# Experimentation Design Report

## I. CONTEXT

In order to grow net profit, we either need to increase the number of movies rented per user or the number of users renting movies. There are several ways to achieve this objective, one of which is leveraging our recommendation engine to surface relevant suggestions to users that influence their decision to rent more movies.

Currently, our users discover movies via three modules, 1) movies recommendation, 2) communities recommendation, 3) search. In this document, we will mostly be discussing the strategies to use “communities module” to drive more movies being rented.

## II. HYPOTHESIS

By surfacing relevant communities to the user we’ll improve movie discovery which will then increase the number of movies rented by users

## III. TARGET USERS

- Logged in users

## IV. ASSUMPTIONS

- Keeping the UI constant across all variants
- The UI shows 5 recommendations at a time
- Taking 90 days user journey data to capture seasonality change

## V. EXPERIMENTATION DESIGN

- A/B testing

### CONTROL (33% traffic)

- No changes introduced

### TEST VARIANTS (33%/33% traffic)

1. Genre based communities recommendation by user’s viewing history in the past 6 months
2. Communities recommendation by popularity

# VI. METRICS

a) Business metrics

| Indicator | Metrics | Explanation |
|---|---|---|
| Primary metrics | % of users renting a movie control vs test variant | See an increase in this metric to know the user behavior impact |
| Secondary | Avg no. of movies rented per user per month | This is a lagging indicator and will take some to evaluate the impact, especially if the test is concluded within 15 days |
| Guardrail | Net revenue control vs test variant | This is a do-no-harm to make sure there no negative impact |

b) Recommendation model performance metrics
Where K = {5, 10, 15, 20}

| Metric | Definition | Explanation |
|---|---|---|
| Hit rate @K | Percentage of users that joined at least one predicted recommendation in top K predictions | A high hit rate value reflects the algorithm's ability to provide something relevant to the user |
| sPrecision @K | K number of accurate predictions/total predictions (K) | Represents the percentage of predicted works that the user shelved |
| Coverage @K | Unique number of communities recommended in top K recommendations | Higher coverage means more diverse recommendations across all users. While having high coverage is good, it has to be taken into consideration with relevance score |