Session-Based Recommendation Systems in E-commerce

AN488 Oral Presentation

Presented by:
Elumalai Logeswaran| Y1882271

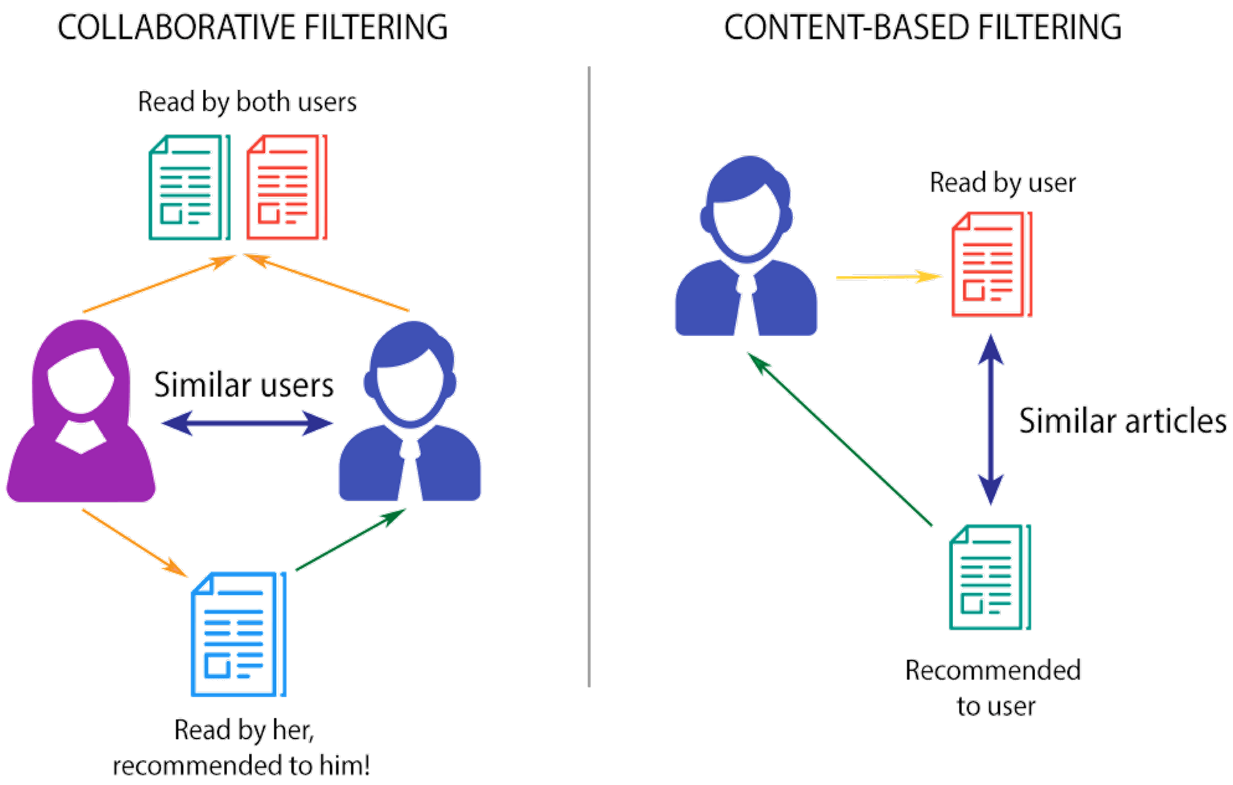
Recommendation Systems:

“applications that use data from various sources to learn about customers and infer their interests to recommend product(s) or service(s) that they might find helpful (Schafer et al., 2000)”

- Information Overload *(variety of products/services)*
- Increase sales for e-commerce business:

  1. *Turn browsers into buyers*
  2. *Cross-sell*
  3. *Gain customer loyalty*

Conventional Recommendation Systems



Use wisdom of the crowd
"Others have also brought"

Historical user information
"You might also like:"

- Typical recommendation techniques *(type of input data available)*

Underlying assumption:

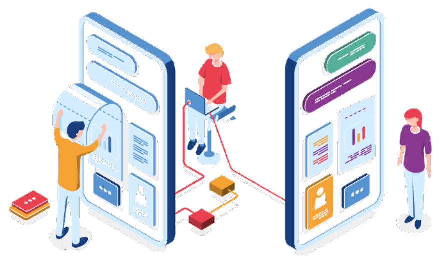
"all of the historical interactions are equally important to the user's current preference"

- Long-term yet static user preferences

A case for Session-based recommendation Systems

Session-Based Recommendation Systems (SBRS):

"use the short-term and dynamic user preferences generated in a session to make timely recommendations"

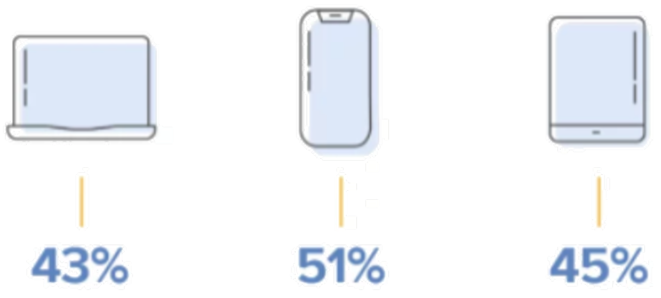


- Session data *(information generated by user)*

  Known part > Learn dependencies > predict unknown

- Why are they needed for e-commerce?

  1. *Few recurring users*
  2. *High bounce rate (47% in 2020)*
  3. *User use site without logging in or anonymous (cold-start)*



(2020 Contentssquare benchmark data for e-commerce sites)

Coveo Data, Business & Data Mining Objectives

- SIGIR eCom Coveo Data Challenge 2021 *(Open-source dataset by Coveo)*

  1. *Browsing events*
  2. *Search events*
  3. *Product catalogue*

coveo™

Business Objective:

- Increase sales/profit by converting anonymous browsers into buyers

Unnamed Mid-size Shop Z

Data Mining Objective:

- Use the session data provided by Coveo to develop a session-based recommendation system

Data Understanding I

{ Browsing_train: Browsing information generated from anonymous user session. }

Attribute	Description	Data Type	Data Quality Issues
Session_id_hash	Hashed Id of shopping session Total: 36,079,307 Unique: 4,934,699	object	Missing data: 0
Event_type	pageview = 25,647,696 event_product = 10,431,611	object	Missing data: 0
Product_action	detail = 9,707,890 add = 329,557 remove = 316,316 purchase = 77,848 All from event_product	object	Missing data: 25,647,696
Product_sku_hash	All from event_product 57,483 unique product SKU	object	Missing data: 25,647,696
Server_timestamp_epoch_ms	Epoch time, in milliseconds	integer	Missing data: 0
Hashed_url	Hashed url	object	Missing data: 0

Data Understanding II

{ Search_train: Search activities information related to user sessions. }

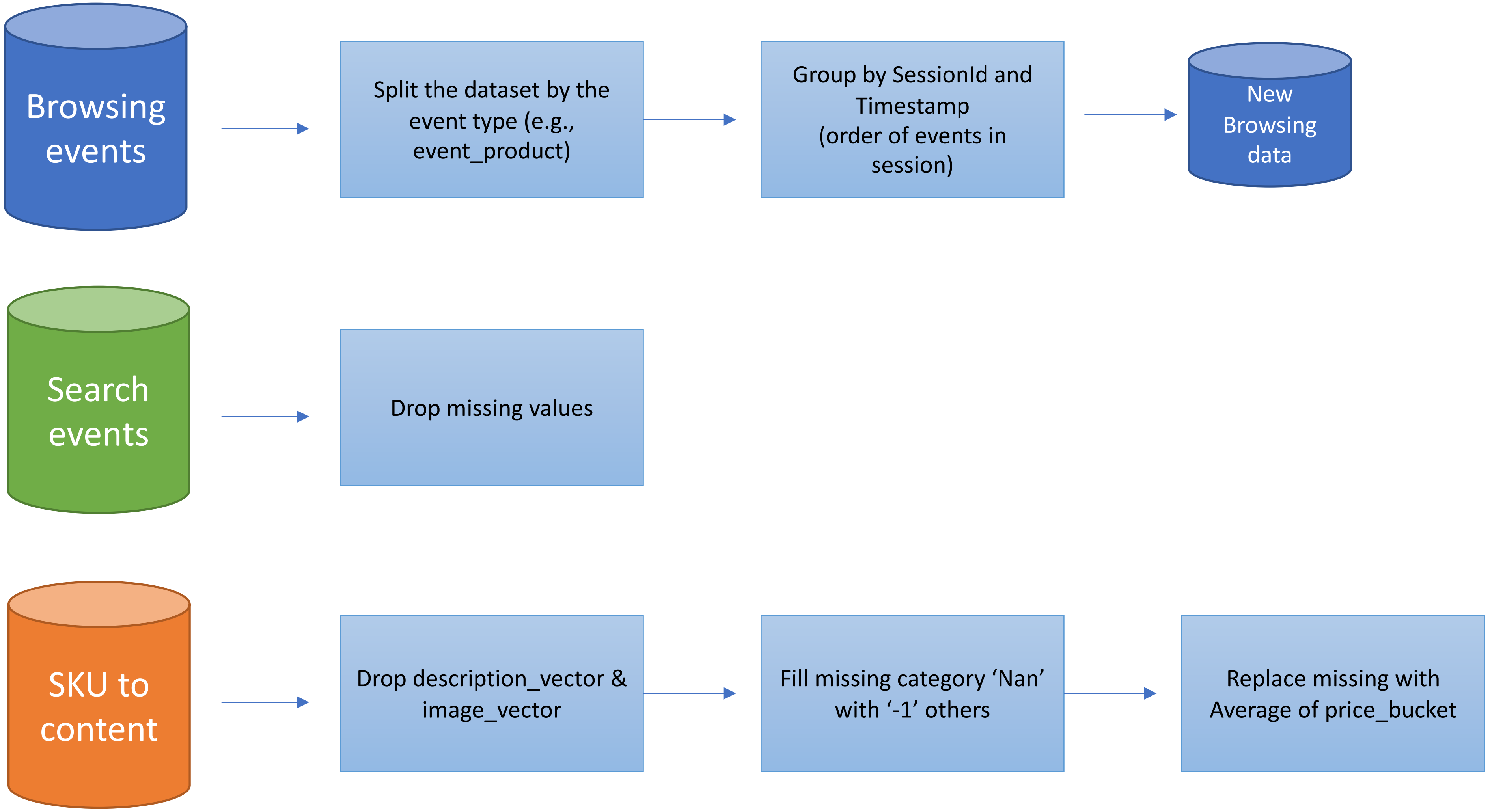
Attribute	Description	Data Type	Data Quality Issues
Session_id_hash	Hashed id of shopping session Total: 819, 516 Unique: 555, 100	object	Missing data: 0
Query_vector	Vectorised search query	object	Missing data: 0
Clicked_skus_hash	Hashed Product skus that was Clicked Unique: 73,311	object	Missing data: 640,021
Product_skus_hash	Hashed Products displayed from search	object	Missing data: 216,762
Server_timestamp_epoch_ms	Epoch time, in milliseconds	integer	Missing data: 0

Data Understanding III

Sku_to_content: Catalogue of products & information related to them.

Attribute	Description	Data Type	Data Quality Issues
Product_skus_hash	Hashed product sku (id) Total: 66, 386 Unique: 66,386	object	Missing data: 0
Description_vector	Dense representation of textual metadata	vector	Missing data: 34436
Category_hash	Hashed category of the product Unique: 174	object	Missing data: 34334
Image_vector	Dense representation of image metadata	vector	Missing data: 38016
Price_bucket	10-quartile integer 1.0, 2.0, ….10.0		Missing data: 34348

General Pre-processing



Model Selection

Recommendation task:

Next Event Prediction - Predict and recommend next item to user based on previous product interactions in the session.

SBRS Approaches:

1. Conventional Approaches
- e.g., Pattern/rule mining, K-NN

2. Latent Representation Approaches
- e.g., latent factor model

3. Deep Neural Network based approaches
- e.g., Graph Neural Network

Difficult to understand & unintuitive

Chosen method:

NLP-based Approach - Word2Vec

Word2Vec for product recommendation

Word2Vec:

uses the <u>co-occurrence of words</u> in a <u>sentence</u> to learn embeddings for each word that capture the semantic meaning of that word.

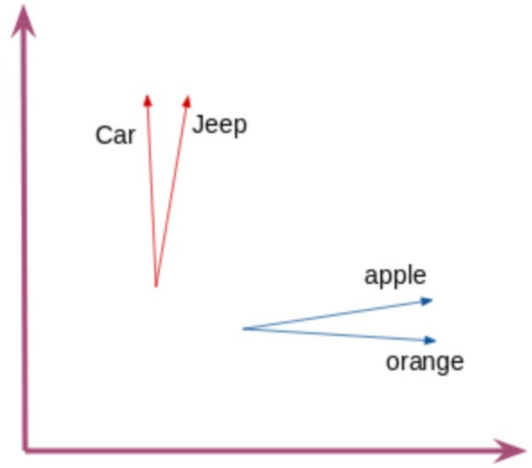
King – man +woman ~= Queen,

- Simple neural network *(single hidden layer)*

  - predict the nearby words for each and every word in a sentence

- Fundamental property that Word2Vec exploits in natural language *(Sequential Nature of text)*

  "these most been languages deciphered written of have already"



Word2Vec for product recommendation

- Sequential nature of sessions data *(order of products, pattern in consumer behavior)*

- Word2Vec for Product recommendation *(cast sessions as a NLP problem)*

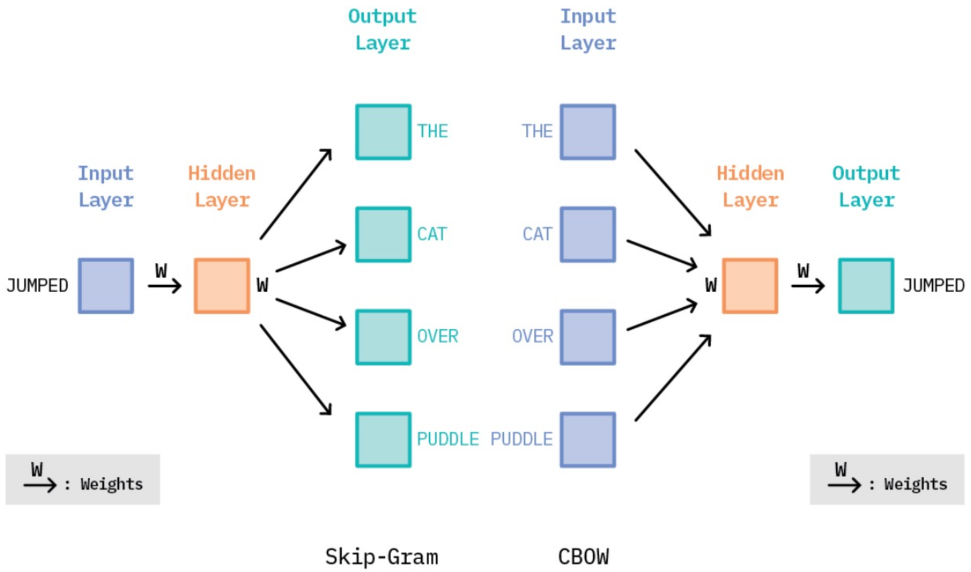
  1. *Treat each Session as a Sentence*

  2. *Treat each product in the Session as a word in a Sentence*

  3. *Collection of Sessions (user browsing histories) as Corpus*

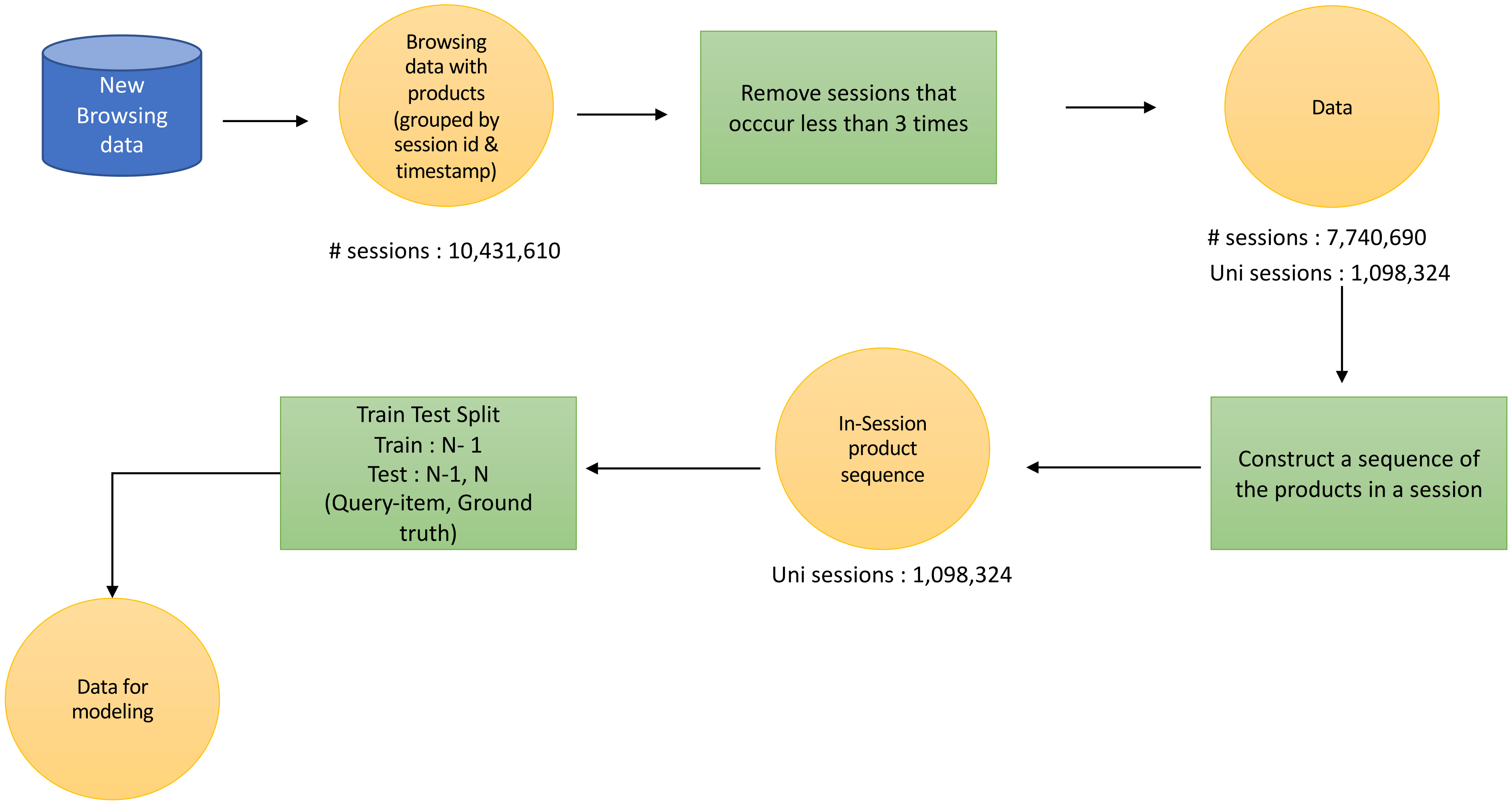
Word2Vec for product recommendation

- Variants of Word2Vec *(Skip-gram, Continuous bag of words - CBOW)*



- Word2Vec to learn dependencies between products *in the context of user browsing behavior (embedding for each product)*
- Semantic similarity between product IDs *(learned embeddings)*

Model Pre-processing



Modelling & Model Evaluation

Train Data

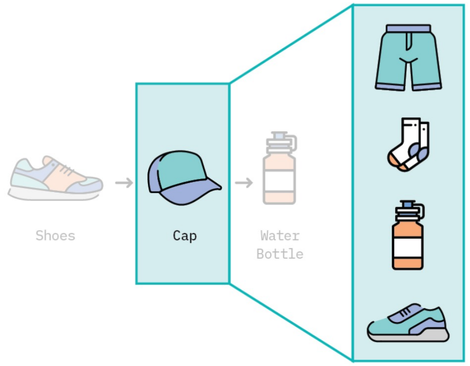
```
['D45425k', 'D45425k', 'D45425k', 'D45425k'],
['K17855w', 'K17855w', 'K17855w', 'I43384t'],
['E53621s', 'F76150x'],
['D53070j', 'J49710v', 'J49710v', 'K59963f', 'I12749s', 'E71639s', 'C51199h'],
['D65181b', 'D65181b', 'D65181b', 'D65181b', 'D65181b', 'I35618v', 'I35618v'],
['F58682n',
 'F58682n',
 'F58682n',
 'F58682n',
 'E65666t',
 'E65666t',
 'E65666t',
 'E65666t',
 'G46638q'],
['K35037y', 'K35037y', 'K35037y', 'K35037y'],
['J73206t', 'F56493z', 'G28059h', 'G28059h', 'G28059h', 'H54691i'],
['G27850j', 'G27850j', 'G27850j'],
['F67427s', 'H58805w', 'E32919k', 'E32919k', 'A43442x', 'A68665b', 'I58663f'],
['F77602e',
 'F77602e',
 'F77602e',
```

Uni sessions : 1,098,324

Test Data

```
['I57387h', 'I57387h'],
['A46170q', 'C67809m'],
['B48107f', 'B48107f'],
['E24658a', 'A43332t'],
['D13844o', 'D13844o'],
['D18816e', 'D38242x'],
['E65666t', 'E65666t'],
['I76131k', 'I76131k'],
['J26005h', 'J26005h'],
['K43529h', 'J68707y'],
['H74447s', 'G38960o'],
['D30685h', 'D30685h'],
['I36828o', 'I36828o'],
['D67084j', 'G75436a'],
['H74755h', 'C71670u'],
['B46369p', 'I25047s'],
['D20576a', 'D20576a'],
```

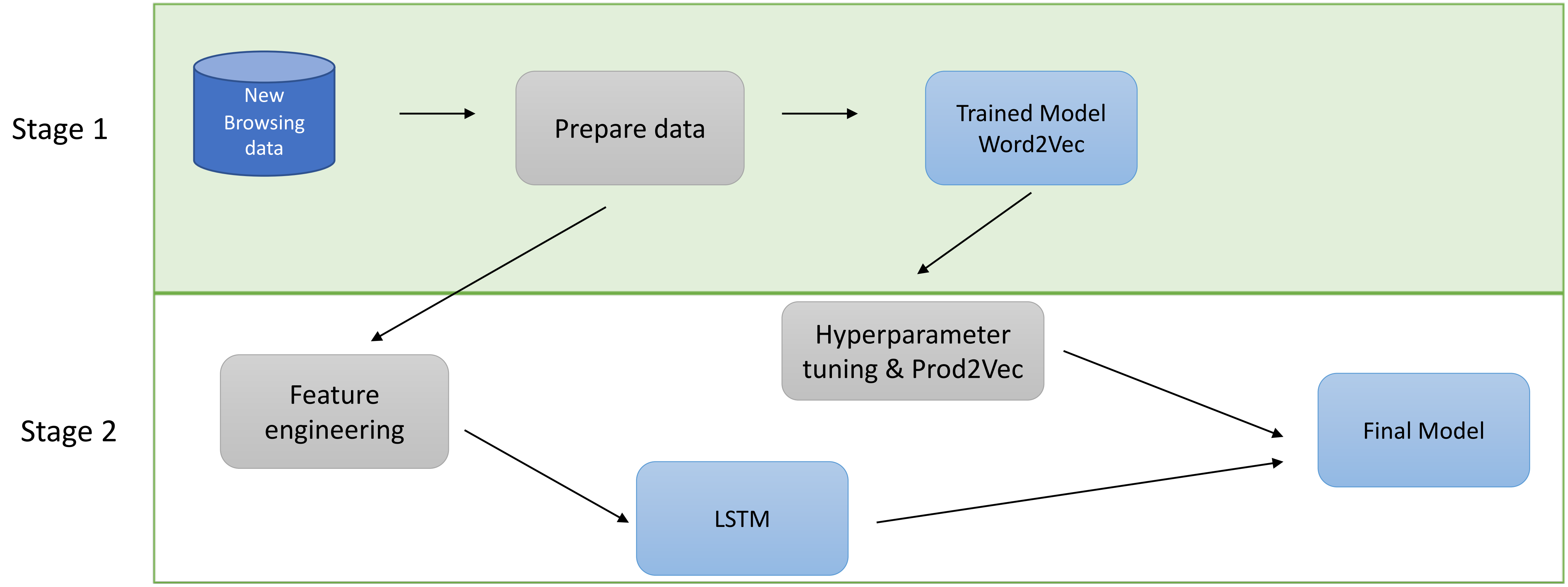
Uni sessions : 120,000



Word2Vec: Skip-gram method results

Modelling Results	K=10
Recall@K	0.23535
MRR@K	0.124558

Next Steps: Model Improvement



Next Steps: Model Improvement

Feature engineering

SessionId	last_event_length	last_hashed_url	same_url	pageview	cum_pageview	cum_product	cum_search
0	267592.0	NaN	False	False	0	1	0
0	998119.0	0aa1084eddfb08e4dffbb5a2aa98a5e9679382d982dd97...	False	False	0	2	0
0	6436.0	83b4fdad686c1be4eba335f70d23ae202b84b6153e109e...	False	True	1	2	0
0	21993.0	0ad6fab1eb3ac76010ea2fa6399a4e993b00f6501c88a2...	False	True	2	2	0
0	281069.0	e93e5c83aab0987e41d8fd65a30b54d2ce87491b4a7f9b...	False	True	3	2	0
...	...	...	...	...	...	...	...
5010681	18765.0	189a154674efcb6ab196fd1f5341be5b3fd5cf4422bf0f...	True	True	17	9	0
5010681	22536.0	189a154674efcb6ab196fd1f5341be5b3fd5cf4422bf0f...	False	False	17	10	0
5010681	55768.0	400a4ea44e23f4cfec9e8129d84a9cb90a0cffecbb5406...	False	False	17	11	0
5010681	29090.0	0ca76955e075818c7eadaafddd0d7b565260fc83e84e06...	False	False	17	12	0
5010681	1454691.0	c556502992acafffde649f4fff0748f1fcce22f21f5e02...	False	True	18	12	0

cum_pageview	cum_product	cum_search	cum_event	first_time	lapse	sum_pageview	sum_search	num_late_pageview
0	1	0	1	1552423391039	0	9	0	9
0	2	0	2	1552423391039	998119	9	0	9
1	2	0	3	1552423391039	1004555	9	0	8
2	2	0	4	1552423391039	1026548	9	0	7
3	2	0	5	1552423391039	1307617	9	0	6
...	...	...	...	...	...	...	...	...
17	9	0	26	1550717835512	699908	18	0	1
17	10	0	27	1550717835512	722444	18	0	1
17	11	0	28	1550717835512	778212	18	0	1
17	12	0	29	1550717835512	807302	18	0	1
18	12	0	30	1550717835512	2261993	18	0	0

```
In [145]: train_all3.columns

Out[145]: Index(['event_type', 'product_action', 'product_sku_hash',
       'server_timestamp_epoch_ms', 'hashed_url', 'is_search', 'SessionId',
       'last_event_length', 'last_hashed_url', 'same_url', 'pageview',
       'cum_pageview', 'cum_product', 'cum_search', 'cum_event', 'first_time',
       'lapse', 'sum_pageview', 'sum_search', 'num_late_pageview'],
      dtype='object')
```

Hyperparameter tuning with Ray Tune


The End: Q & A..