Testing KumoRFM with temporal user status

Create a synthetic dataset to test KumoRFM dealing with temporal dependency that is not in the transactional history, but instead is in a "user type" variable that changes over time.

Users of 2 types: premium/free. Premium users can do actions that free users cannot.

Testing if/how KumoRFM detects this dependency and uses during inference.

Dataset description

- Two types of users: free tier, premium tier. Users can upload files.
- For simplicity, there are 2 sizes: 5GB and 50GB.
- Premium tier users can upload both sizes.
- Free tier users can only upload files with size of 5GB.
- Users can change their tier no earlier than 24 hours after the last change.
- There are 50 users, covering 5 cohorts:
  - Always premium → a single interval that spans the whole window.
  - Always free → a single interval that spans the whole window.
  - Premium → Free once (≥24h after start).
  - Free → Premium once (≥24h).
  - Free → Premium → Free (each change ≥24h apart; all within window).
- The history of transactions lasts 10 days from March 1st to March 10th, 2025.
- The prediction task of interest is the likelihood of a user uploading a 50GB in the next k hours.
- Expectations:
  - For users that just became free tier this should be 0.
  - For users that just became Premium it should be (1 – exp(-lambda*k)), by design.

Tables

Users: 50 users, 10 in each cohort	
user_id	name

Items: 45 of 5 GB, 35 of 50 GB	
item_id [PK]	size_gb

tiers				
tier_status_id	user_id	from_date	until_date	status

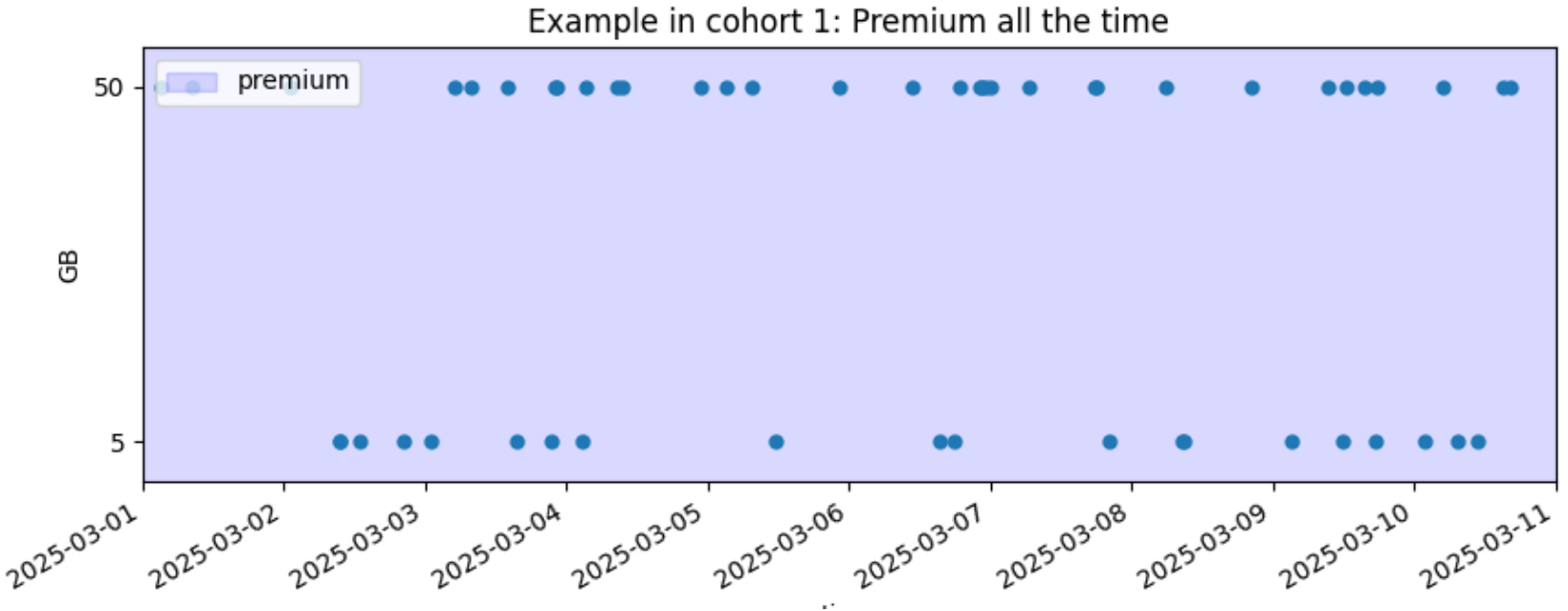
uploads			
upload_id	user_id	item_id	datetime

Dataset generation

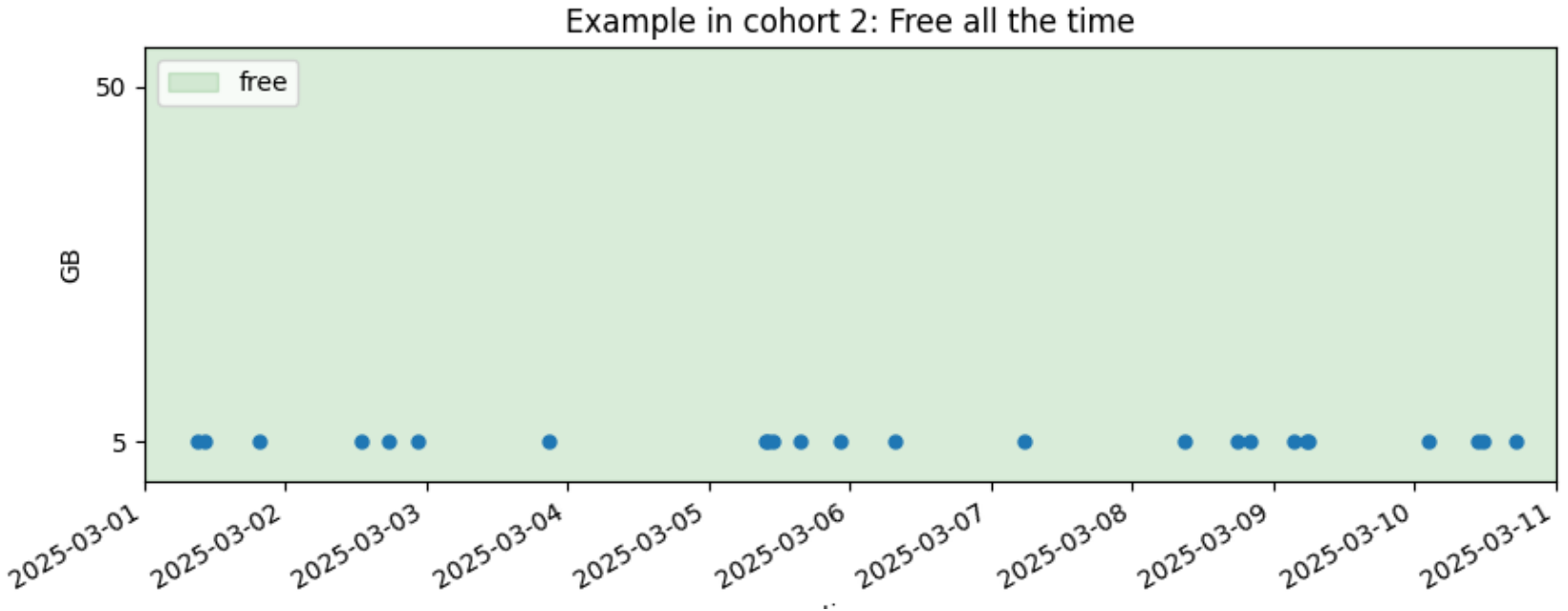
- Users: 50 user ids / names
- Items: 80 items ids (45 of 5GB, 35 of 50GB)
- Tiers: 10 in each cohort: pick random times during from March 1st to March 10 for the status change.
- Uploads: generated so that for premium users:

  P[50GB upload with next h hours] ~ 1 - exp(-h*lambda)

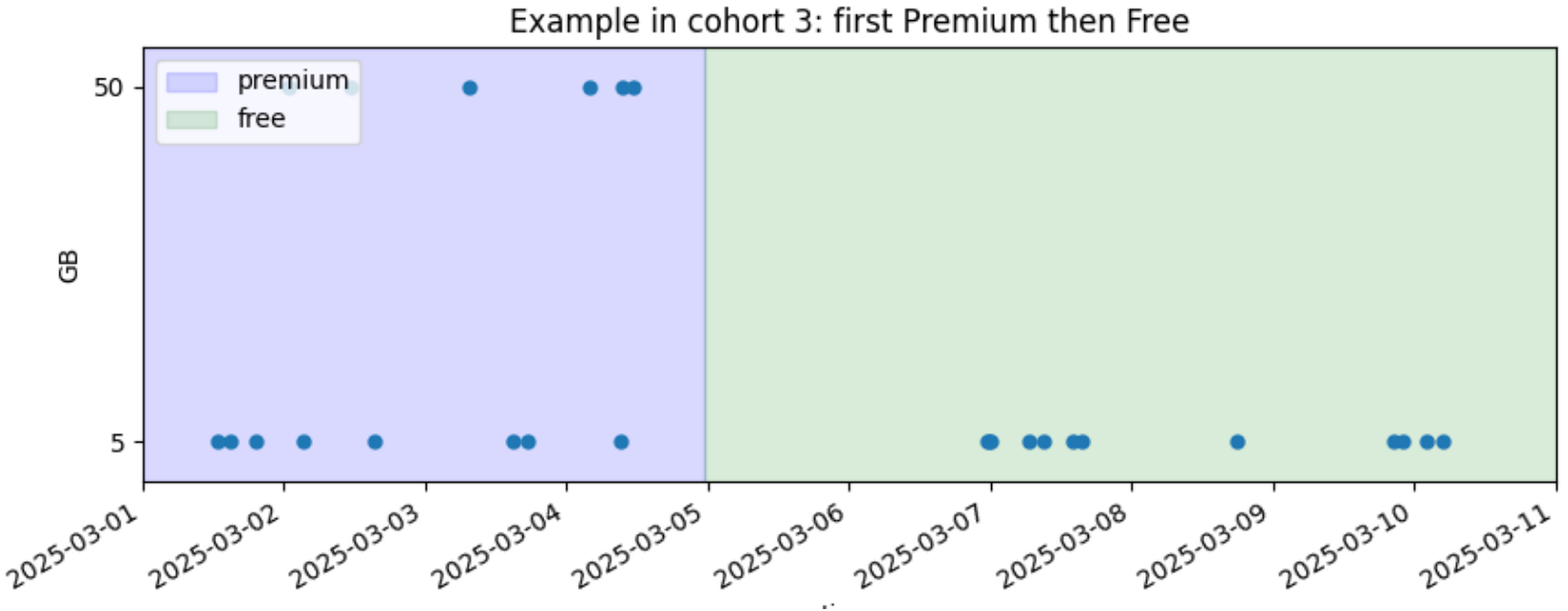
Premium uploads example



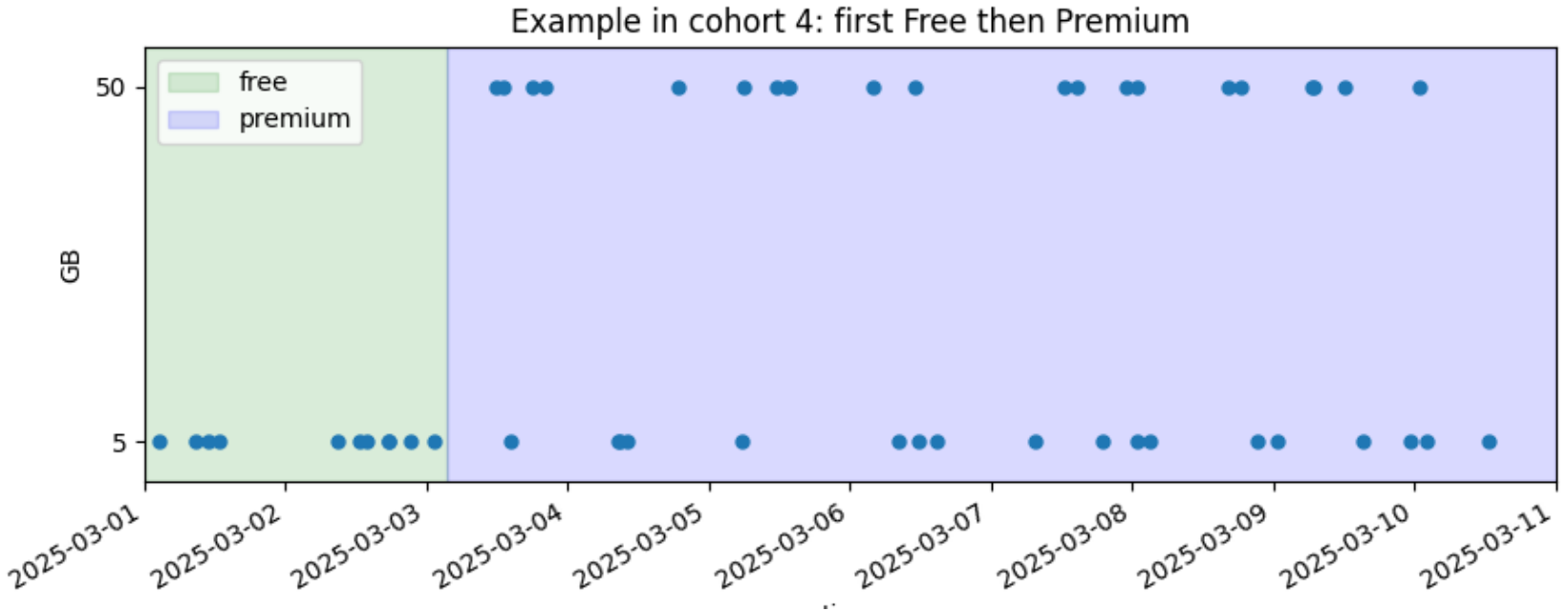
Free uploads example



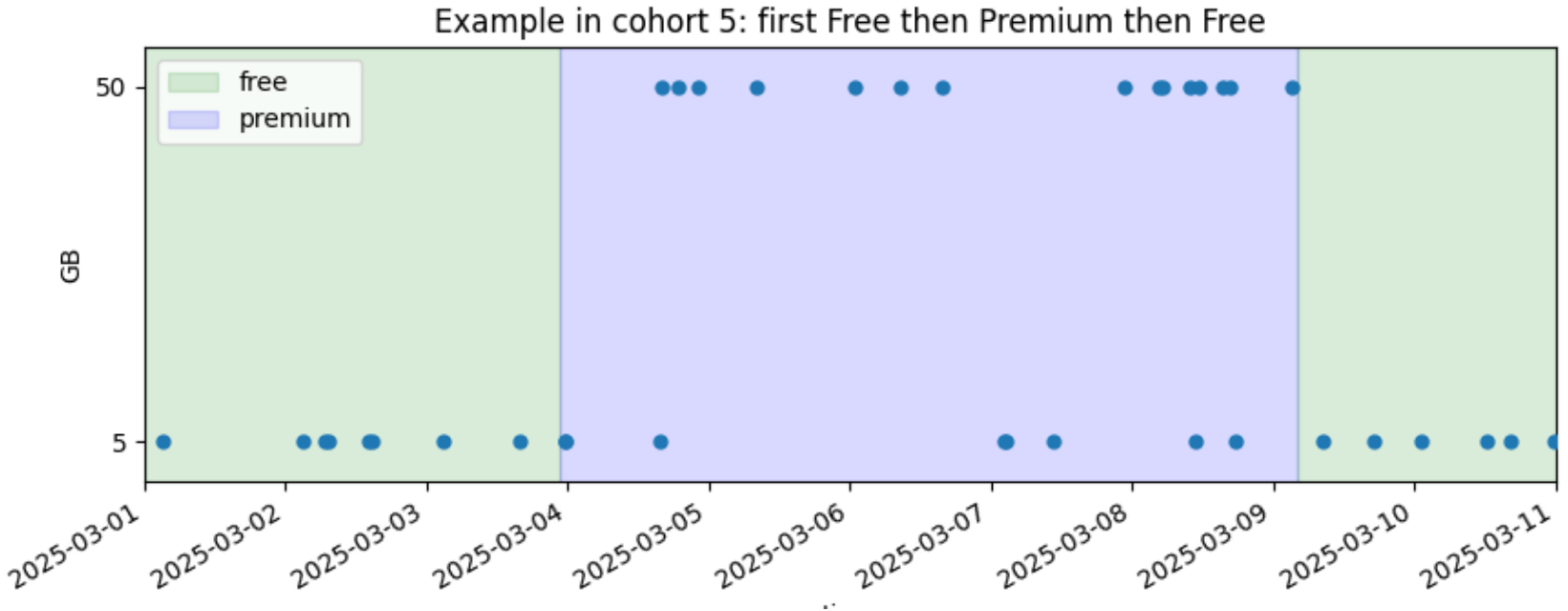
Premium-Free uploads example



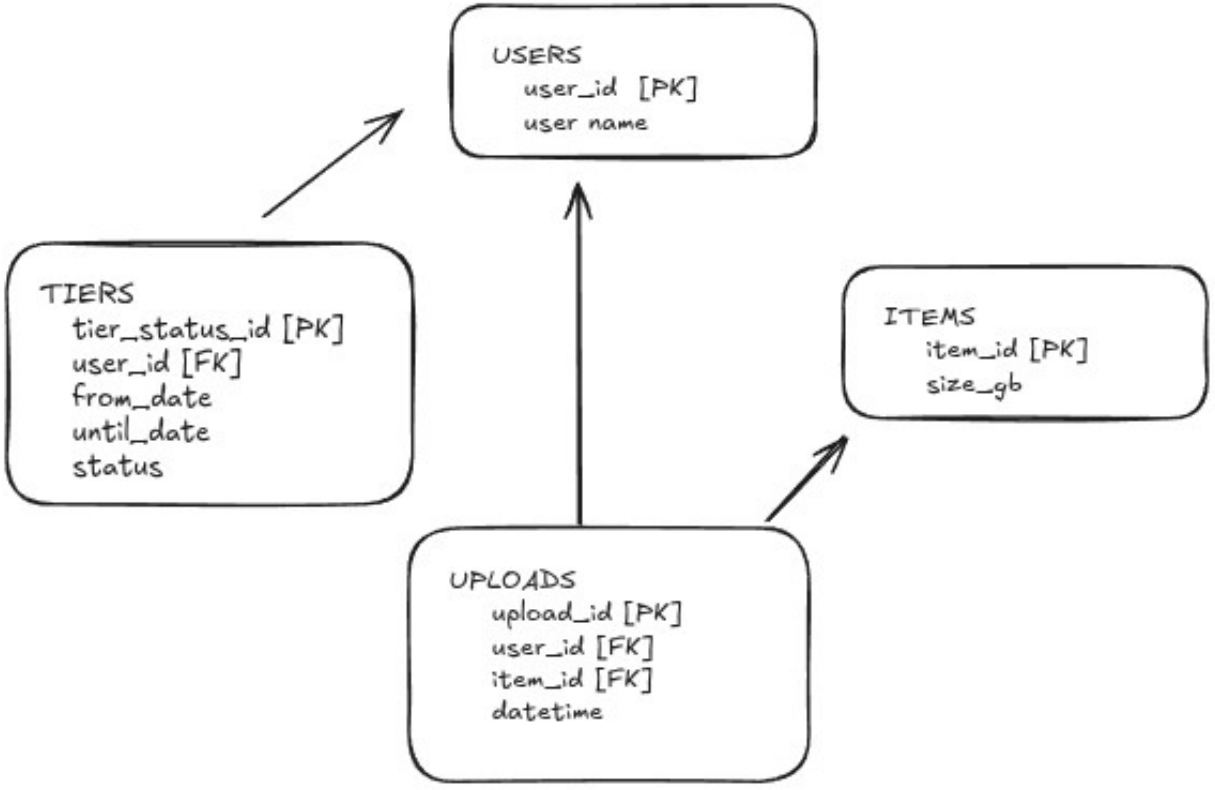
Free-Premium uploads example



Free-Premium-Free uploads example



Graph for KumoRFM



First attempt at prediction (failed)

```
PREDICT
COUNT(uploads.* where items.size_gb=50, 0, 3, hours)>0
FOR users.user_id = 5
```

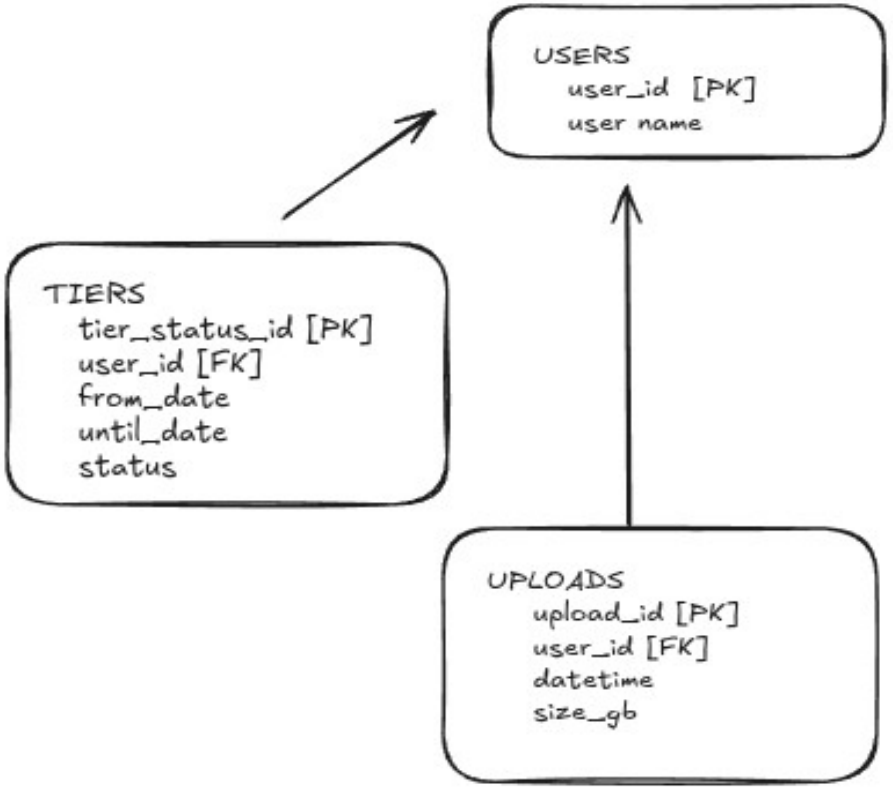
- Error: Unsupported implicit join.

  Static references to columns from other tables that implicitly contain a foreign key to primary key connection are not supported in foundation model queries.

  Your query implicitly requires a join uploads.item_id -> items.item_id. Please remove the reference to table items and retry, or train a new model instead of the foundation model.

- My solution: add size_gb to uploads by joining with items (then ignore items).

Graph for KumoRFM (after fix)

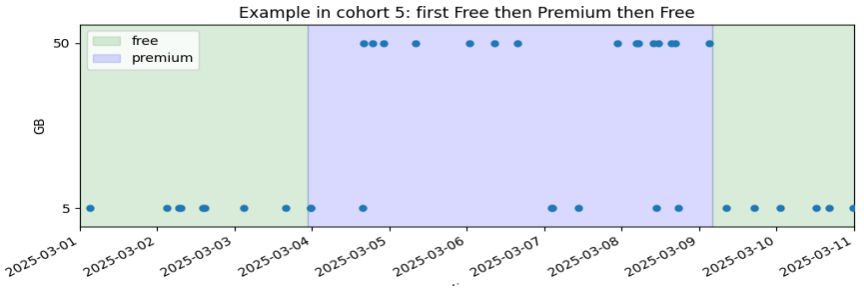


Predictions

```
query = f"""
      PREDICT
      COUNT(uploads.*
       where uploads.size_gb=50, 0, {hours}, hours)>0
      FOR users.user_id = {uid}
      """
```

- Compare predicted vs actual probabilities.
- Premium actuals: 1 – exp(-lambda* hours)
- Free actuals: 0

Actual vs Predicted for Free-Premium-Free user

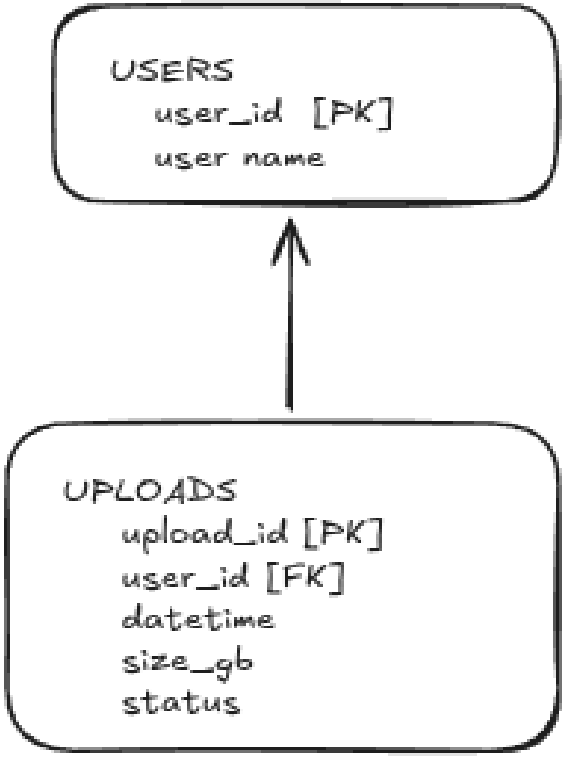


user_id	datetime	status	Hours (prediction window	Actual Probability	Estimated Probability
45	2025-03-02 11:15:07	Free	1	0.0000	0.0005
			2	0.0000	0.0009
			4	0.0000	0.0088
	2025-03-06 13:20:27	Premium	1	0.0952	0.1037
			2	0.1813	0.1719
			4	0.3297	0.3336
	2025-03-10 02:05:20	Free	1	0.0000	0.0534
			2	0.0000	0.1665
			4	0.0000	0.1198

Observations on predicted probabilities

- Probability estimation on Premium works well.
- For Free, it works well only in the first period.
- For Free, in the second period, seems influenced by the stats during the Premium period.
- Increasing number of hops does not help.
- Final attempt: include status in uploads.

Graph for KumoRFM (after 2nd fix)



Predictions

```
query = f"""
            PREDICT
            COUNT(uploads.*
                        where uploads.size_gb=50
                        and uploads.status='{status}', 0, {hours}, hours)>0
            FOR users.user_id = {uid}
        """
```

Actual vs Predicted for Free-Premium-Free user

user_id	datetime	status	Hours (prediction window	Actual Probability	Estimated Probability Now	Estimated Probability Before
45	2025-03-02 11:15:07	Free	1	0.0000	0.0000	0.0005
			2	0.0000	0.0000	0.0009
			4	0.0000	0.0000	0.0088
	2025-03-06 13:20:27	Premium	1	0.0952	0.0996	0.1037
			2	0.1813	0.1648	0.1719
			4	0.3297	0.2843	0.3336
	2025-03-10 02:05:20	Free	1	0.0000	0.0000	0.0534
			2	0.0000	0.0000	0.1665
			4	0.0000	0.0000	0.1198

Final Questions

- How to deal with the temporal dependencies on 2 columns.
- What's a recommended approach to process the tables in order to extract maximum value from data and the models.