Boosting UA Performances with User LTV Predictive Models

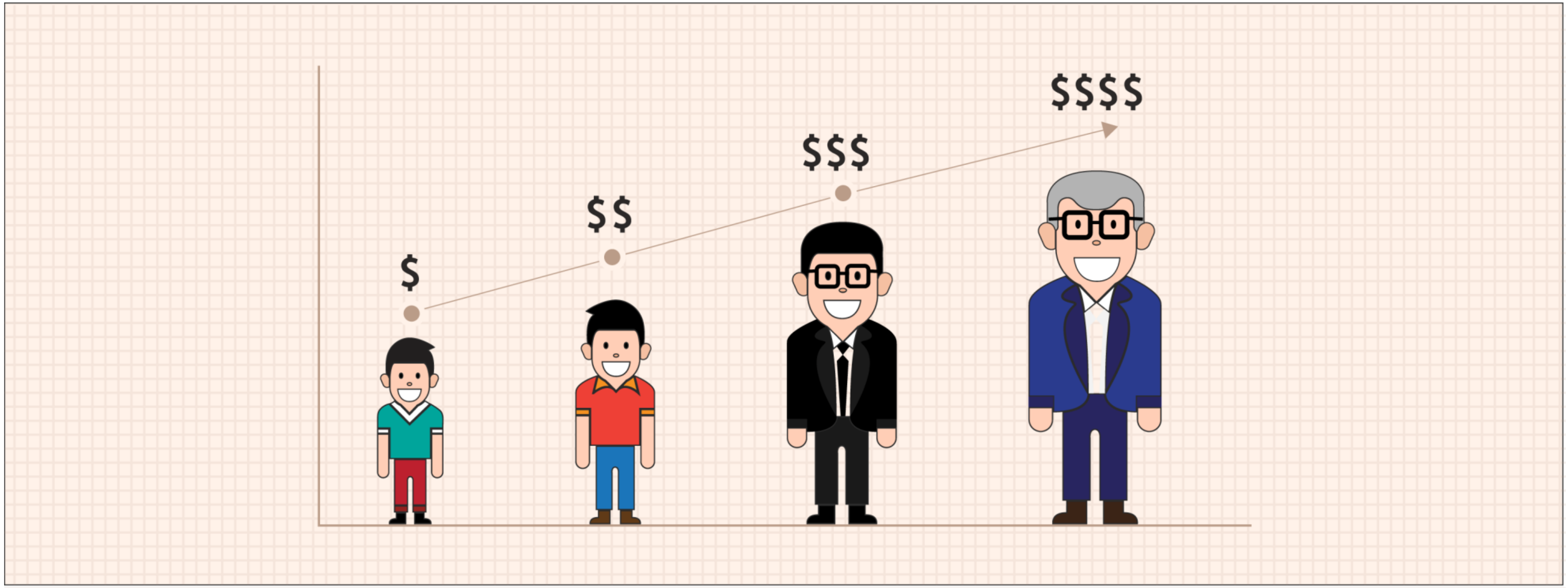
DataTalks #11
Michael Kozak, Marketing Data Scientist

Lecture Outline

- Introduction to LTV
- Initial State – ROAS 3D KPI
- User Level Based LTV
- Results
- LTV 2.0

LTV (Lifetime Value)

Definition:
Prediction of the net profit to the entire (or partly) future relationship with a customer.



General Motivation

Predicting LTV is a cross-industries problem, which, if solved, can help the company make better decisions on its customers.

- **Banking Industry**: Knowing the LTV of a client can help the bank with how much credit it can give to the client.
- **Food Industry**: Starbucks made an LTV model that showed them that if they invest more than $14K to acquire a new customer, they are losing money.
- **Gaming Industry:** Users purchasing in the app and it is important to assess how much they will be worth in the future.

Our Motivation

The LTV model we’ve created is for the **UA** use.

- UA (User Acquisition) is a domain where the company buy users by publishing ads in different channels (FB, ads within other apps etc.).
- The UA team have to allocate it’s monthly budget in a favorable channels (=bring back their investment within **12 months**).
- For this purpose, the UA team needs to know how valuable are the users they bought in the last few days.

Challenges in Predicting LTV

- We need to know the prediction very fast.
- Predicting unreliable result can cause:
  - When being too **optimistic**: wasting the budget on a channel that doesn’t produce profitable users.
  - When being too **pessimistic:** missing opportunities to scale on good performing channels.

Few Definitions Before We Start..

Cohort: A group of users that installed the app in a given timeframe (same hour/day/month…)

Business Unit: A group of users that has a common attribution (i.e. came from a certain cohort/campaign/network/country etc.)

Recoup: regain of money invested through subsequent profits.

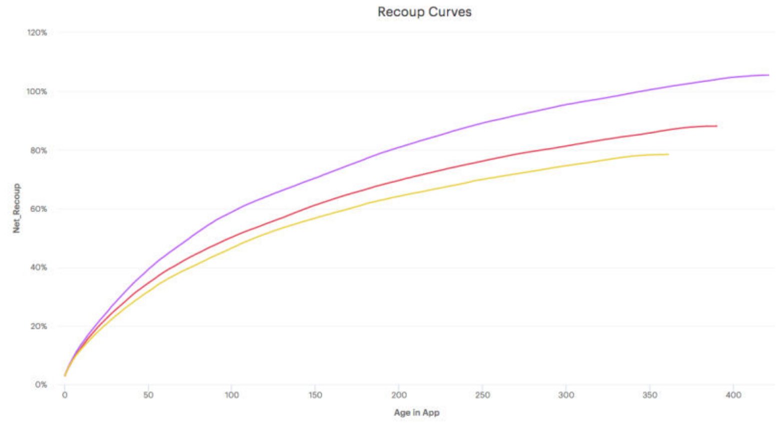
Conversion: Change status from Non-Paying to Paying user.

Initial State - 3D ROAS

- At first, we used 3D ROAS as the main KPI to decide which business units will recoup in one year since born.

$$ROAS\ 3D = \frac{sum\ of\ revenue\ in\ first\ 3\ days}{amount\ spent}$$

- The 3D ROAS was meant to



be recouped after a year.

- The 3D ROAS decision rule was:
    - If a business unit's ROAS 3D is above X% - we will predict this unit **will recoup** after 12 months, and therefore might **increase** the network's budget
    - If a business unit's ROAS 3D is below X% - we will predict this units **will not recoup** after 12 months, and

ROAS 3D KPI

Advantages	Disadvantages
Very intuitive	High variance
Many channels optimize their algorithms to ROAS based targets	Not compatible to all business units levels (campaign, sub-campaign etc.)
	Not accurate enough

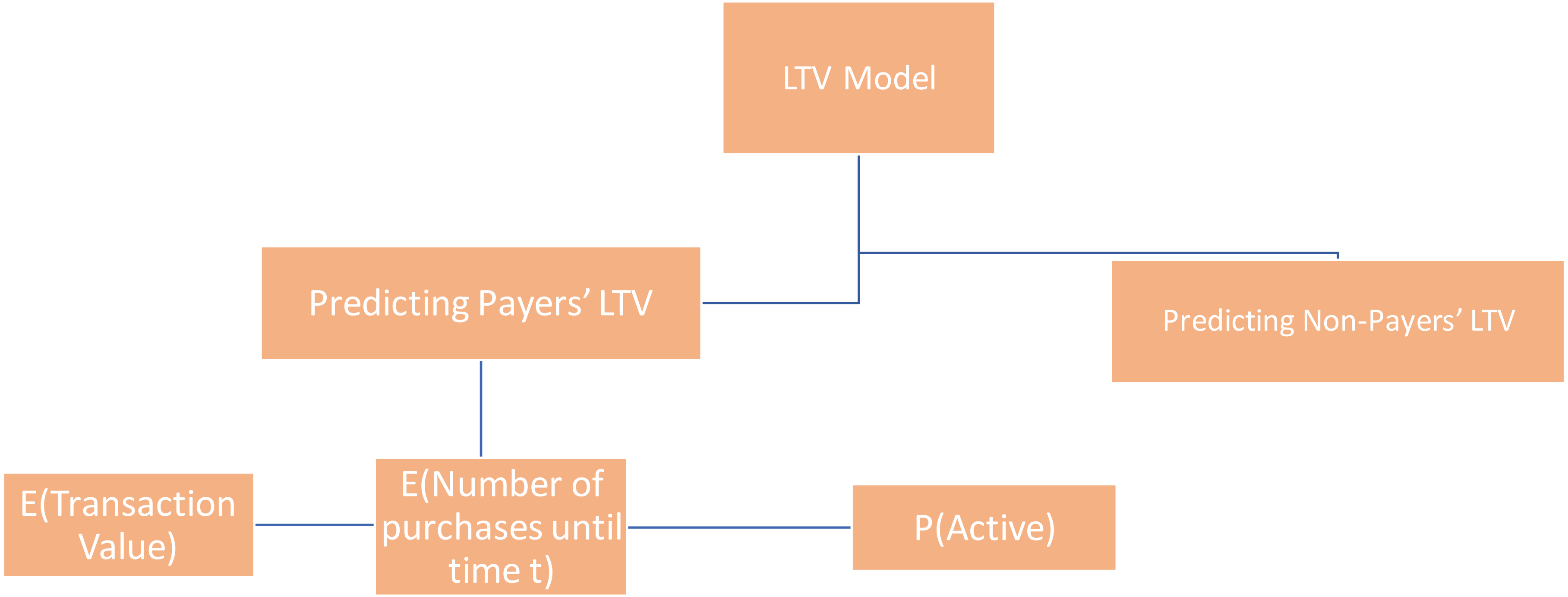
User level based LTV

The model we will review will be based on a user-level prediction.

Features:

- Ability to escalate predictions to the required business unit level.
- Making one single model and not multiple models for each business unit level.
- Extend the uses of the users predictions outside of UA needs.
- Be able to handle better small sampled business units levels, such as campaign/ sub-campaign levels.

LTV Model Components

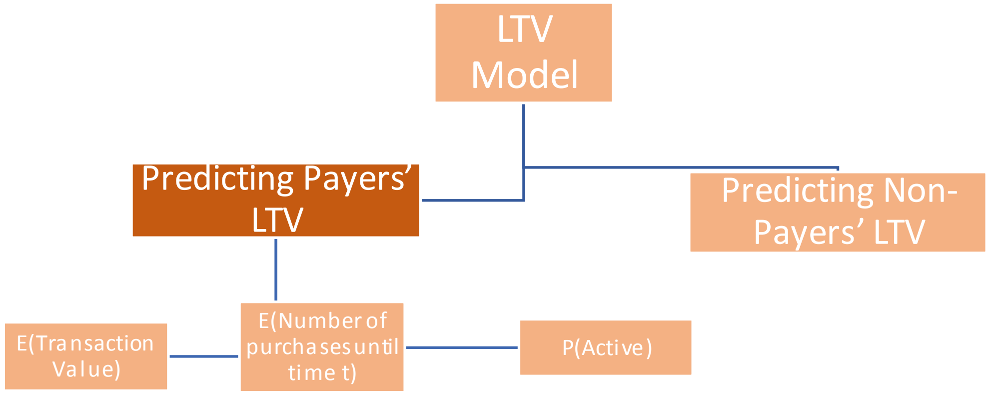


Why dividing the users to Payers/Non-Payers?

- As a FTP app, most of our users don’t pay.
- Most of our payers are generated in the first few days of their life in the app, thus we get most of the users that contribute the the business unit’s LTV relatively fast.

Age	FTDs Percent	Aggregated Percent	Aggregated Percent from revenue
1	36.2%	36.2%	44.5%
2	7.1%	43.3%	52.5%
3	4.5%	47.8%	57.5%
4	3.7%	51.5%	61.7%
5	2.9%	54.4%	64.8%
6	2.7%	57.1%	67.3%
7	2.3%	59.4%	69.4%

Predicting Payers’ LTV



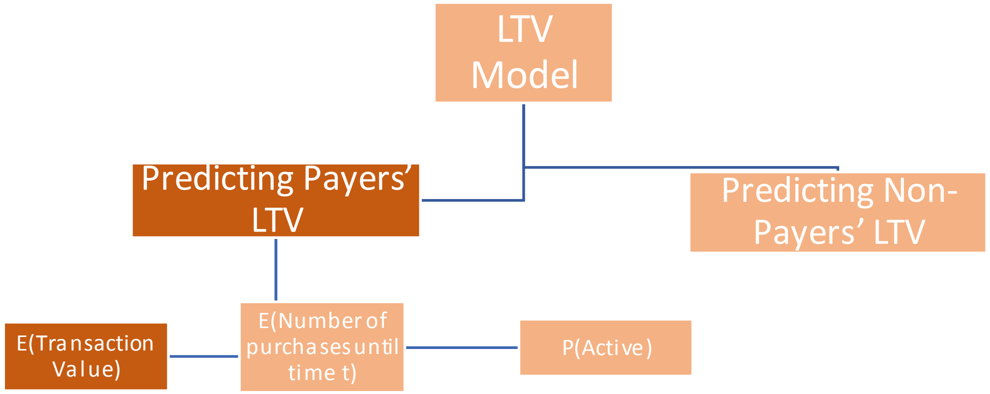
We can model the LTV by:

$$LTV(t) = \text{Current Revenue} + E(Transaction\ Value) * E(Number\ of\ Purchases\ until\ time\ t) * P(Active)$$

The predicted LTV for a payer is mainly a function of his **RFM Variables**.

Recency - The last time the user purchase.
Frequency - Number of purchases user made in the observed time frame.
Monetary - Sum of revenue the user generated in the observed time frame.

Expected Purchase Amount

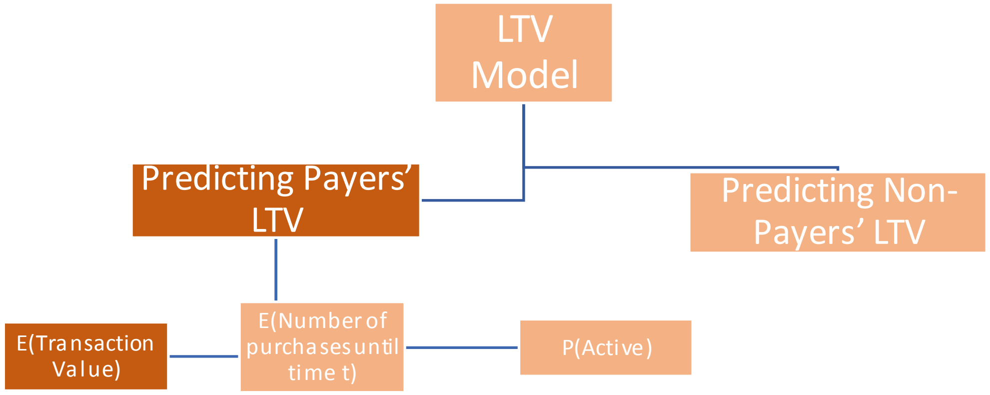


We will now formulate the first component of the model: E(Transaction Value).

Assumption: Users will purchase in an amount close to their expected value in the future.

Naïve Approach: Simple mean

Expected Purchase Amount



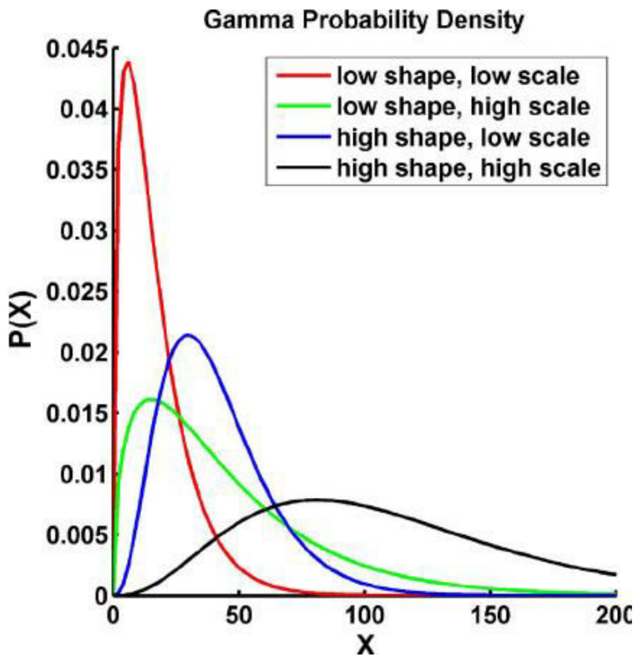
Example:

A user made a purchase with the amount of $1.99.
What is this user's purchasing amount expectation?

With one observation, it is really hard to estimate this overall expectation, but if we had some prior knowledge, we could make a more accurate estimate.

After all, many of our users don't do many purchases in the first 3 days, hence estimating their purchasing amount expectation through their current purchases can create large bias.

Expected Purchase Amount



We will turn to the Bayesian approach in this case.
Model:

Each transaction Z_i will be distributed as $gamma(p, v)$.

- Summation of k (purchases) gamma variables is distributed $gamma(pk, v)$.
- Dividing this variable by k give us the distribution of the mean, m_k, which also distributed $gamma(pk, vk)$

The assumption here is that v has a prior distribution of: $v \sim gamma(q, \gamma)$
The parameter p will be constant, since we assume that the individual-level coefficient of variation is the same for all users.

Expected Purchase Amount

After finding the posterior distribution of v, we can find the posterior expectation for M:

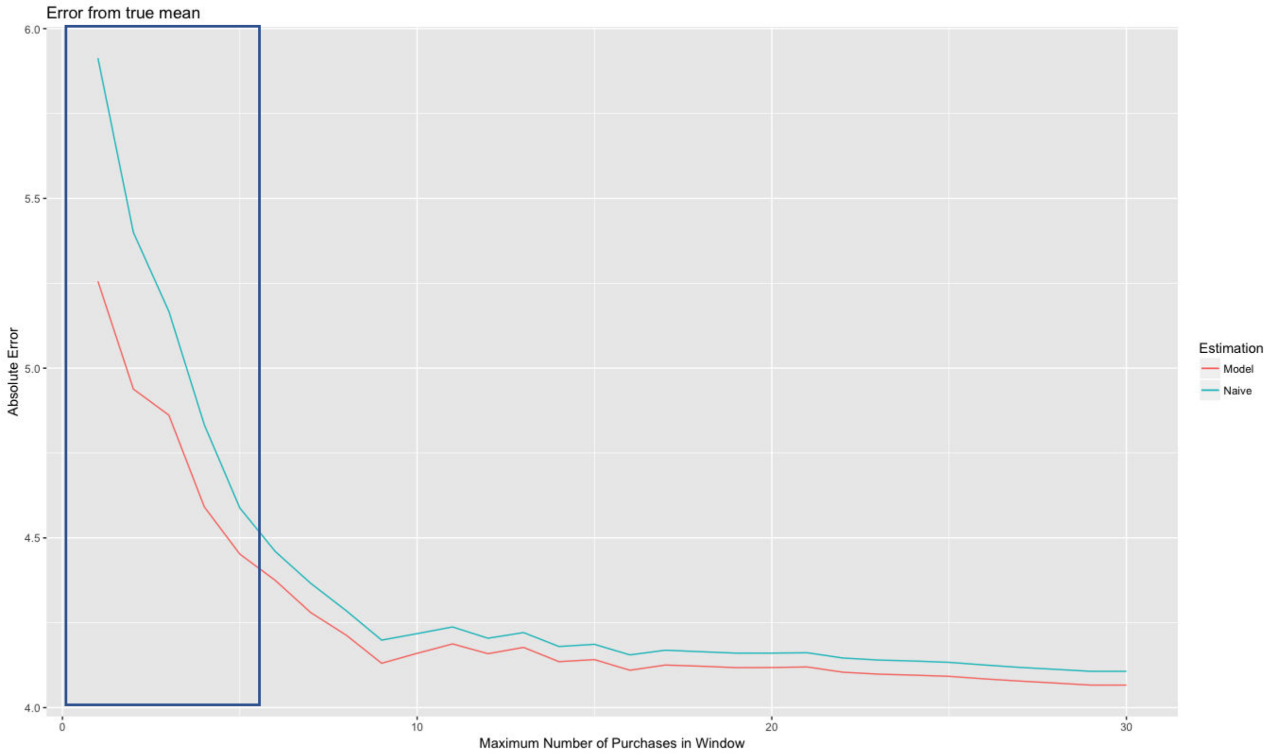
$$v|\, m_k, k, p, q, \gamma \sim gamma(pk + q, \gamma + m_k k)$$

$$E(m_k) = E\left(\frac{p}{v}\right)_v = p \, * E\left(\frac{1}{v}\right)_v = p \, * \frac{\gamma + m_k k}{pk + q \, - 1} =$$

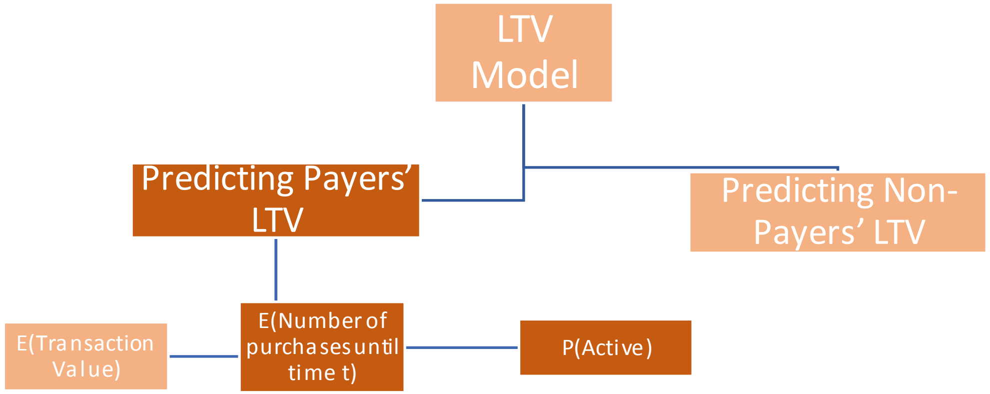
$$\underbrace{\frac{q-1}{pk+q-1}}_{w1} * \frac{\gamma p}{q-1} + \underbrace{\frac{pk}{pk+q \, -1}}_{w2} * m_k$$

Expected Purchase Amount

Results:
We will now compare the results we got from the Bayesian model to the ones we get from the naïve model presented earlier (regular mean) – estimating the expected purchase amount by simply averaging the transactions values.



Pareto/NBD Model



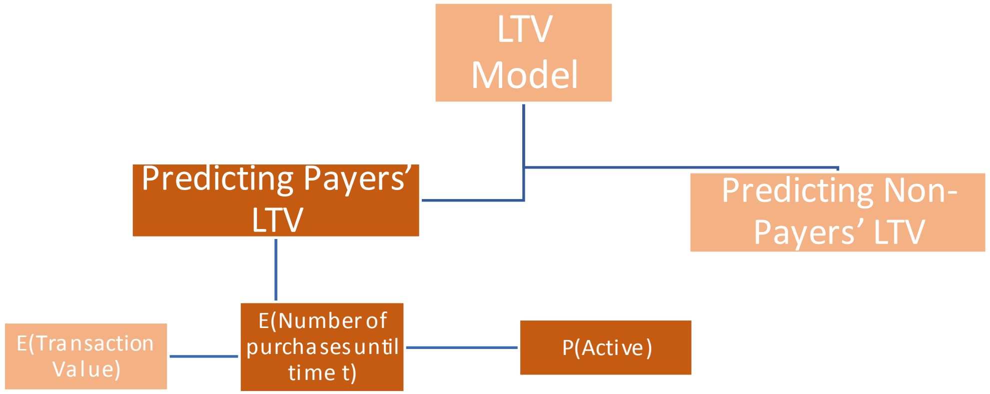
The Pareto/NBD (A.K.A "Buy until you die") model is used to predict the future activity of users.

This model uses order history as the primary input, and in particular takes into account the **frequency** and **recency** of orders.

It simulates two events:
1. It uses a "coin" to determine whether a customer churns (Pareto).
2. Then it uses "dice" to determine how many items a customer will order (NBD).

The more information you have on a customer, the better the models can fit them to a specific distribution and the more accurate the predictions end up being.

Pareto/NBD Model



The model assume each user has a prior dropout rate ($\boldsymbol{\mu} \sim Gamma(s, \beta)$) and transaction rate ($\boldsymbol{\lambda} \sim Gamma(r, \alpha)$).

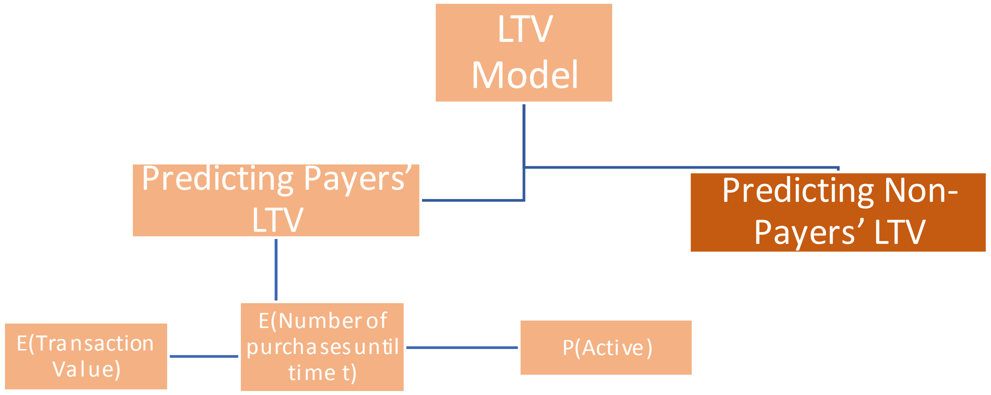
Let the random variable $Y(t)$ denote the number of purchases made in the period $(T, T+t)$.

We are interested in computing $\mathrm{E}\big(Y(t)\big|x, t_x, T\big)$, the expected number of purchases in the period $(T, T+t)$ for a customer with purchase history (x, t_x, T).

$$E[Y(t)\,|\,r,\alpha,s,\beta,x,t_x,T] = \overbrace{\left\{\frac{\Gamma(r+x)\alpha^r\beta^s}{\Gamma(r)(\alpha+T)^{r+x}(\beta+T)^s}\Big/L(r,\alpha,s,\beta\,|\,x,t_x,T)\right\}}^{P(Active|x,t_x,T)}$$

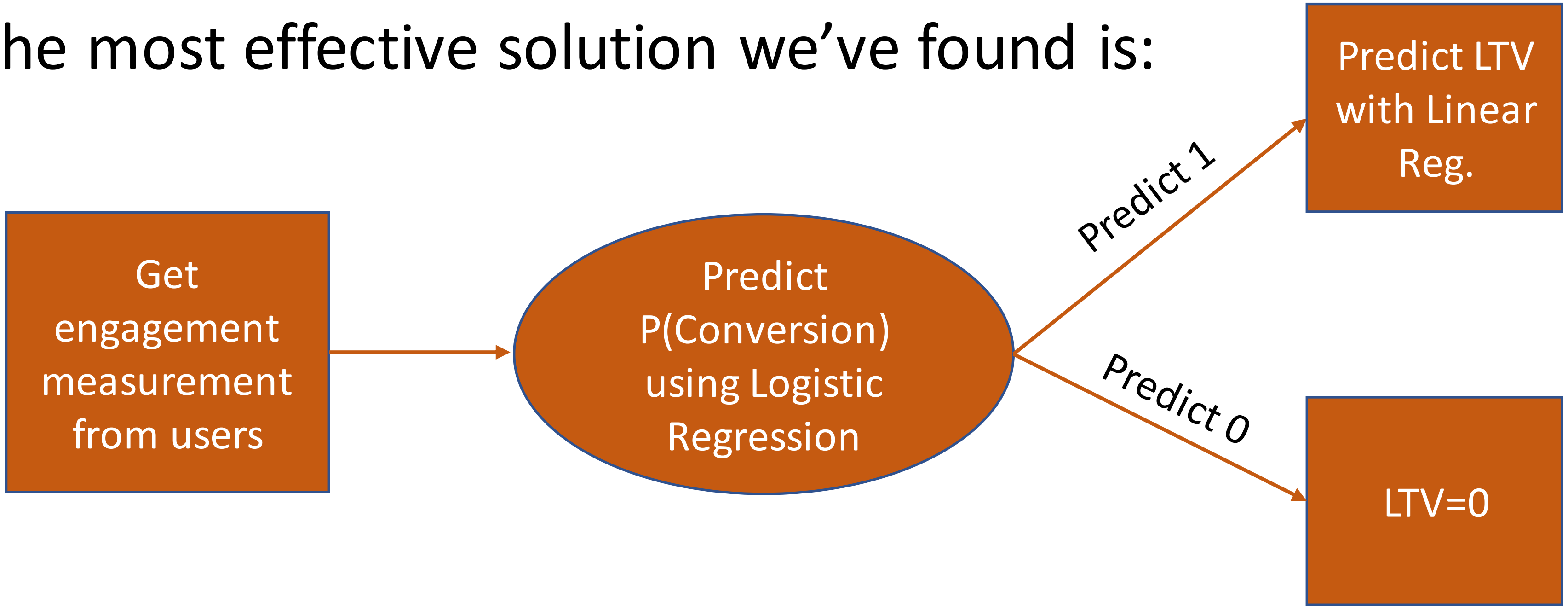
$$\times \underbrace{\frac{(r+x)(\beta+T)}{(\alpha+T)(s-1)}\left[1-\left(\frac{\beta+T}{\beta+T+t}\right)^{s-1}\right]}_{\text{Mean of Pareto/NBD Model}}.$$

Predict Non-Payers LTV



- No RFM Variables
- Low payer rate – under 5% conversion rate causes **imbalanced data classification problem**.

The most effective solution we've found is:

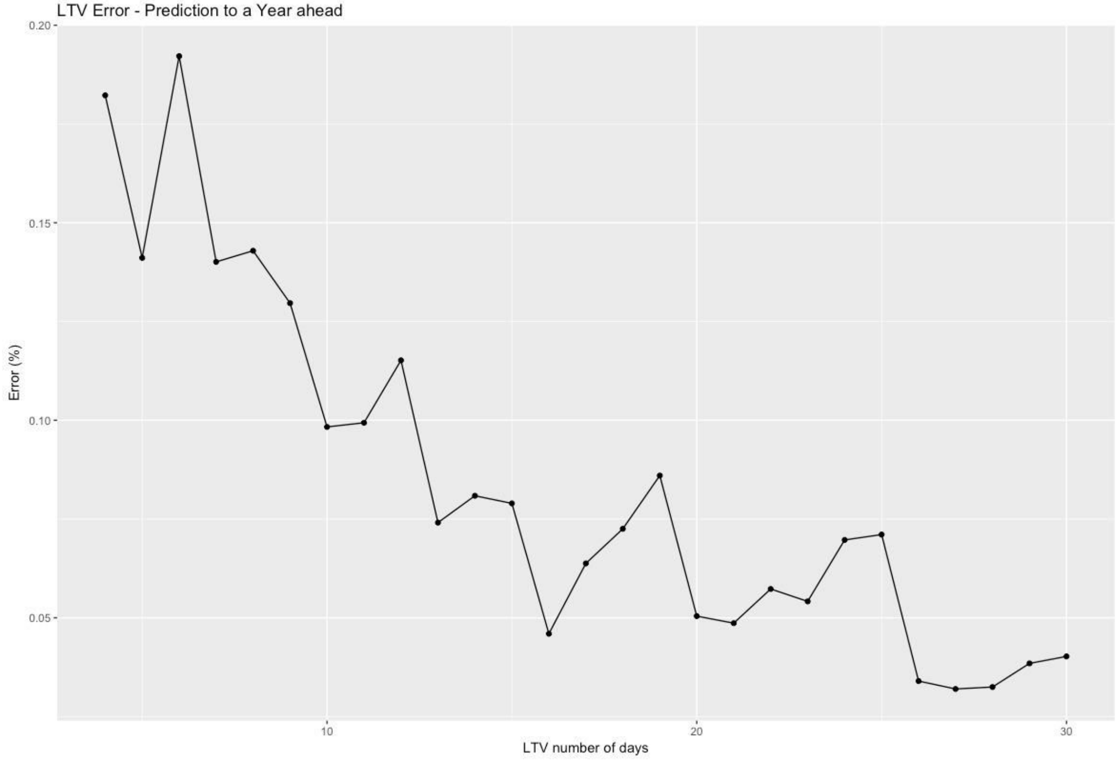


Results of Full Model

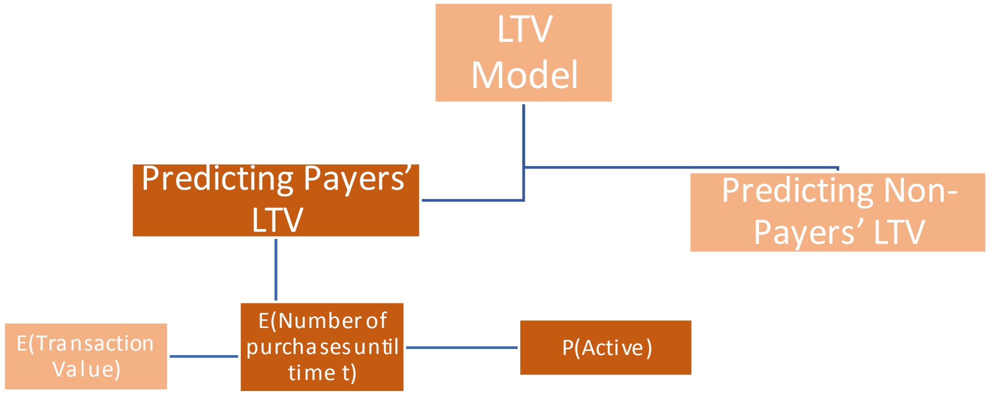
We will measure the error of the model by this term:

$$\frac{sum(Predicted\ LTV) - sum(Actual\ LTV)}{sum(Actual\ LTV)}$$

As we can see, the models tends to over-estimate new cohorts, but as time goes by gets more accurate, resulting **under 5%** error after 25 days of life time.



LTV 2.0



As was mentioned on the recent slides, the model tends to over-predict on the user level, when looking at new cohorts.

To address this problem, we implemented improvements on the second and third part of the model.

Now, outside of the RFM variables, we are taking engagement features as well:

User age
Number of sessions (last 10 days)
Number of levels gained (last 10 days)
Avg. session length (last 10 days)
Num. days entered (total)
Num. days entered (last 10 days)

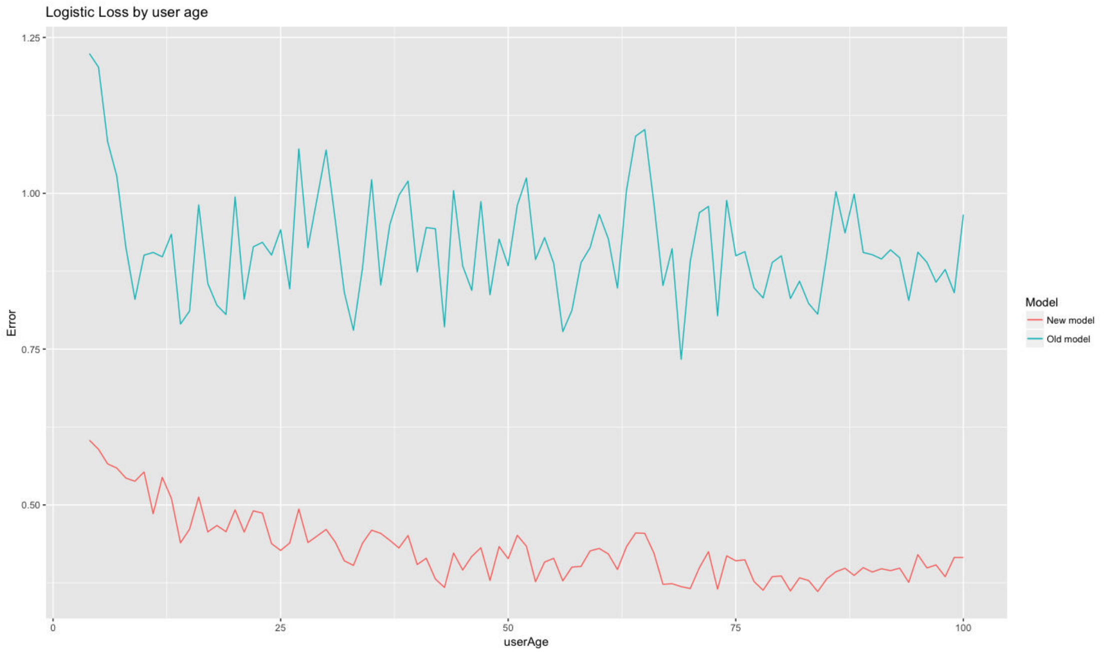
Last time seen
Last time purchased
Num. purchases (last 3 days)
Num. purchases (last 7 days)
Num. purchases (last 14 days)
Window monetary

P(Active) – Predict probability to purchase in the future through **Neural Nets.**

E(Number of purchases until time t) – Predict expected future purchases through **Random Forest**.

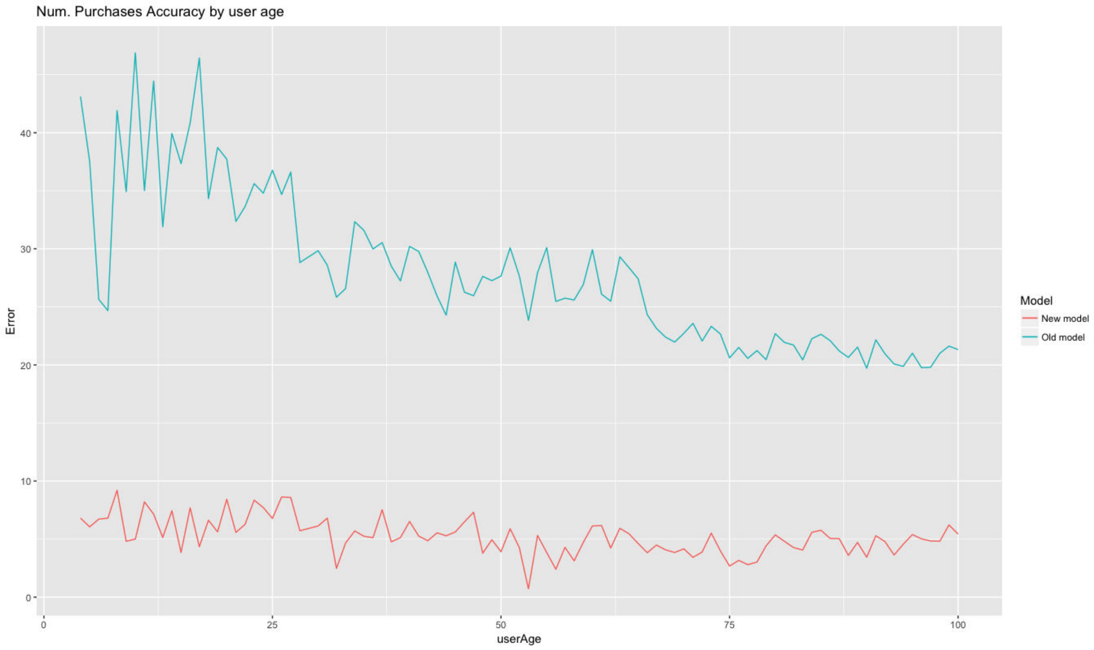
Improvement Results

P(Active)



$$\text{Error} = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(\widehat{y}_i) + (1-y_i)\log(1-\ \widehat{y}_i)\right)$$

E(Number of purchases until time t)

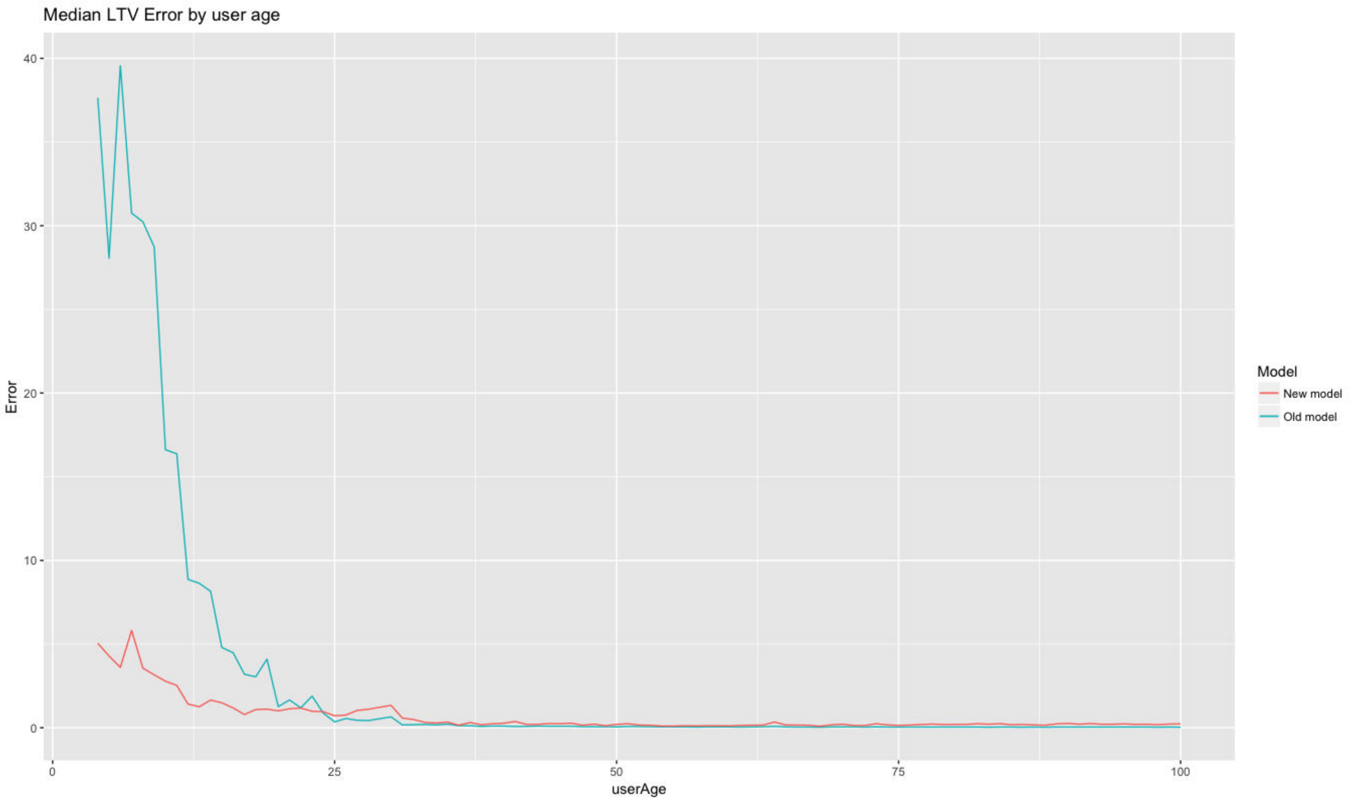


$$\text{Error} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)$$

Improvement Results

Entire model

Modeling the payers' LTV with the new components and comparing in the the initial model, we see a significant improvement in the results for new users, as the median error has reduced more than **6 times.**



Improvement Results

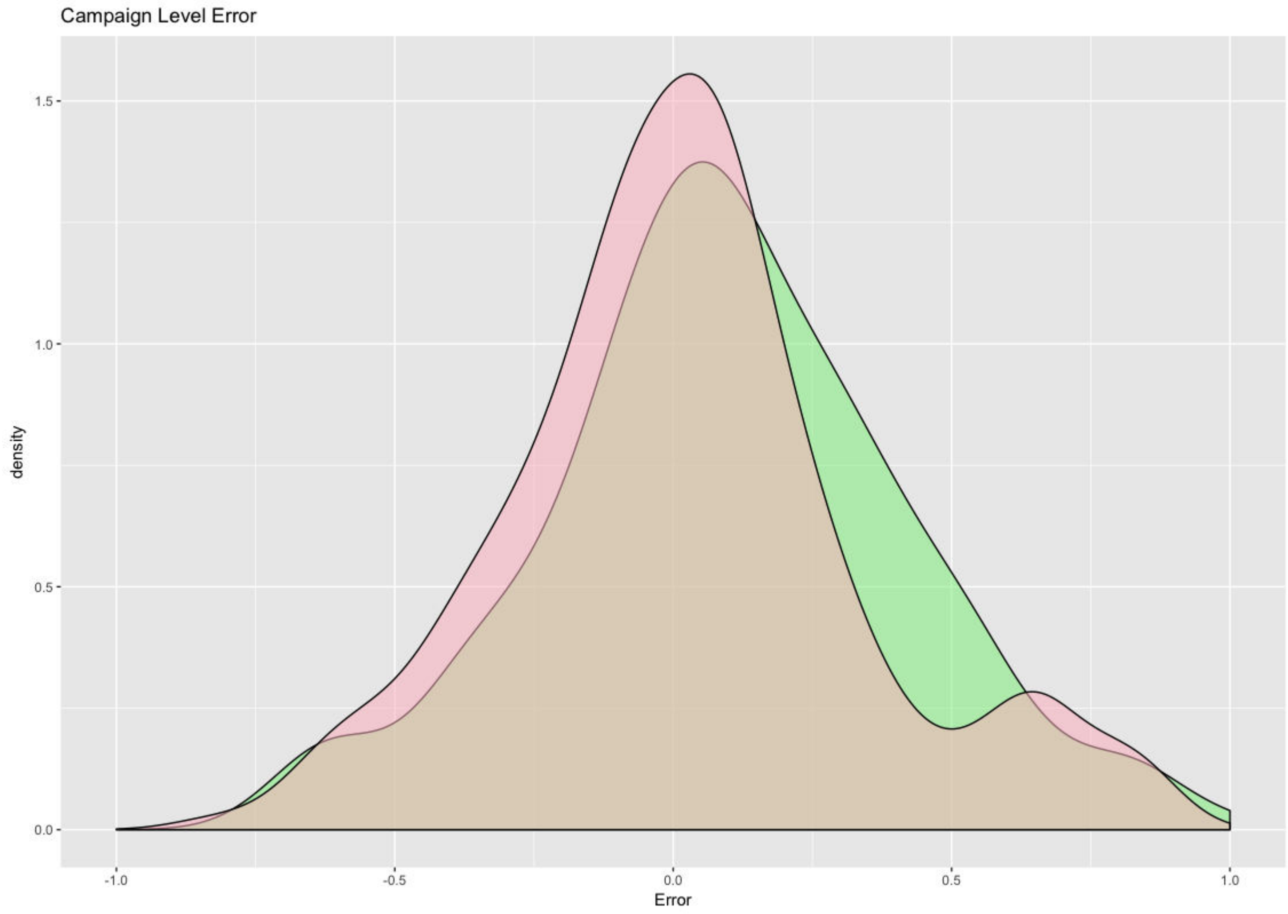
There was also an improvement in the campaign level prediction:

New Error Mean: 3%

New Error SD: **35%**

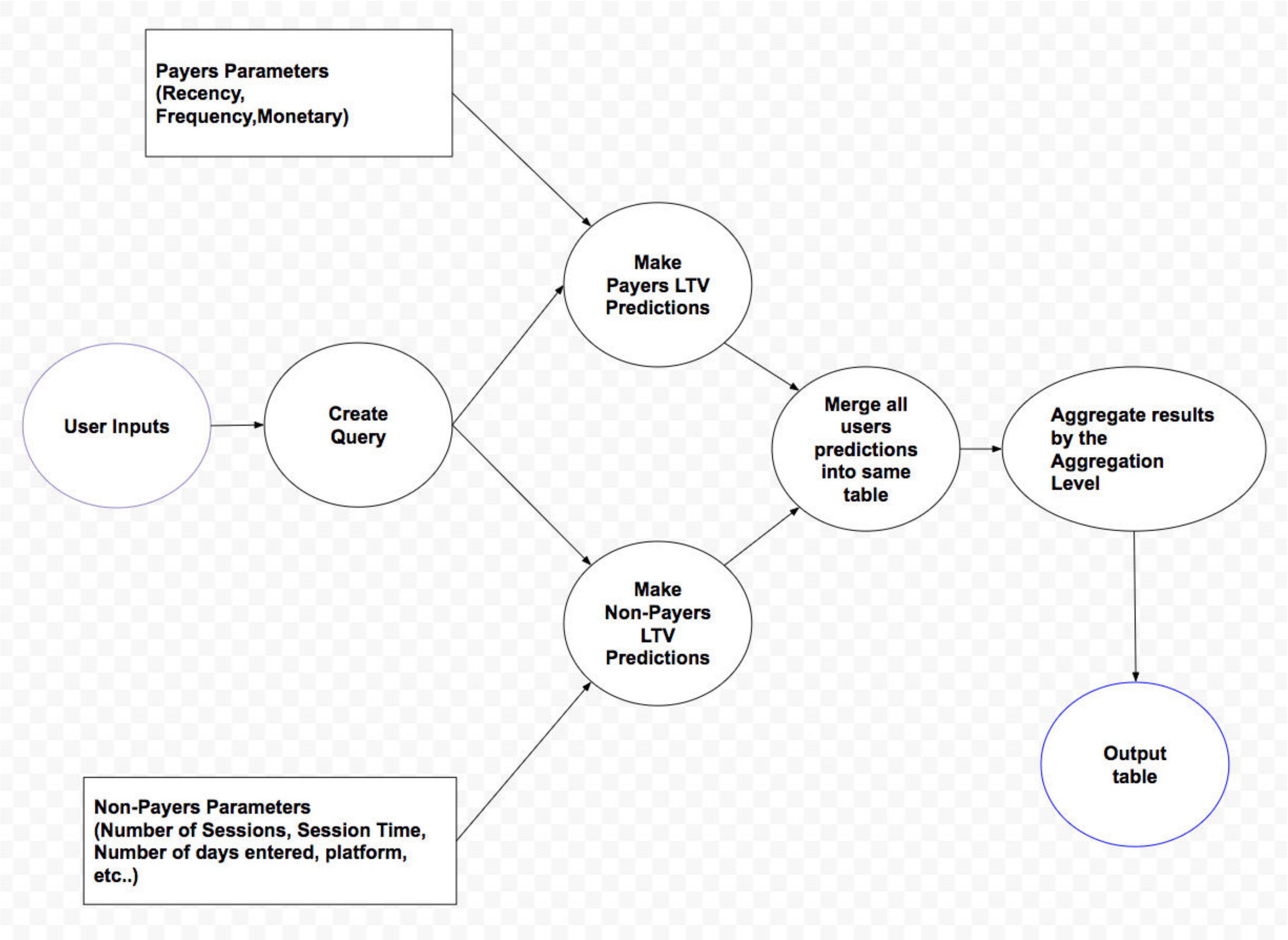
Old Error Mean: 20%

Old Error SD: **67%**



LTV Model - Building Blocks

We can look at the LTV model creation through this flow:



Summary

1. We use LTV models in order to know our users/customers better, and to help us make **better decisions**.

2. In the past, we used **ROAS 3D** metric in order to predict the quality of our users in the future.

3. Due to low performance, we changed our approach to **user level LTV**, which was based on the **RFM variables**.

4. The new model performed much better than the ROAS 3D model did, but still we had **large error on new users segment.**

5. We then moved to a new model based on NN and Random Forest to decrease the median error in these segment **by 6 times less**.

Thank You !