
Churn predictiction Problem Ideation:

As data is monthly, this is monthly churn problem. because

Understand given data

- Data timeline: monthly cohort
- Given data: 20k users, 3 months activity(app_installation, app_start(open) and brochure_view behavior)

How do we define churn in this case?

To define churn we have below options:

- User that are active in April-June but less activity in july?
- User that are active in April-June but innactive in july? ("faeza karmran": ye to definetely hoga.)

We choose Users that are completely innactive in july. Because

- As original casestudy states `which user is still active in July based on the behavior of April until June`
- To be 100% sure that these users are churned and this is not a seasonal behavior.
- In any case for the people less active in july, we will eventually have churn labels from August. (Monthly churn prediction)
- If we really wanna consider churning people, we could add another label to Y (not-churn, churning, churned). (Basically targeting early churning phase)
- We choose Users that are completely innactive in july. Because

---

To calculate churn, what could be useful features?

- Customer_Lifetime (days since app installed/joined)
- app_open_count, app_opening_rate (Frequency), days_since_last_opened (Recency), avg_app_open_gap, app_open_to_brochure_view_funnel
- brochur_view_count, brochur_view_rate(Frequency), time_since_last_brochur_viewed (Recency), commulative_broucher_view_duration, avg_broucher_view_time, maximum_visited_broshur
- Further useful features could be (not added to notebook)
  - average_app_session_time, maximum_brochurs_seen_in_one_go, app_open_to_borchur_view_gap (could probably show user's interest)
  - complains / sentiment / rating about app
  - User Demograhics:
    - age, gender, location, financial_status (Occupation, based on home-address (expensive areas), Home Ownership (Rent, Own), and Mobile model maybe!) , Marital Status

---

- To create these features, we have these options:
  - By merging all tables on user_id then drive these features from a single DF…()
  - Take a user_id list (as starting **user_base**) and for each id (from user base) we drive features using other given files (app_open and borchure_view)

We choose option 2, because:

- Option 2 is efficeint and sophisticated. Stepwise data validation.

- Also, option 2 doestnt work well because Df1 size gets multipled by Df2 size (N x M)
- ————————————————————————————————————
- **user base** candidates.
    - 22493 unique users in merged_Df (installs + app_open + brochur_viewed) (Contain Alien users: Some people from app_start and brochur_view dataframes arent in installs, I called them alient users)
    - 20k unique users from installs
    - 8k unque users from brochure_view (brochure viewed only base).
- **We choose option 2** (20k users from installs.txt). Because,
    - Original task mentioned 20k users: Probably expected me to solve for them.
    - We have whole lifestime data for these users. So, we can still calculate all important features (e.g `weekly_brochur_view_rate = 0`, `commulative_broucher_view_duration = 0`)
    - **Cons of other options**
        - Incomplete data for alien users, e.g we do not know their Lifetime with app. (we can fill missing values, but not worth for now)
        - Only considering people highly active if only consider Option 3: People who not seen brochure are churned indeed, if they are viewing brochure then this kind of means we are looking at engaged users only.
        - in unseen (test) data, we might have similar situations where user didn't engage.

————————————————————————————————————