OFFLINE AND ONLINE RANKING MODEL ELVALUATION IN INDUSTRY

What happens in the industry, where real users interact with the system, business interests affect the concept of relevance and pre-defined relevance judgments are not available?

STEP 1. DATA COLLECTION

IMPLICIT FEEDBACK
user interactions

1. Collection of users' interactions:
   REST API **SaaS SOLUTION**
2. Model interactions as **JSON** objects.
3. Relevance label estimation using interactions aggregation:
   Click-Through Rate, Add-To-Cart Rate, ...
4. **Test set** extraction/creation and Kibana **dashboard** creation.


```
{
    "interactionType": "ProductImpression",
    "productId": 21356,
    "productPrice": 34.3,
    "productCategories": [43, 64, 100],
    "querySelectedCategory": 23,
    "userId": "46b60",
    "userFavouriteCategories": [157, 12, 81]
} {
    "interactionType": "ProductClick",
    "productId": 465,
    "productPrice": 14.9,
    "productCategories": [43, 21, 103],
    "querySelectedCategory": 23,
    "userId": "62k67",
    "userFavouriteCategories": [142, 12, 75]
} {
    "interactionType": "ProductImpression",
    "productId": 2473,
    "productPrice": 104.0,
    "productCategories": [22, 74, 124],
    "querySelectedCategory": 12,
    "userId": "46b60",
    "userFavouriteCategories": [157, 12, 81]
}
```


ALTERNATIVE

EXPLICIT FEEDBACK
team of experts

1. Judge **<query-document>** pairs → no position bias
2. Judge **search results list** items

Judgement Collector Chrome **plugin** available.

STEP 2. EVALUATION APPROACH

A/B TESTING

Approach choice:

REST API **SaaS SOLUTION**

- Design parametric search-API
- Assign users to a population through cookies
- Tag interactions with the test group

INTERLEAVING

Team Draft available from:

Solr version 8.8.0

Winner estimation process needed (python script):

- $$\Delta_{AB} = \frac{wins(a) + \frac{1}{2} ties(A,B)}{wins(A) + wins(B) + ties(A,B)} - 0{,}5$$
- Query distribution analysis (Long tail vs Uniform)

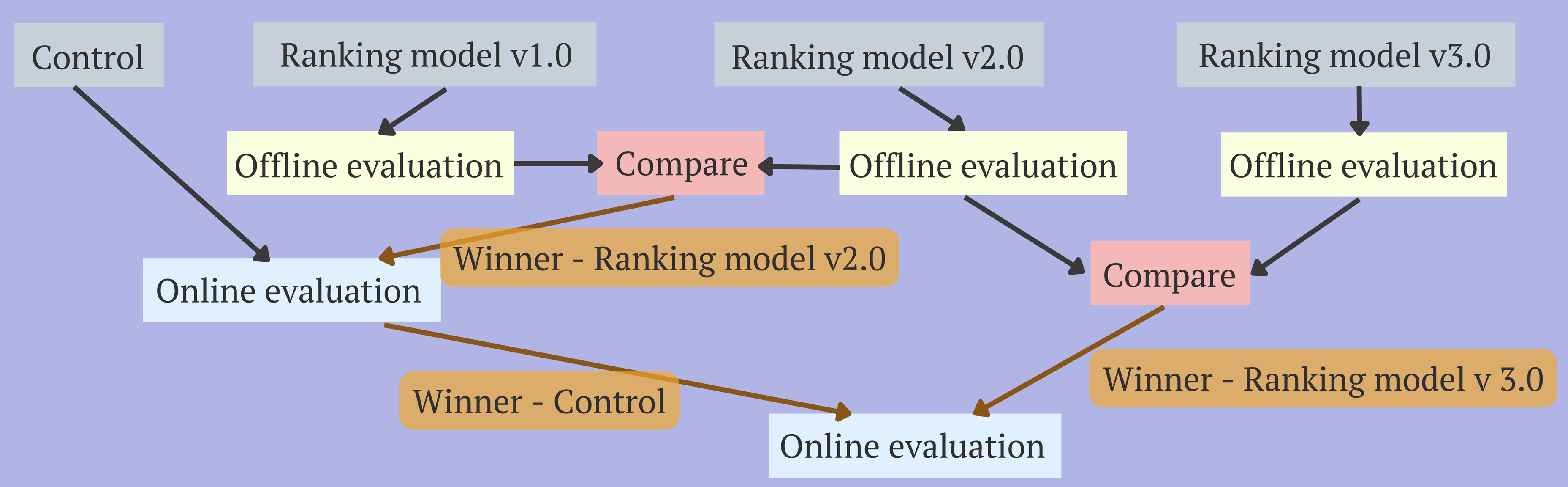
FOR BOTH

STATISTICAL SIGNIFICANCE

- Relevance label estimation
- Check sample distribution
  - (outliers, normality, homogeneity)
- Transformation to normal distribution

1. **One-way ANOVA** test
   - Effect size
   - Tukey test
2. **Kruskal-Wallis** test
   non-parametric (no normal distribution required)
   - Effect size
   - Dunn test

STEP 3. EXPERIMENT DESIGN



Baby steps between experiments: each experiment compares similar models (few features more, different normalization).
One experiment per platform (desktop, mobile, ...).
Evaluate 2-3 models at the time.

WHEN DO WE STOP?

1. **Statistical significance**
2. Number of users
3. Time = development iteration length

STEP 4. PITFALLS

QUERY

1. Query id generation: **too-specific** vs **too-generic**
   - Too short/too long ranked lists per query → balanced is needed
2. Number of results: **large result set query** vs **small result set query**
   - Small result set queries → expected small ranking model impact
   - Many small result set queries → cause noise in the evaluation

INTERACTIONS

Noise: **position bias** - **source pages** - **errors during collection**
- Users tend to click on top-ranking results
- Online evaluation → select only interactions from pages that use rank models

METRIC

Choose metrics: **industry's interests**
- Estimate offline relevance label → business objective (clicks, add-to-cart, downloads, ...)
- Offline metrics needs support by Online metrics

TEST SET

Per query data and relevance distribution: **unbalanced**
queries with a single sample
queries with a single relevance type


```
{
    "relevance": 1,
    "productId": 21356,
    "queryId": 23,
    "userId": 46b60
} {
    "relevance": 1,
    "productId": 465,
    "queryId": 23,
    "userId": 62k67
} ...
```

```
{
    "relevance": 1,
    "productId": 465,
    "queryId": 23,
    "userId": 62k67
} {
    "relevance": 1,
    "productId": 21356,
    "queryId": 11,
    "userId": 46b60
} {
    "relevance": 3,
    "productId": 2473,
    "queryId": 23,
    "userId": 46b60
} ...
```


OPEN SOURCE TECHNOLOGIES

pandas
Solr
elastic
kibana
statsmodels
scikit learn

Sease INFORMATION RETRIEVAL APPLIED

Alessandro Benedetti - Director and R&D Software Engineer
Anna Ruggero - R&D Software Engineer and Search Consultant