RetailRocket ALS Recommender

Xiaolong, Fusheng, qing Gao, Wenxiong

1.

Business Goal and Value

recommend fashion products to customers

Business Goal

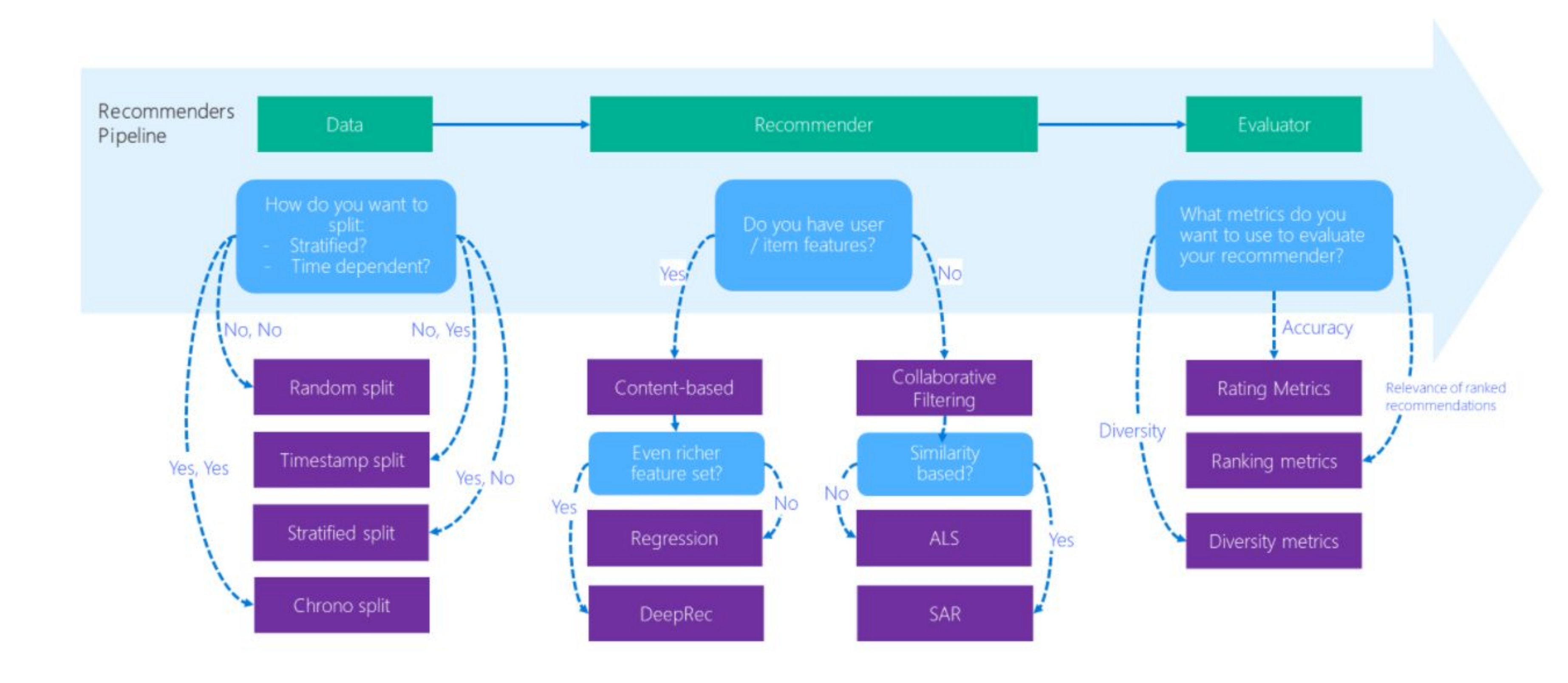
- ✘ Improve the customer experience by solving the problem of choice overload

- ✘ Encourage the customers to explore more products

2.

Methodology and techniques

Overview


Recommenders Pipeline
Data
Recommender
Evaluator
How do you want to split:
- Stratified?
- Time dependent?
No, No
No, Yes
Yes, Yes
Yes, No
Random split
Timestamp split
Stratified split
Chrono split
Do you have user / item features?
Yes
No
Content-based
Even richer feature set?
Yes
No
Regression
DeepRec
Collaborative Filtering
Similarity based?
No
Yes
ALS
SAR
What metrics do you want to use to evaluate your recommender?
Accuracy
Diversity
Relevance of ranked recommendations
Rating Metrics
Ranking metrics
Diversity metrics

Data Preprocessing

- ✘ Encode Event type to the following:

  "View":1, "Add to Cart": 5, "Purchase": 10

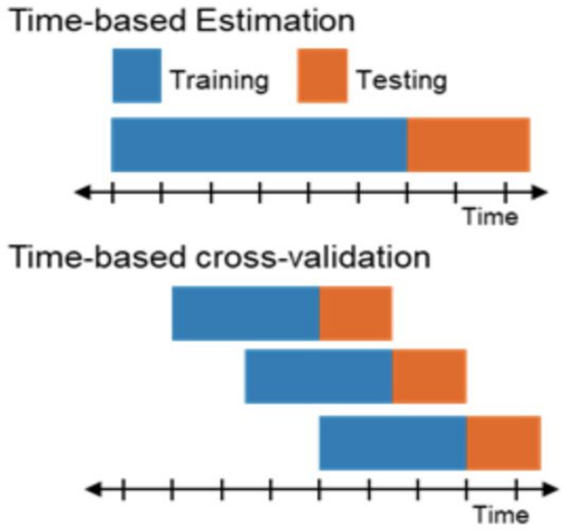
- ✘ Transfer Timestamp to readable Datetime Value

	timestamp	visitorid	event	itemid	transactionid
0	2015-06-02 05:02:12.117	257597	1	355908	NaN
1	2015-06-02 05:50:14.164	992329	1	248676	NaN
2	2015-06-02 05:13:19.827	111016	1	318965	NaN
3	2015-06-02 05:12:35.914	483717	1	253185	NaN
4	2015-06-02 05:02:17.106	951259	1	367447	NaN

Data Splitting

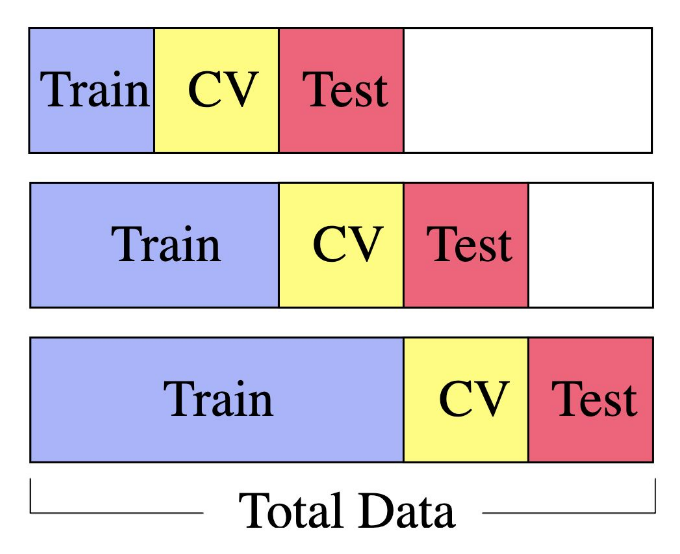
- ✘ Random Split
- ✘ Time Series Nested Cross Validation

"Sliding window" training Approach



Time based cross validation approach

Time Series Nested Cross Validation



Modeling

- ✘ Spark ALS

  Implicit (Library)

Library: Implicit

To install:

```
pip install implicit
```

Basic usage:

```
import implicit

# initialize a model
model = implicit.als.AlternatingLeastSquares(factors=50)

# train the model on a sparse matrix of item/user/confidenc
model.fit(item_user_data)
```

Implicit

build passing build passing

Fast Python Collaborative Filtering for Implicit Datasets.

This project provides fast Python implementations of several different popular recommendation algorithms for implicit feedback datasets:

- Alternating Least Squares as described in the papers Collaborative Filtering for Implicit Feedback Datasets and Applications of the Conjugate Gradient Method for Implicit Feedback Collaborative Filtering.
- Bayesian Personalized Ranking.
- Logistic Matrix Factorization
- Item-Item Nearest Neighbour models using Cosine, TFIDF or BM25 as a distance metric.

All models have multi-threaded training routines, using Cython and OpenMP to fit the models in parallel among all available CPU cores. In addition, the ALS and BPR models both have custom CUDA kernels - enabling fitting on compatible GPU's. Approximate nearest neighbours libraries such as Annoy, NMSLIB and Faiss can also be used by Implicit to speed up making recommendations.

3.

Meaningful results and discussion

Recommendation Example

	Random Split	Temporal Split
Train RMSE	1.07	0.98
Test RMSE:	2.006	1.9

Model Comparison

- "View":1, "Add to Cart": 5, "Purchase": 10
- rank = 10
- Test Error = 1.62

visitorid	itemid	rating	prediction
2133	137697	10	4.9893203
3465	8523	10	2.9910188
3465	114485	10	0.98868304
3465	434048	10	-0.30423677
3896	407518	10	2.9953449
4113	231807	10	1.4403526
4899	46156	10	1.7981739
6029	294267	10	1.7468264
6468	378760	10	4.910337
6952	461686	10	2.2747154

- "View":1, "Add to Cart": 5, "Purchase": 10
- rank = 10
- implicitPrefs = True
- Test Error = 1.45

visitorid	itemid	rating	prediction
2133	137697	10	0.010105458
3465	8523	10	0.003613429
3465	114485	10	1.640226E-4
3465	434048	10	1.8516456E-4
3896	407518	10	7.90489E-5
4113	231807	10	1.0887056E-4
4899	46156	10	0.18599321
6029	294267	10	0.011176619
6468	378760	10	0.0054947375
6952	461686	10	1.144534

- "View":1, "Add to Cart": 3, "Purchase": 10
- rank = 10
- Test Error = 1.45

visitorid	itemid	rating	prediction
2133	137697	10	2.991397
3465	8523	10	1.9925888
3465	114485	10	0.99090487
3465	434048	10	-0.4591227
3896	407518	10	1.996004
4113	231807	10	1.2194085
4899	46156	10	1.3979611
6029	294267	10	1.3835899
6468	378760	10	2.9256706
6952	461686	10	1.617531

4.

Rational Next Step

- ✘ Precision at k evaluation
- ✘ Handle long tail data
- ✘ Model Tuning to increase predicting power

Q & A