# Customer Lifetime

Max Kochurov

Moscow State University

January 8, 2023

# Agenda

1. Background
   - Framework
   - The BG-NBD model
2. Custom Likelihood

# What is it?

The model rescribes customer relations, they can be the following

| | **Non-Contractual**<br>Unobserved quit | **Contractual**<br>Observed quit |
|---|---|---|
| **Continuous**<br>non-scheduled | e-commerce<br>grossery purchase | credit cards<br>SIM cards |
| **Discrete**<br>scheduled | event tickets<br>weekly magazine | netflix<br>gym membership |

# What is it?

The model rescribes customer relations, they can be the following

| | **Non-Contractual** Unobserved quit | **Contractual** Observed quit |
|---|---|---|
| **Continuous** non-scheduled | e-commerce grossery purchase | credit cards SIM cards |
| **Discrete** scheduled | event tickets weekly magazine | netflix gym membership |

# Conceptual framework

Rock solid facts about the model:

But we need some simplifying assumptions:

# Conceptual framework

Rock solid facts about the model:

- Users quit relations without notice
- Users interact with unspecified and unobserved frequency

But we need some simplifying assumptions:

# Conceptual framework

Rock solid facts about the model:

- Users quit relations without notice
- Users interact with unspecified and unobserved frequency

But we need some simplifying assumptions:

- Once quit, users never return
- Unobserved frequency is constant in time

# Conceptual framework

Rock solid facts about the model:

- Users quit relations without notice
- Users interact with unspecified and unobserved frequency

But we need some simplifying assumptions:

- Once quit, users never return
- Unobserved frequency is constant in time

**The challenge**

Hard to differentiate between customers who have indefinitely churned and those who will return in the future.

# Conceptual framework

Rock solid facts about the model:

- Users quit relations without notice
- Users interact with unspecified and unobserved frequency

But we need some simplifying assumptions:

- Once quit, users never return
- Unobserved frequency is constant in time

**The challenge**

Hard to differentiate between customers who have indefinitely churned and those who will return in the future.

BG-NBD model can help

# The Intuition of the BG-NBD Model

BG-NBD is Beta-Gamma Negative Binomial Distribution

- focuses on predicting the number of transactions

It is a part of the LTV model

$$\text{LTV} = \text{number of transactions} \times \text{value of transaction}$$

# The Intuition of the BG-NBD Model

BG-NBD is Beta-Gamma Negative Binomial Distribution

- focuses on predicting the number of transactions

It is a part of the LTV model

$$\text{LTV} = \text{number of transactions} \times \text{value of transaction}$$

- value of transaction is usually an easy thing to get
- we mostly care about number of transactions

# Example

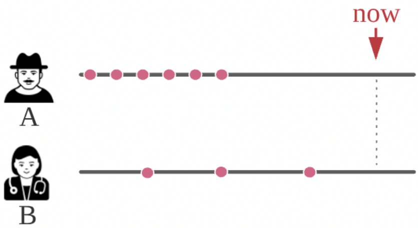
- You are a bakery owner
- You have purchase records of your customers (with id)
- You want to plan next year revenue

# Practical Aspects

In addition to standard assumptions:

- Once quit, users never return
- Unobserved frequency is constant in time

There is couple more:



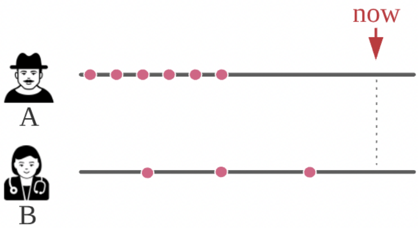
# Practical Aspects

In addition to standard assumptions:

- Once quit, users never return
- Unobserved frequency is constant in time

There is couple more:

- Frequency is user specific



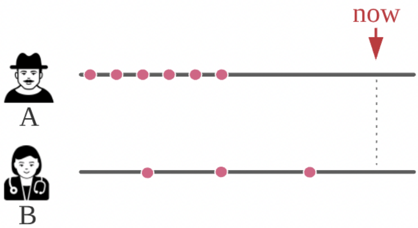
# Practical Aspects

In addition to standard assumptions:

- Once quit, users never return
- Unobserved frequency is constant in time

There is couple more:

- Frequency is user specific
- Quit probability is also user specific



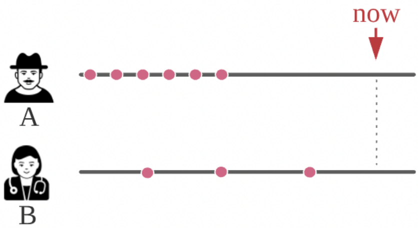
# Practical Aspects

In addition to standard assumptions:

- Once quit, users never return
- Unobserved frequency is constant in time

There is couple more:

- Frequency is user specific
- Quit probability is also user specific




**Interpretation**

So we treat each user separately which is more realistic.
The retention follows the same idea.

# How priors help

There are number of assumptions we may want to bring in:

- We're sure that most of our customers make purchases at a rate of 4 purchases a week
- Our users are "addicted" to the product and have low quit probability, below 30% in a year

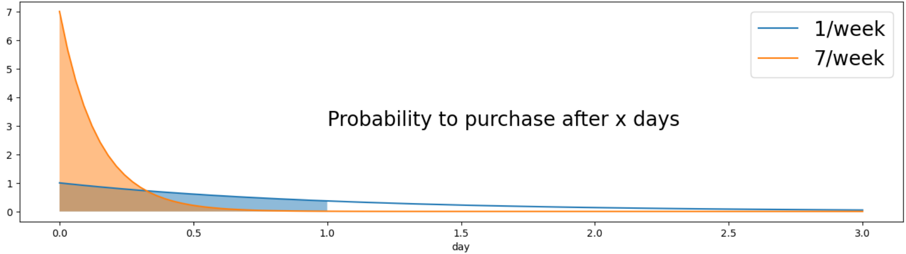
**Uncertainty**

Our assumptions may be not very certain and we'll reflect them in the priors

# The Poisson Process

Let's break complicated likelihood in parts.

- Users follow Poisson Process to make (discrete) purchases
- They make purchases at rate $\lambda = 1$ and 7 per week.
- With this assumption, we can model the time-to-next-purchase $\Delta_t$ as an exponential distribution Exponential($\lambda$)




We are certain the Orange customer will make a purchase next day

# Time to purchase

Every gap can be measured with Exponential($\lambda$)

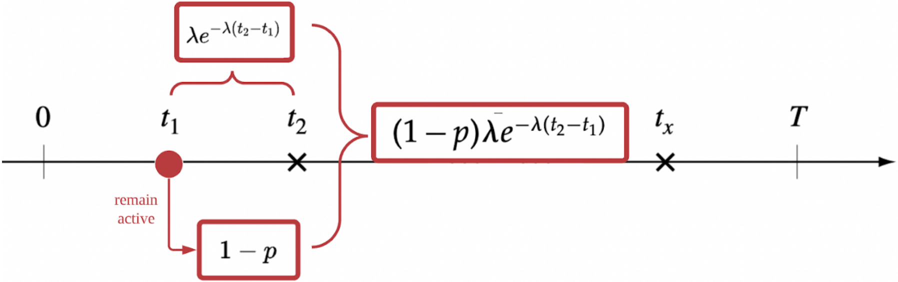

$$P(t_1, \ldots, t_x \mid \lambda) = \prod_{i=1}^{i=x} \lambda e^{-\lambda(t_i - t_{i-1})} = \lambda^x e^{-\lambda t_x} = P(x, t_x \mid \lambda)$$

## Insight

You do not need a sequence $t_1, \ldots, t_x$, you only need $t_x =$ "Age of the customer at last purchase **x**" and **x** = "Number of purchases"

# Deactivation Probability

Every step follows a change to deactivate

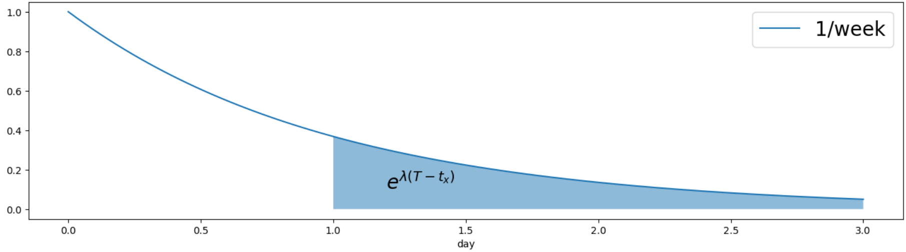



In the previous formula we add $(x-1)$ cases of non deactivation.

$$P(x, t_x \mid \lambda, p) = (1-p)^{x-1}\lambda^x e^{-\lambda t_x}$$

# Deactivation after $t_x$

1. The customer did not deactivate after $t_x$, this probability is $(1-p)e^{-\lambda(T-t_x)}$
2. Customer deactivated with probability $p$ and never returns




Remainder probability is

$$P(T \mid \lambda, t_x, p) = p + (1-p)e^{-\lambda(T-t_x)}$$

# Total Probability

Step by step we figure out the final formula to use

$$P(x, t_x, T \mid \lambda, p) = \underbrace{\left(p + (1-p)e^{-\lambda(T-t_x)}\right)}_{P(T \mid \lambda, p, t_x)} \times \underbrace{(1-p)^{x-1}\lambda^x e^{-\lambda t_x}}_{P(x, t_x \mid \lambda, p)}$$

# Total Probability

Step by step we figure out the final formula to use

$$P(x, t_x, T \mid \lambda, p) = \underbrace{\left(p + (1-p)e^{-\lambda(T-t_x)}\right)}_{P(T \mid \lambda, p, t_x)} \times \underbrace{(1-p)^{x-1}\lambda^x e^{-\lambda t_x}}_{P(x, t_x \mid \lambda, p)}$$

$$\begin{aligned} P(x, t_x, T \mid \lambda, p) &= p(1-p)^{x-1}\lambda^x e^{-\lambda t_x} \\ &\quad + (1-p)e^{-\lambda(T-t_x)}(1-p)^{x-1}\lambda^x e^{-\lambda t_x} \end{aligned}$$

# Total Probability

Step by step we figure out the final formula to use

$$P(x, t_x, T \mid \lambda, p) = \underbrace{(p + (1-p)e^{-\lambda(T-t_x)})}_{P(T \mid \lambda, p, t_x)} \times \underbrace{(1-p)^{x-1}\lambda^x e^{-\lambda t_x}}_{P(x, t_x \mid \lambda, p)}$$

$$\begin{aligned} P(x, t_x, T \mid \lambda, p) &= p(1-p)^{x-1}\lambda^x e^{-\lambda t_x} \\ &\quad + (1-p)e^{-\lambda(T-t_x)}(1-p)^{x-1}\lambda^x e^{-\lambda t_x} \\ P(x, t_x, T \mid \lambda, p) &= p(1-p)^{x-1}\lambda^x e^{-\lambda t_x} + (1-p)^x \lambda^x e^{-\lambda T} \end{aligned}$$

**Important**

This formula is only applicable to customers made at least one purchase, those have $x > 0$ and $t_x$. We need a formula for all customers including $x = 0$ and absence of $t_x$

# Total Probability for all customers

Assumption is that all users after they enroll are active by default.
Given a user without any purchases, we have

$$P(T \mid \lambda, x = 0) = e^{-\lambda T}$$

Combining it with the previous formula we have

$$P(x, t_x, T \mid \lambda, p) = \begin{cases} p(1-p)^{x-1}\lambda^x e^{-\lambda t_x} + (1-p)^x \lambda^x e^{-\lambda T} & x > 0 \\ e^{-\lambda T} & x = 0 \end{cases}$$

Or, equivalently

$$P(x, t_x, T \mid \lambda, p) = \delta_{x>0}\left[p(1-p)^{x-1}\lambda^x e^{-\lambda t_x}\right] + (1-p)^x \lambda^x e^{-\lambda T}$$

# Transferring to PyMC

There is no such a distribution in PyMC, so you need to implement one. This is what is done with `pm.DensityDist`

```
def logp_x_tx_T(value, p, lam):
    # value.shape = (n_obs, 3)
    x, tx, T = value.T[0], value.T[1], value.T[2]
    delta_x = at.where(x>0, 1, 0)
    A1 = x*at.log(1-p) + x*at.log(lam) - lam*T
    A2 = (at.log(p) + (x-1)*at.log(1-p) + x*at.log(lam) - lam*tx)
    A3 = at.log(at.exp(A1) + delta_x * at.exp(A2))
    return A3
```

```
with pm.Model() as model:
    ...
    pm.DensityDist(
        "obs", p, lam,
        logp=logp_x_tx_T,
        observed=data # (n_obs, 3)
    )
```

# References I

A. Meraldo.
Bayesian customer lifetime values modeling using pymc3, 2022.

A. Meraldo.
Customer lifetime value estimation via probabilistic modeling, 2022.