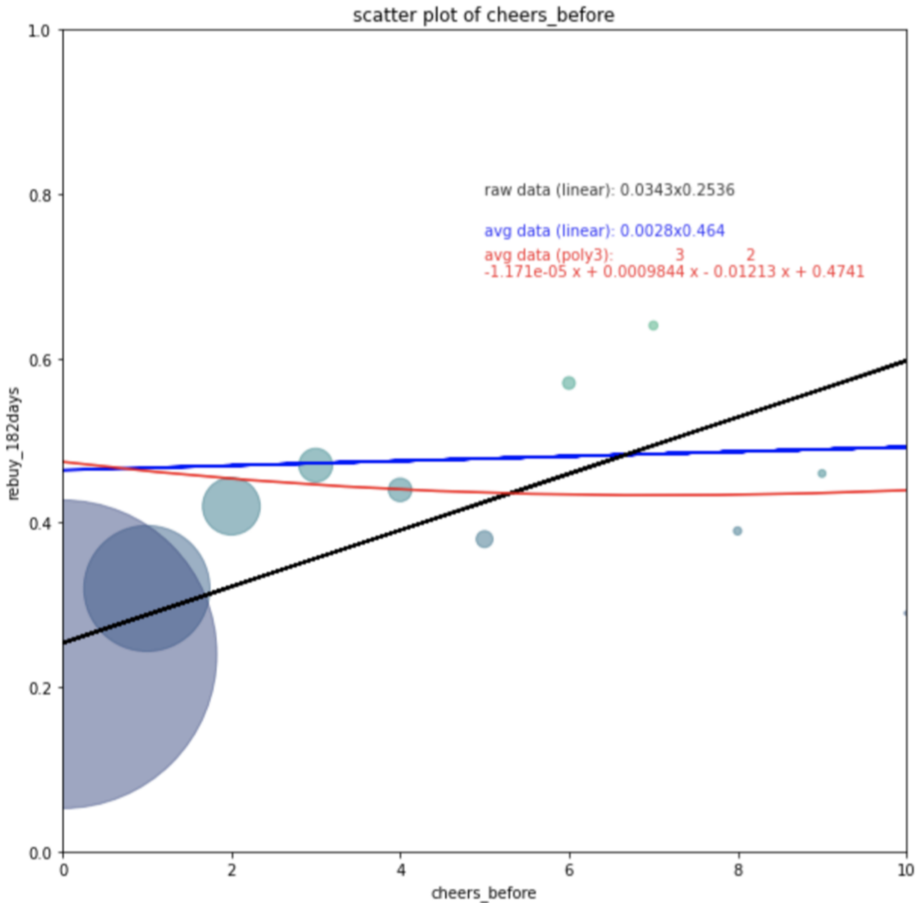


- After the first purchase of the class, the repurchase rate rises sharply each time the user wish-lists an item

E-4) Number of Cheers Before First Buy

•••

```
0.0028x0.464
```

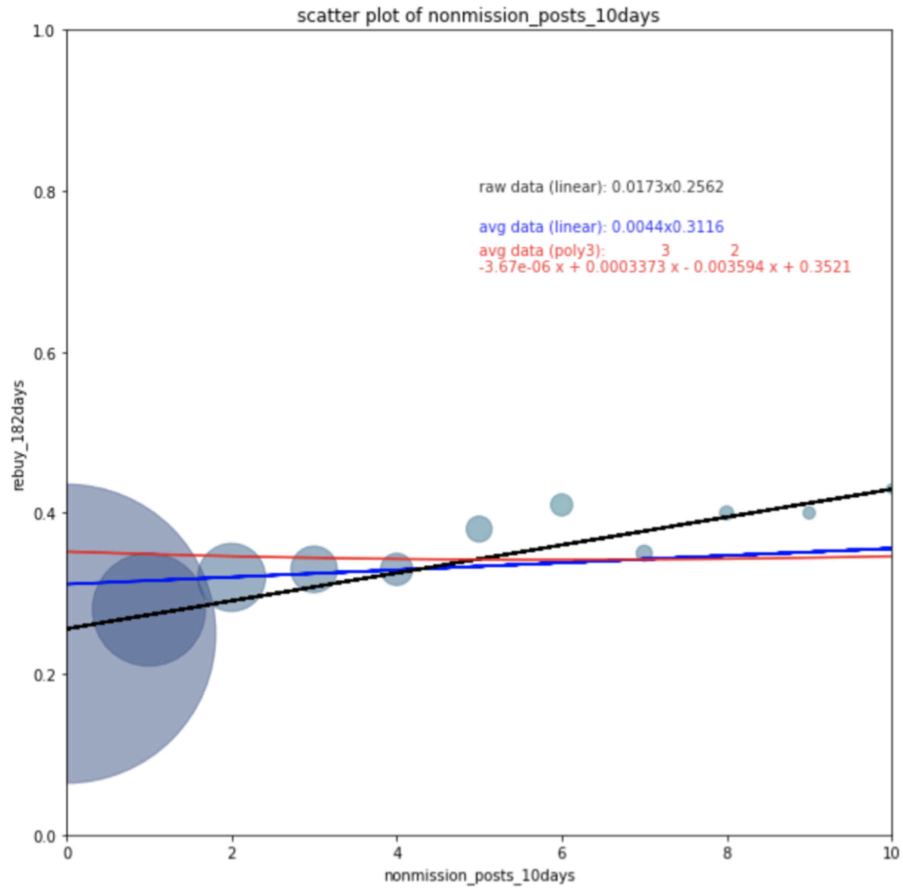


- The higher the number of "classes cheered" before the first purchase of the class, the higher the repurchase rate

E-5) Number of Non-Mission Posts Within 10 Days

•••

0.0044x0.3116



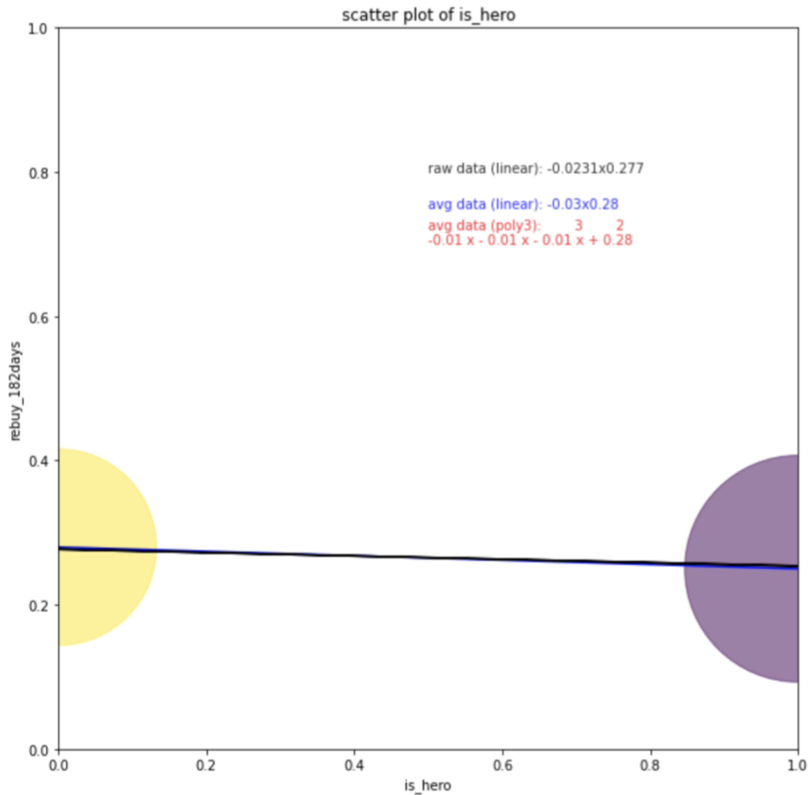
- Data on the number of “non-mission comments” for 10 days after the first purchase shows a similar trend to the E-1 graph.

▼ **<F Other Data>**

F-1) Hero Class?

•••

```
-0.03x0.28
/home/ubuntu/miniconda3/envs/py37/lib/python3.7/site-packages/ipykernel_launche
may be poorly conditioned
```

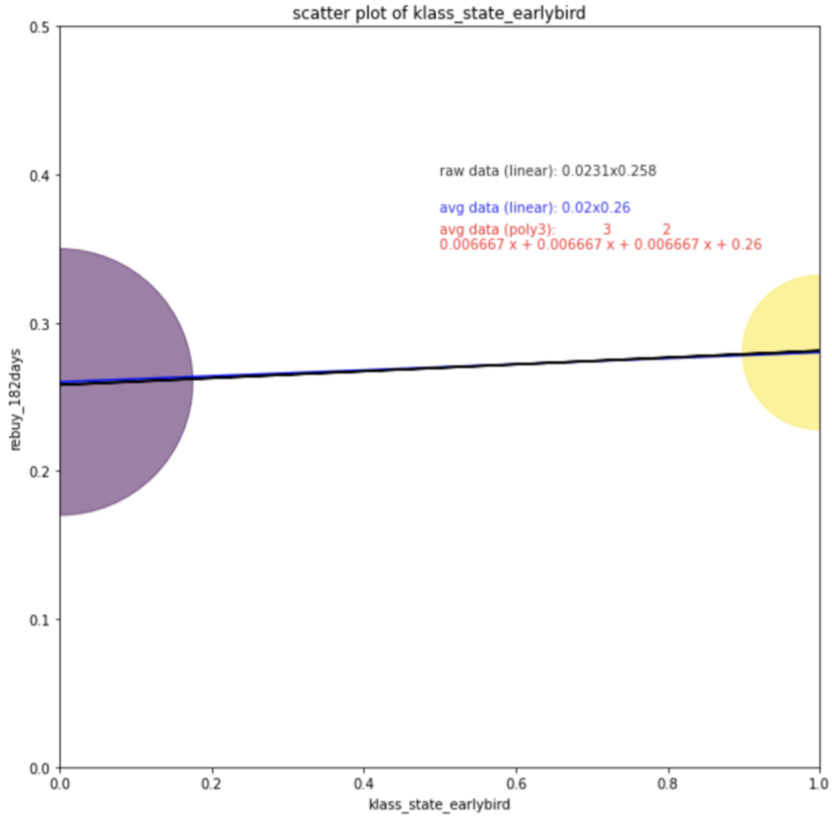


- Whether or not the first purchase class was a hero class (High-selling class, defined internally) did not have a great effect on the repurchase, but we added it because it would be good to refer to.
- Non-hero classes had a slightly higher repurchase rate

F-2) Klass State Earlybird?

•••

```
0.02x0.26
/home/ubuntu/miniconda3/envs/py37/lib/python3.7/site-packages/ipykernel_launche
may be poorly conditioned
```



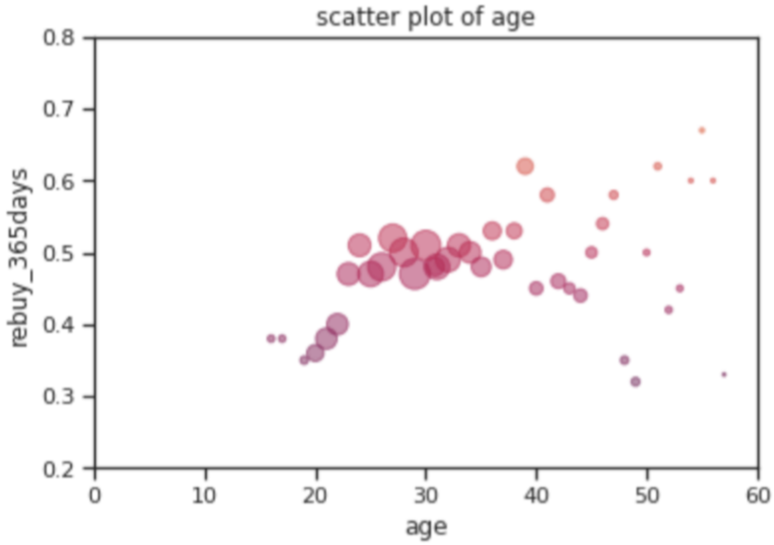
- Whether or not the first purchase class was bought as an early bird did not have a great effect on repurchasing, but I added it because it would be good to refer to.
- Classes purchased in early bird had a slightly higher repurchase rate

6h) By Categories

...

	internal_category	sum_revenue_365days	count_users	avg
0	career	63771813	47	1356847.09
1	founded	1138648329	1348	844694.61
2	dataAndDevelopment	157549281	249	632728.04
3	music	1052003368	1811	580896.39
4	signature	49943681	99	504481.63
5	craft	1537729022	3199	480690.54
6	stock	252614133	588	429615.87
7	cooking	324293092	787	412062.38
8	lifestyle	219265836	557	393655.00
9	art	2034645991	5211	390452.12
10	digitalDrawing	1857098400	4771	389247.20
11	sns	579971950	1506	385107.54
12	careerVideoAndDesign	596776586	1628	366570.38
13	oa	162767905	459	354614.17
14	photograph	599027076	1786	335401.50
15	workout	172852196	528	327371.58
16	writeContent	59706674	203	294121.55
17	onlineShop	1494405852	5504	271512.69

- The category of the first purchase classes did not have a very significant effect on the repurchase, but I did look at the averages (from past analysis)



- Re-analyzed age/gender data again based on 3000 user groups (from past analysis) — we could only perform on 3000 users who used KAKAO talk account as a signup method (acknowledging there is a bias on this)
- There was no significant results for gender.

Questions you are most worried about being asked

- How do the results differ significantly with correlation-based approaches? How does it give more causality?
- How would a Senior Data Scientist improve this project?