

# Learning to Rank Datasets for Search

Oscar Castañeda, Xoom a PayPal Service

#SAISDS8

# About

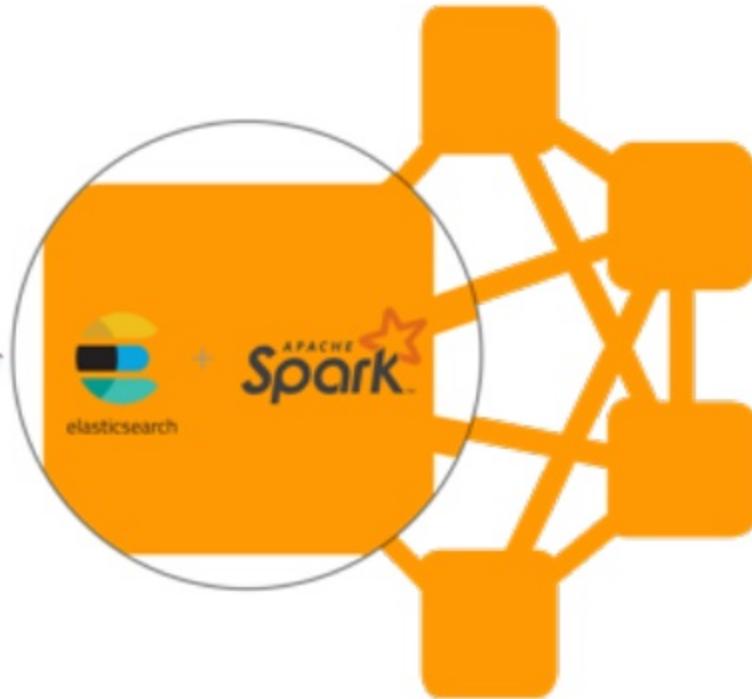
- Data Scientist at Xoom a PayPal service.
- Interests:
  - Data Management,
  - Dataset Search,
  - Learning to Rank.

# Spark cluster with Elasticsearch

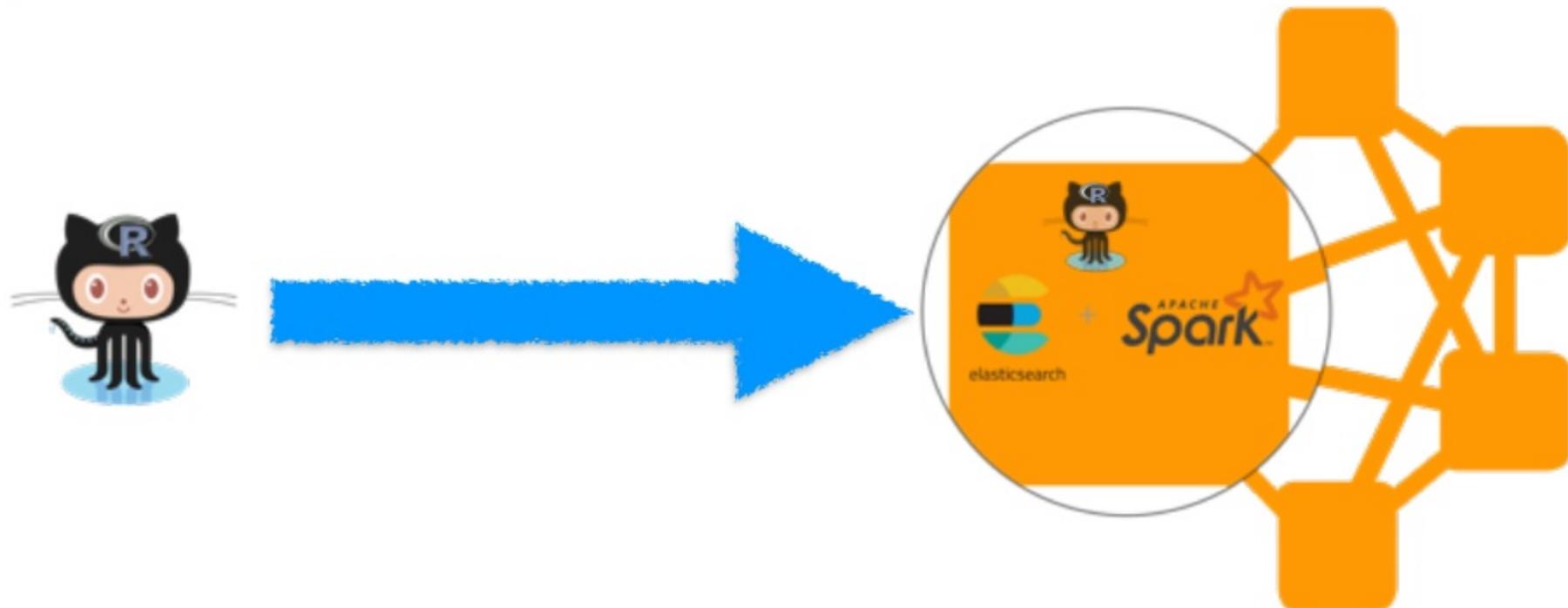
Elasticsearch



<https://mly/2eHM9R0>

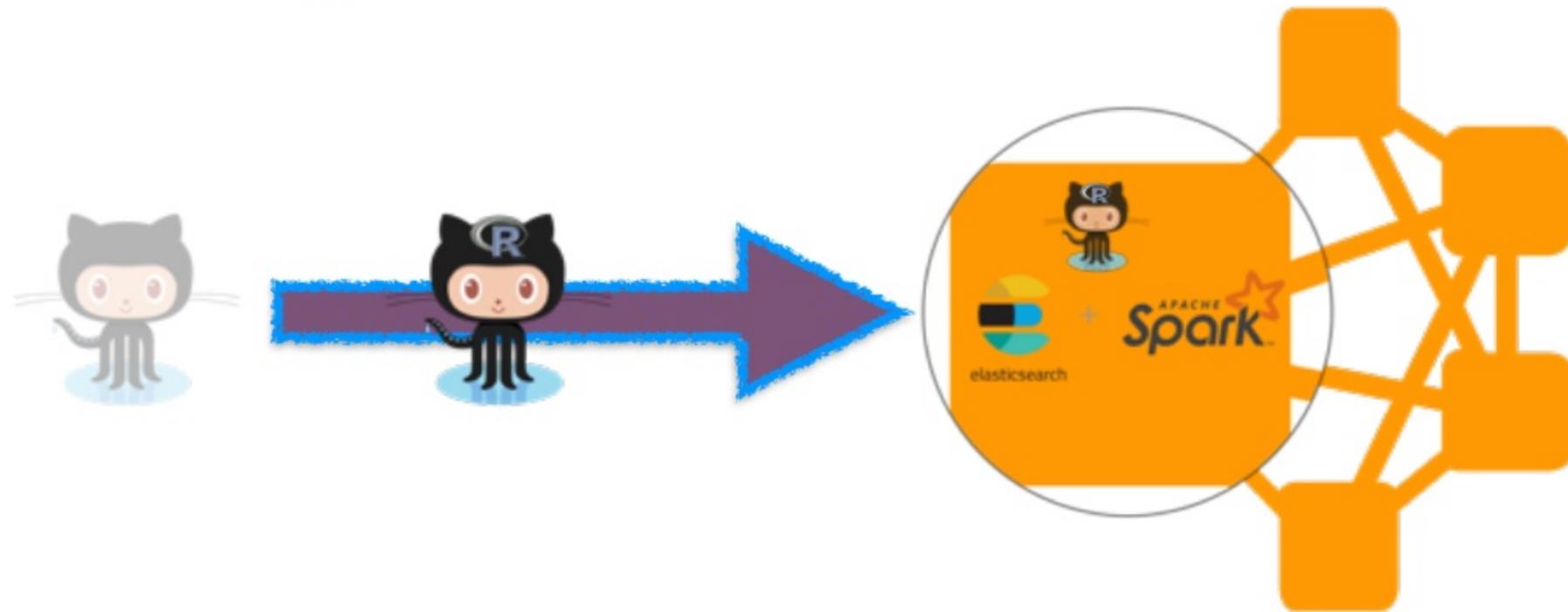


# Spark cluster with Elasticsearch Inside



And Indexed RDatasets

# Learning to Rank Datasets



# Learning to Rank Datasets



# Agenda

- Problem Statement and Motivation
- Elasticsearch Learning to Rank
- Data Pipeline: metadata extraction, judgement list extraction
- Demo: Beginnings of a Dataset Search Engine with Machine-learned relevance ranking.
- Q&A

# Problem Statement (1)

- Despite datasets being a key corporate asset they are generally not given the importance they deserve and as a result they are hard to find.

# Problem Statement (2)

- Specifically, teams within organizations have a hard time finding datasets relevant to their function.

# Topics

- Indexing

# Topics

- Indexing (Spark Summit East 2017).

# Topics

- Indexing (Spark Summit East 2017).
- Ranking

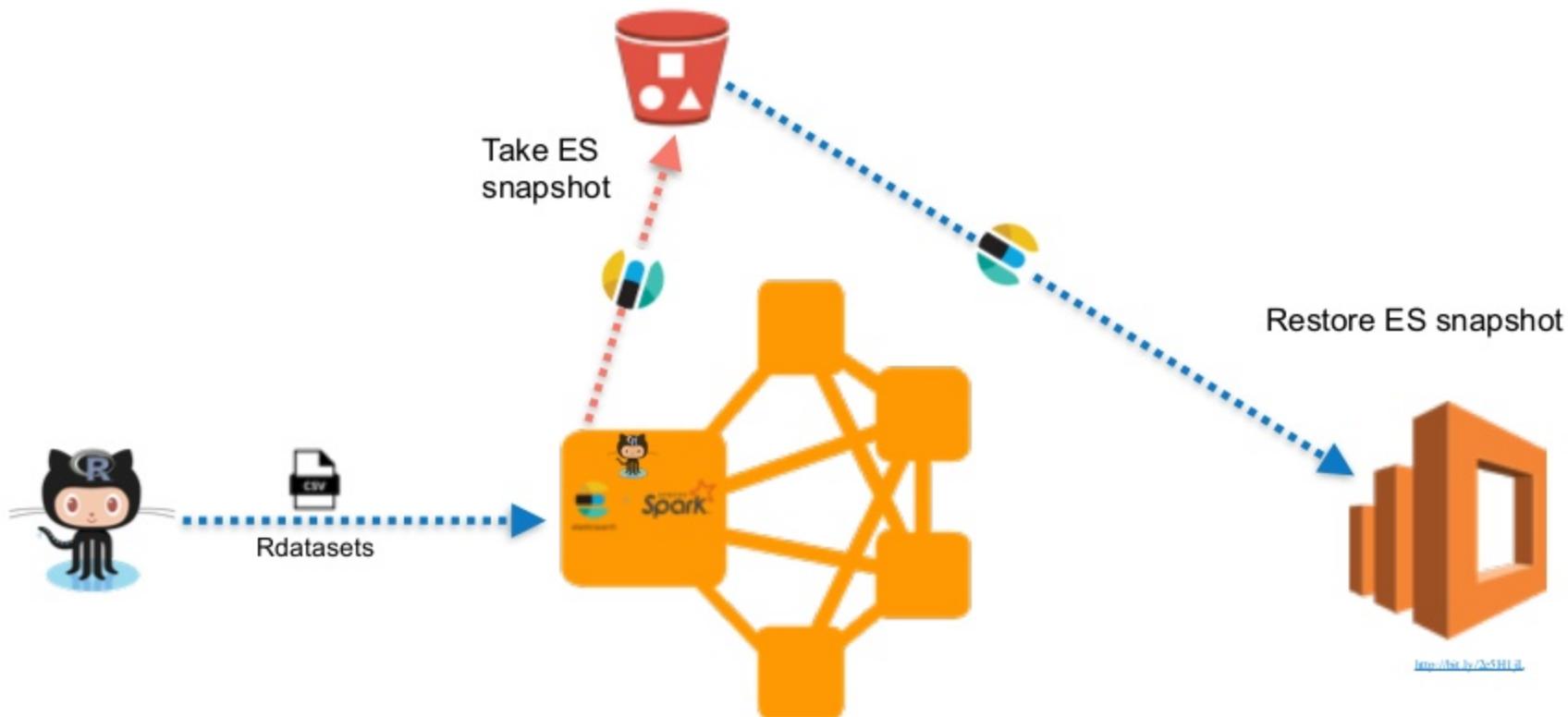
# Topics

- Indexing (Spark Summit East 2017).
- Ranking => today's topic!

# Questions

- How are datasets ranked?
- Can judgement lists (useful for ranking) be generated at dataset production time?

# Overview

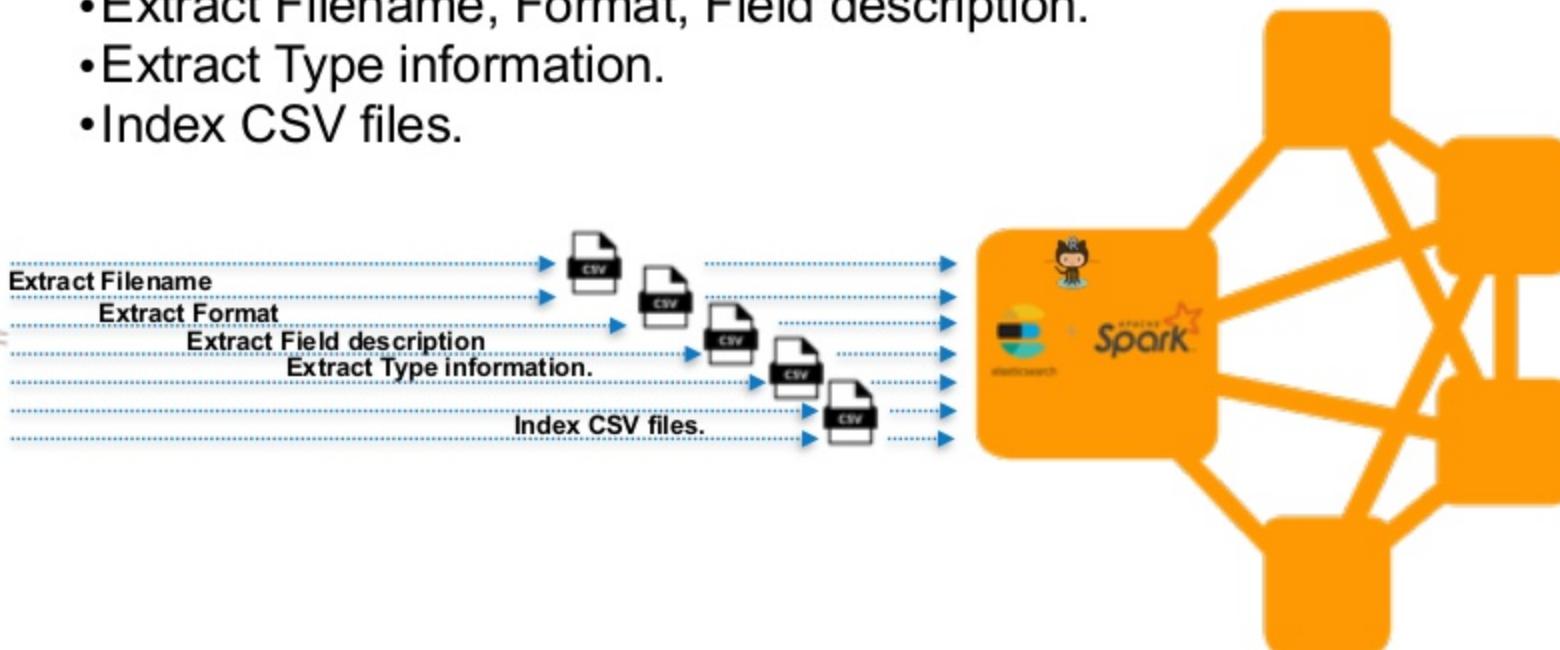


# Overview

## Data Pipelines:

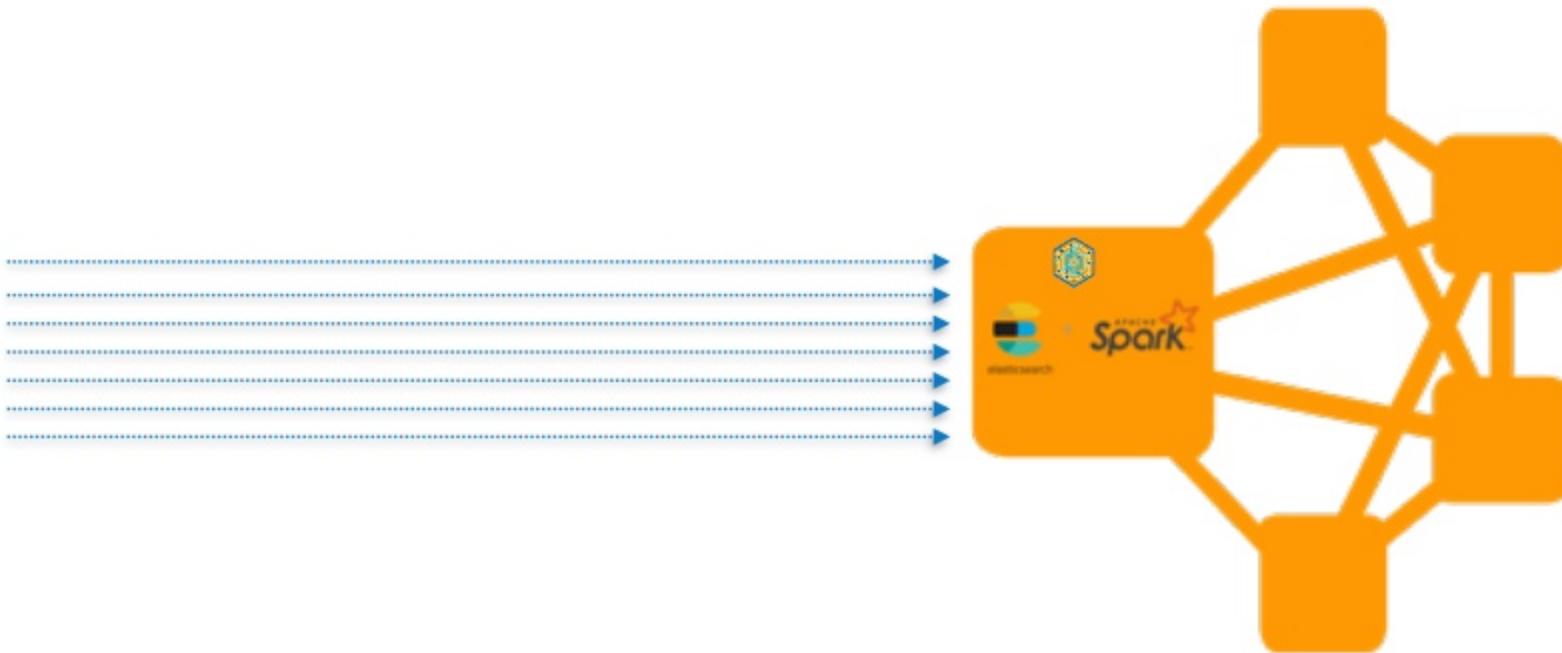
- Extract Filename, Format, Field description.
- Extract Type information.
- Index CSV files.

Rdatasets



# Overview

Data Lake



# Overview

Data Lake



# Motivation (1)

- *Organizing, indexing and ranking Datasets:*

# Motivation (1)

- *Organizing, indexing and ranking Datasets:*
  - Produced by individual data pipelines
  - On Data Lake(s)

# Motivation (2)

- Produce a *ranking function* for datasets that are generated as part of running data pipelines.

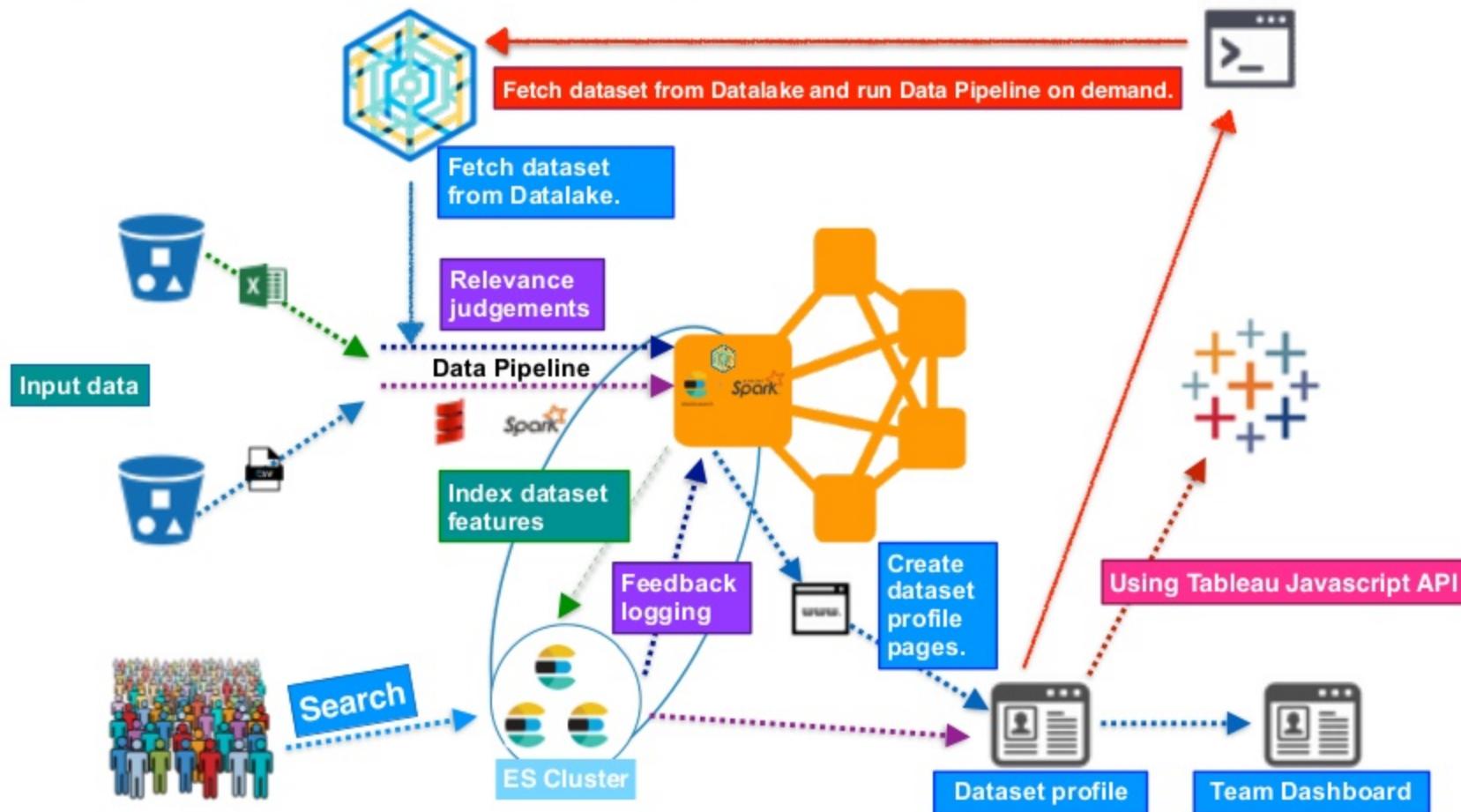
# Motivation (2)

- Produce a *ranking function* for datasets that are generated as part of running data pipelines.
- Extract “*relevance judgements*” and use them to bootstrap a dataset rank model (*a posteriori* vs. post hoc (Halevy et al., 2016)).

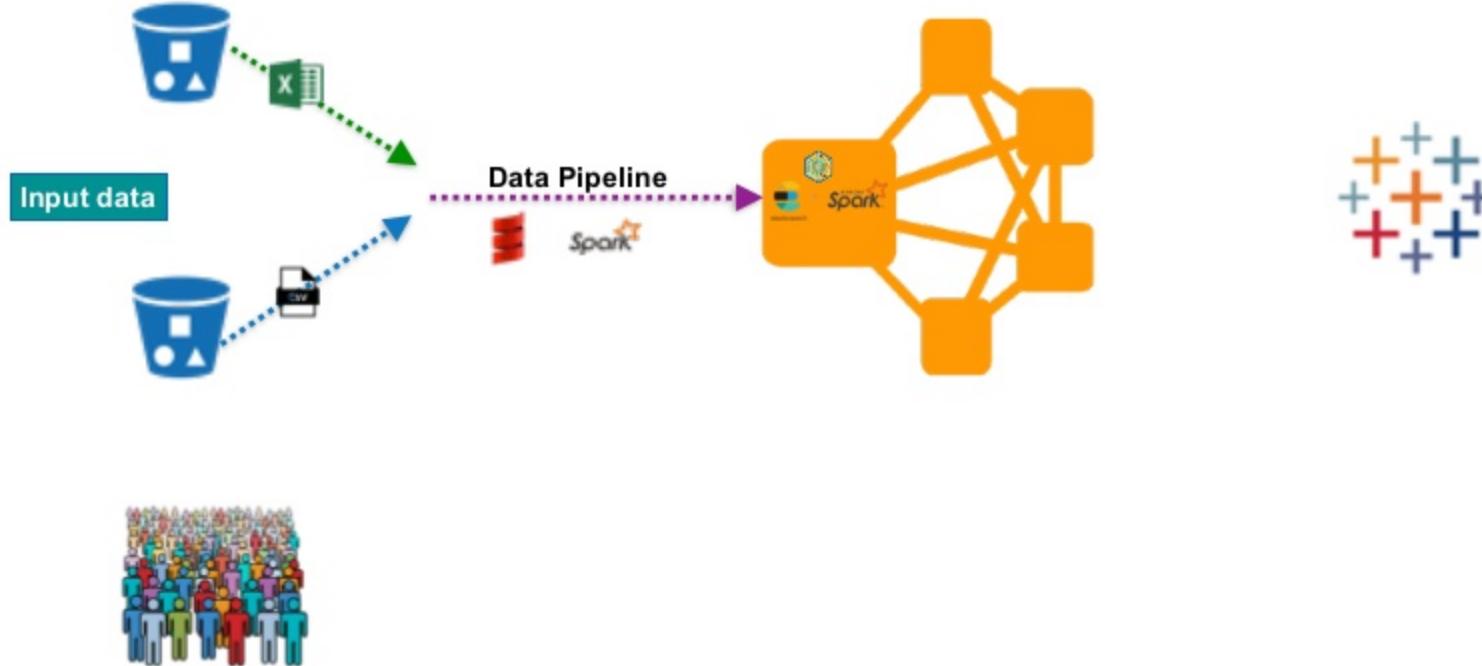
# Motivation (2)

- Produce a *ranking function* for datasets that are generated as part of running data pipelines.
- Extract “*relevance judgements*” and use them to bootstrap a dataset rank model (*a posteriori* vs. post hoc (Halevy et al., 2016)).
  - In a feedback loop leveraging click-through data on dataset profile pages.

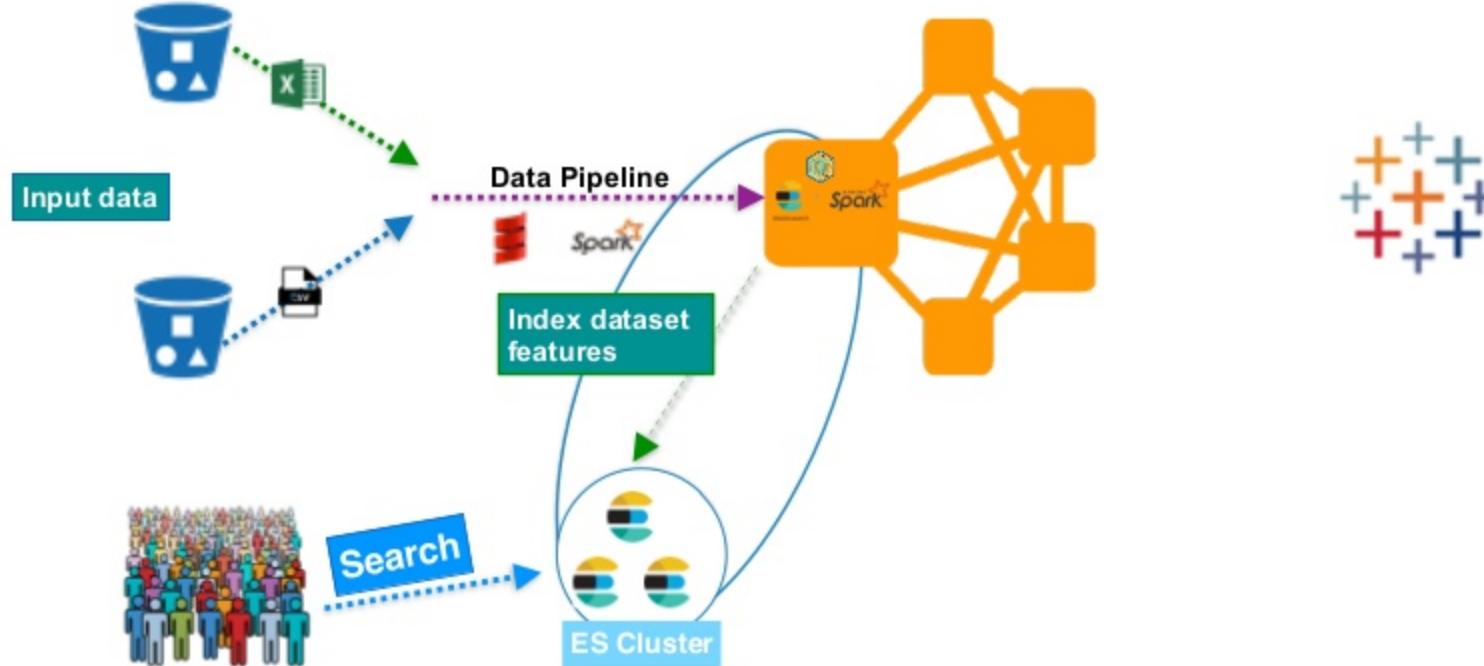
# Organizing, Indexing and Ranking Datasets



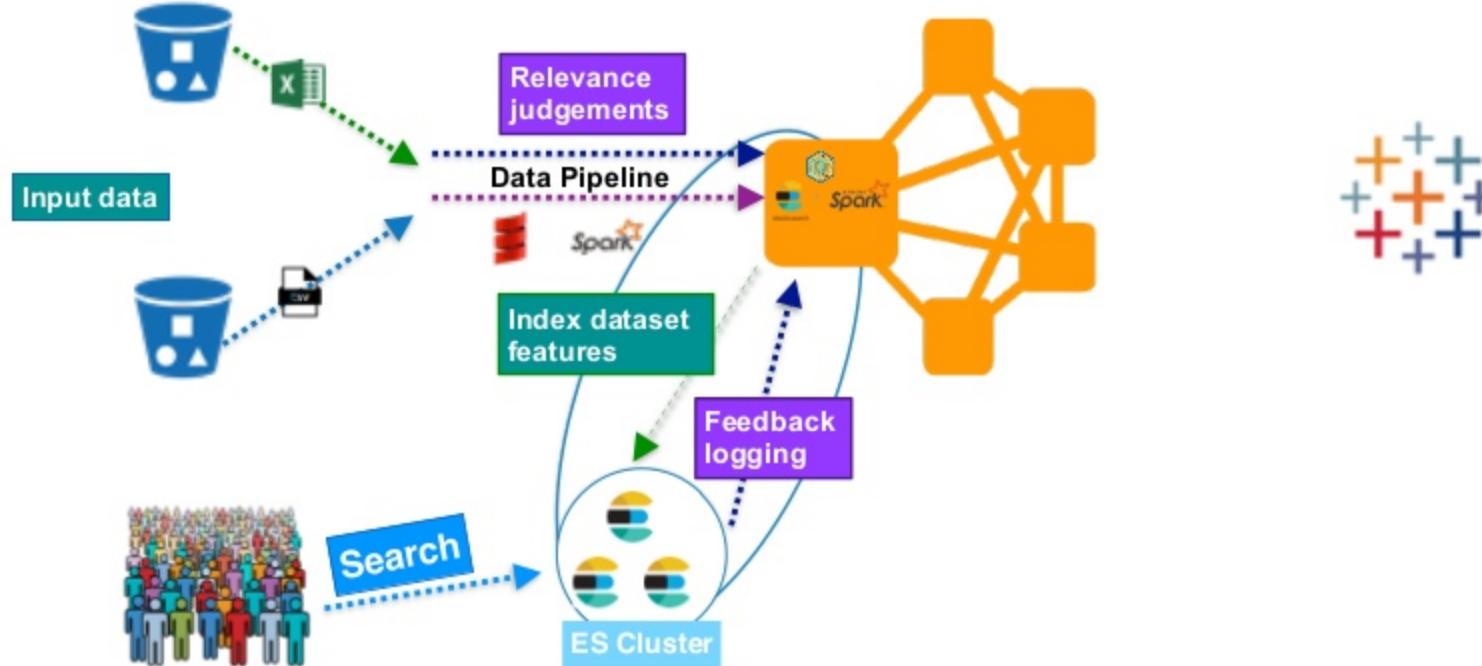
# Organizing, Indexing and Ranking Datasets



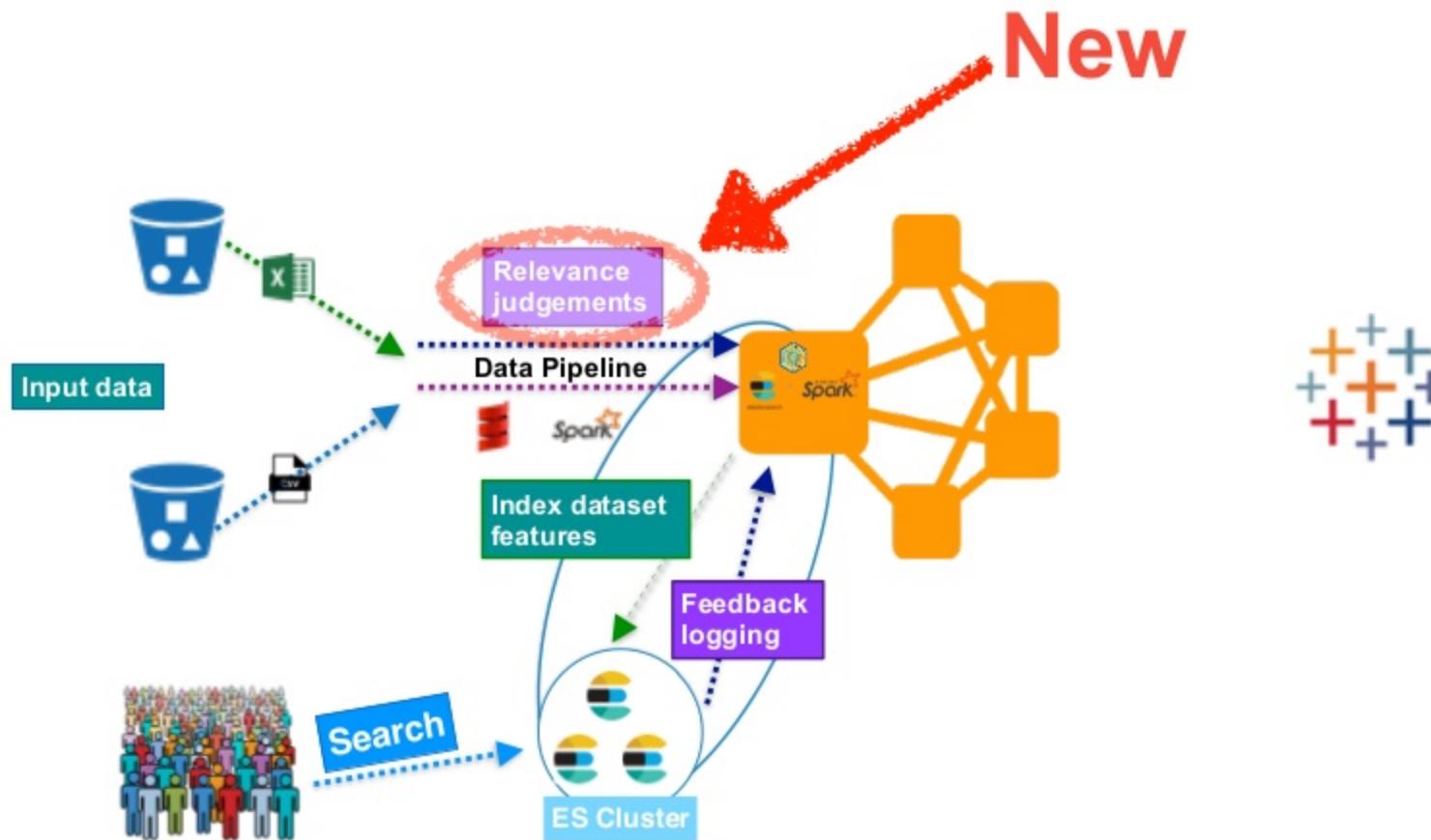
# Organizing, Indexing and Ranking Datasets



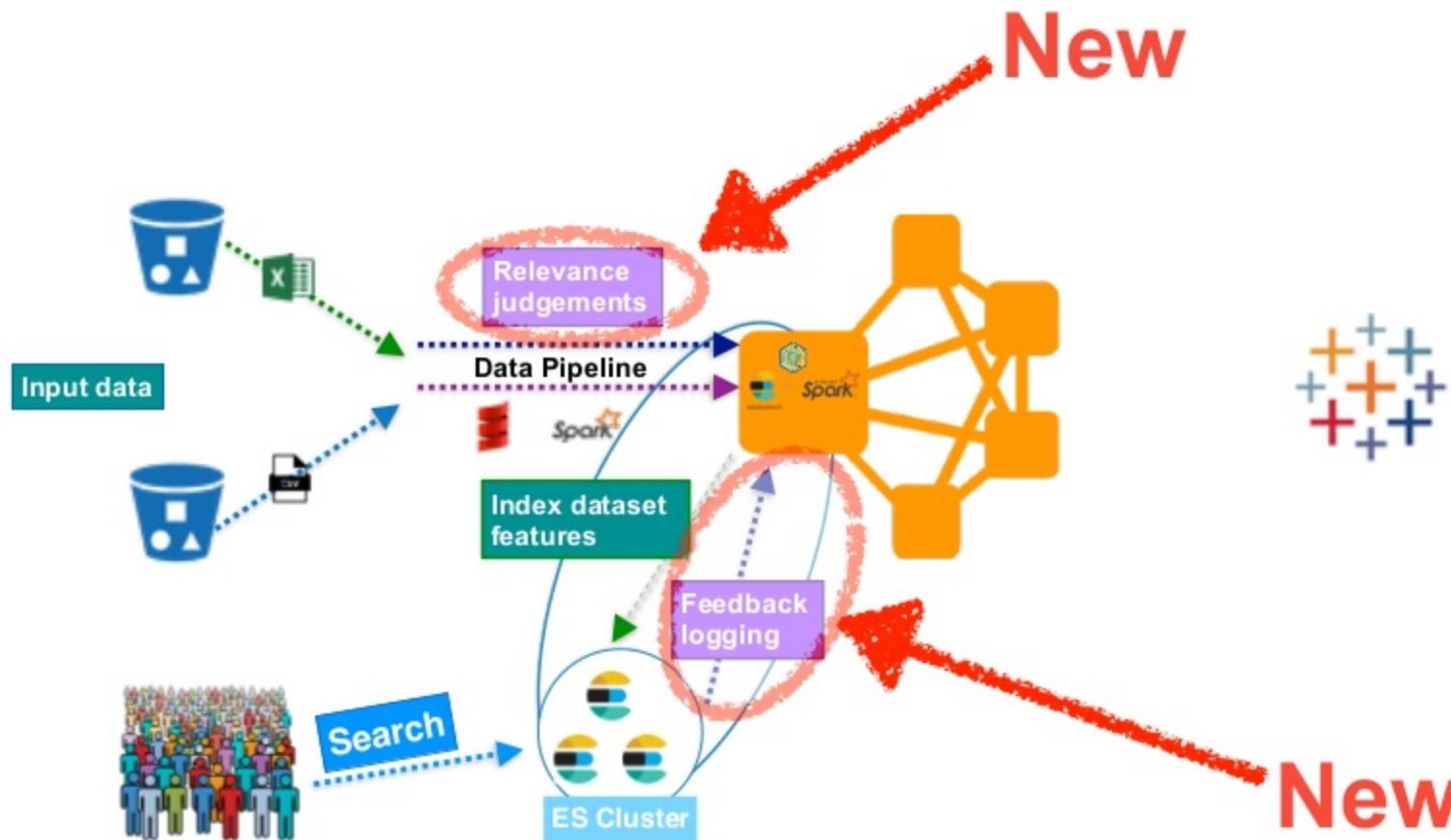
# Organizing, Indexing and Ranking Datasets



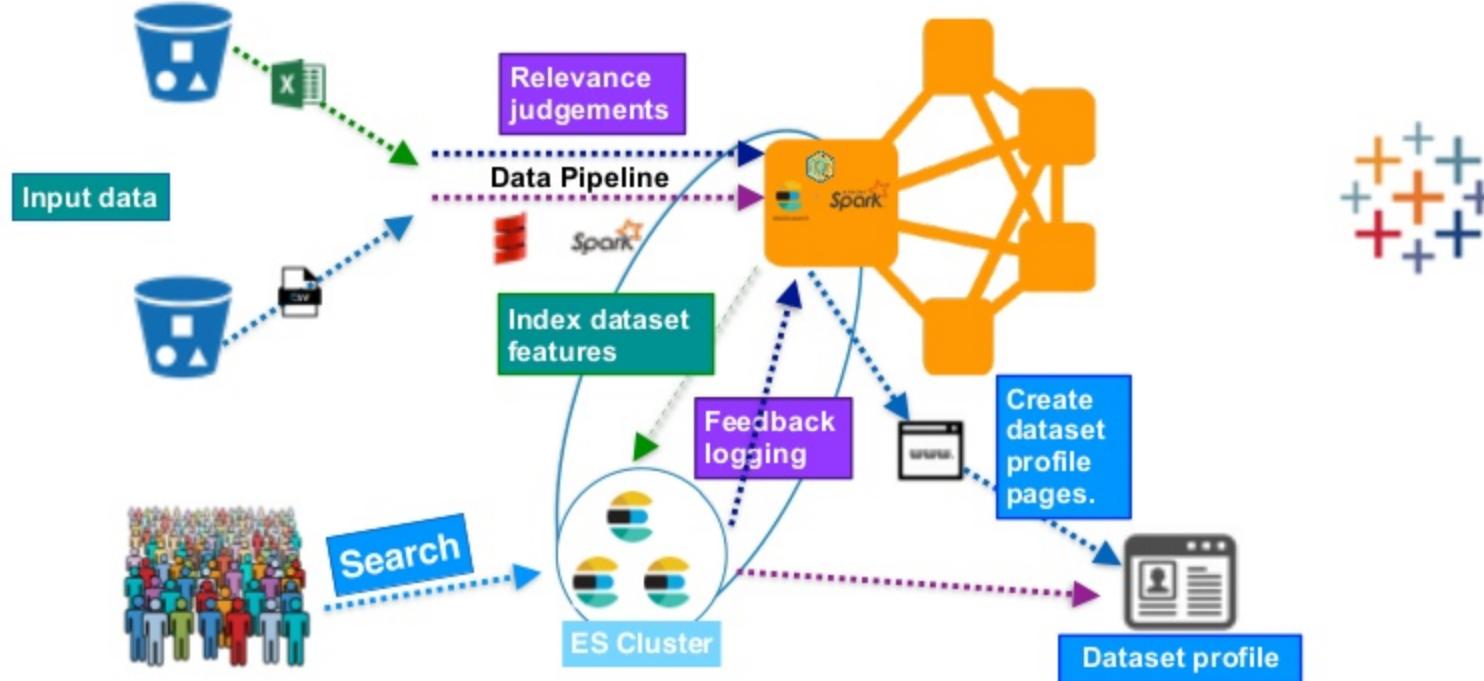
# Organizing, Indexing and Ranking Datasets



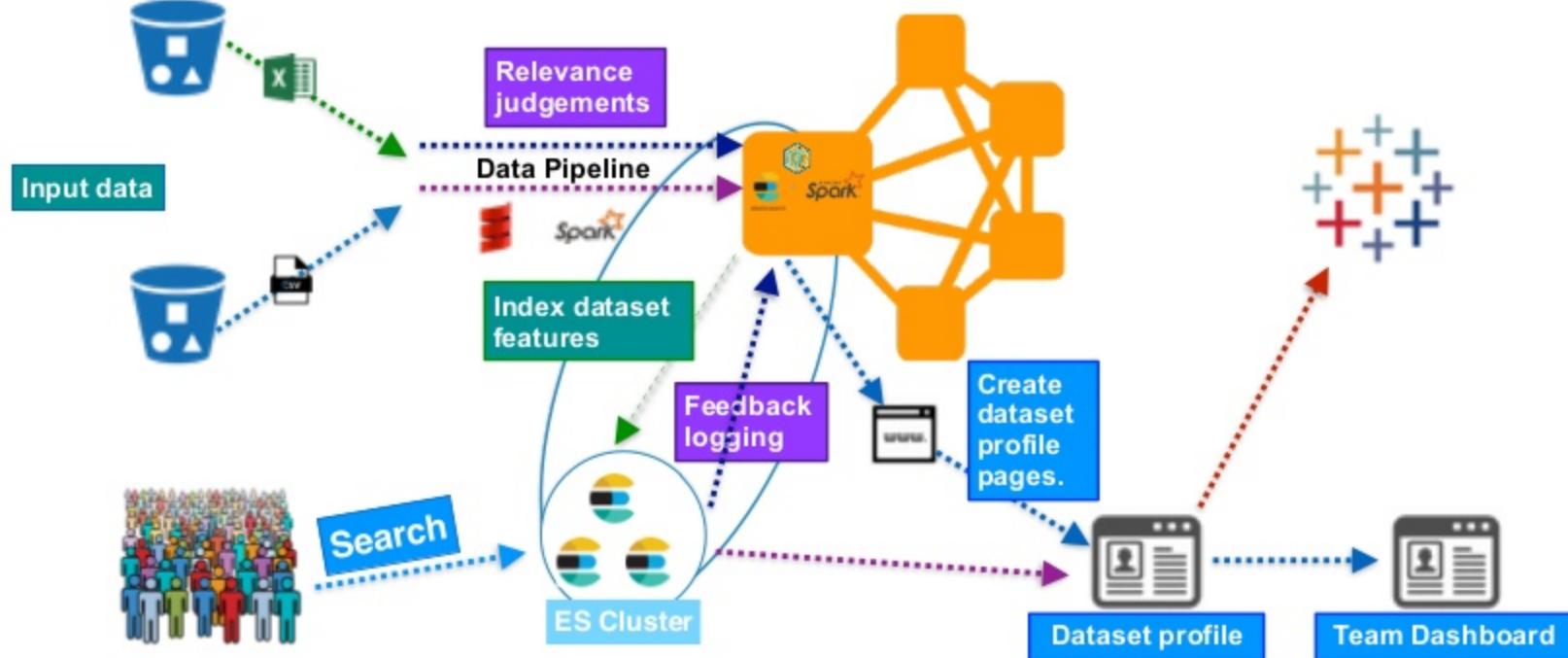
# Organizing, Indexing and Ranking Datasets



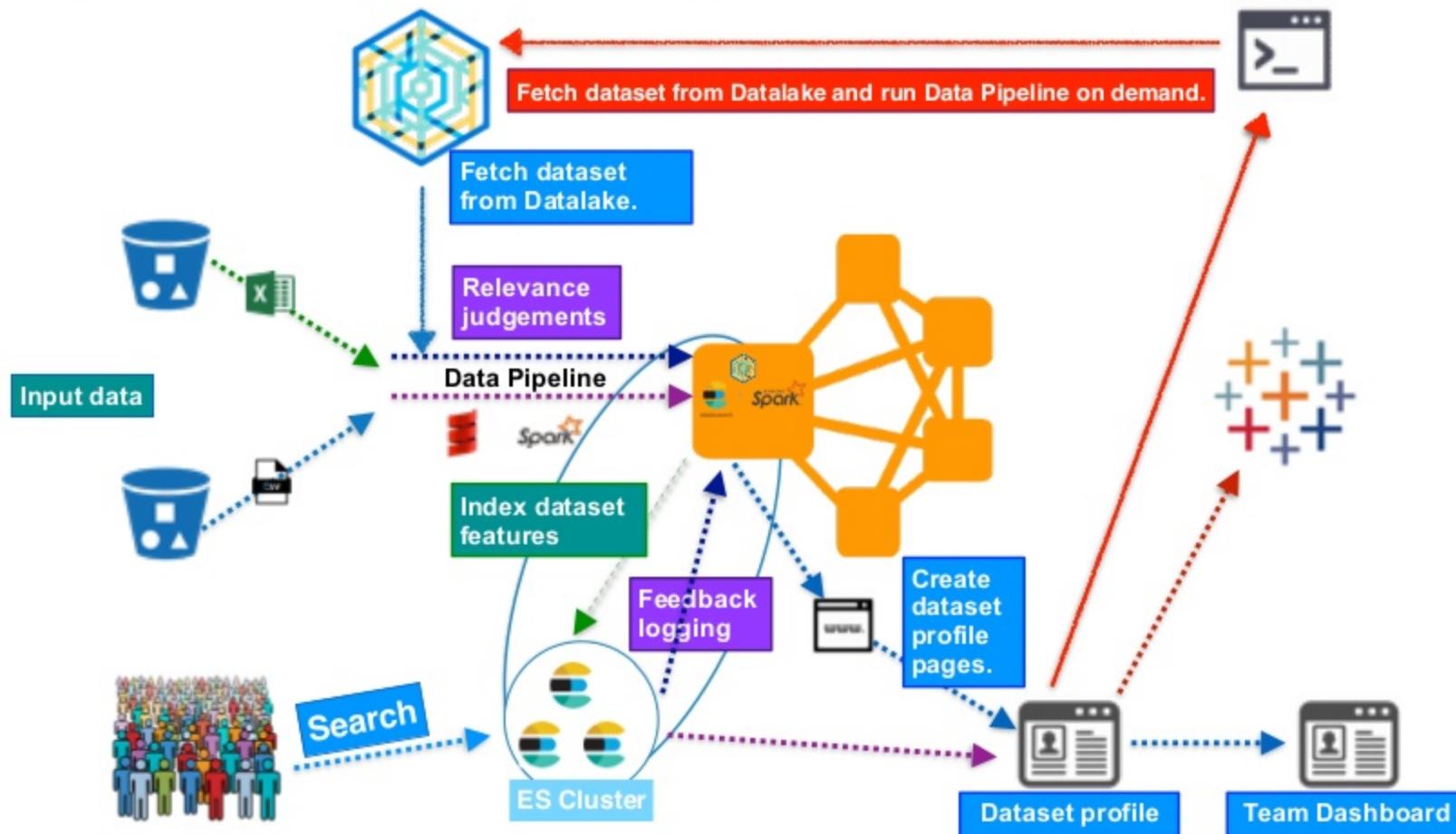
# Organizing, Indexing and Ranking Datasets



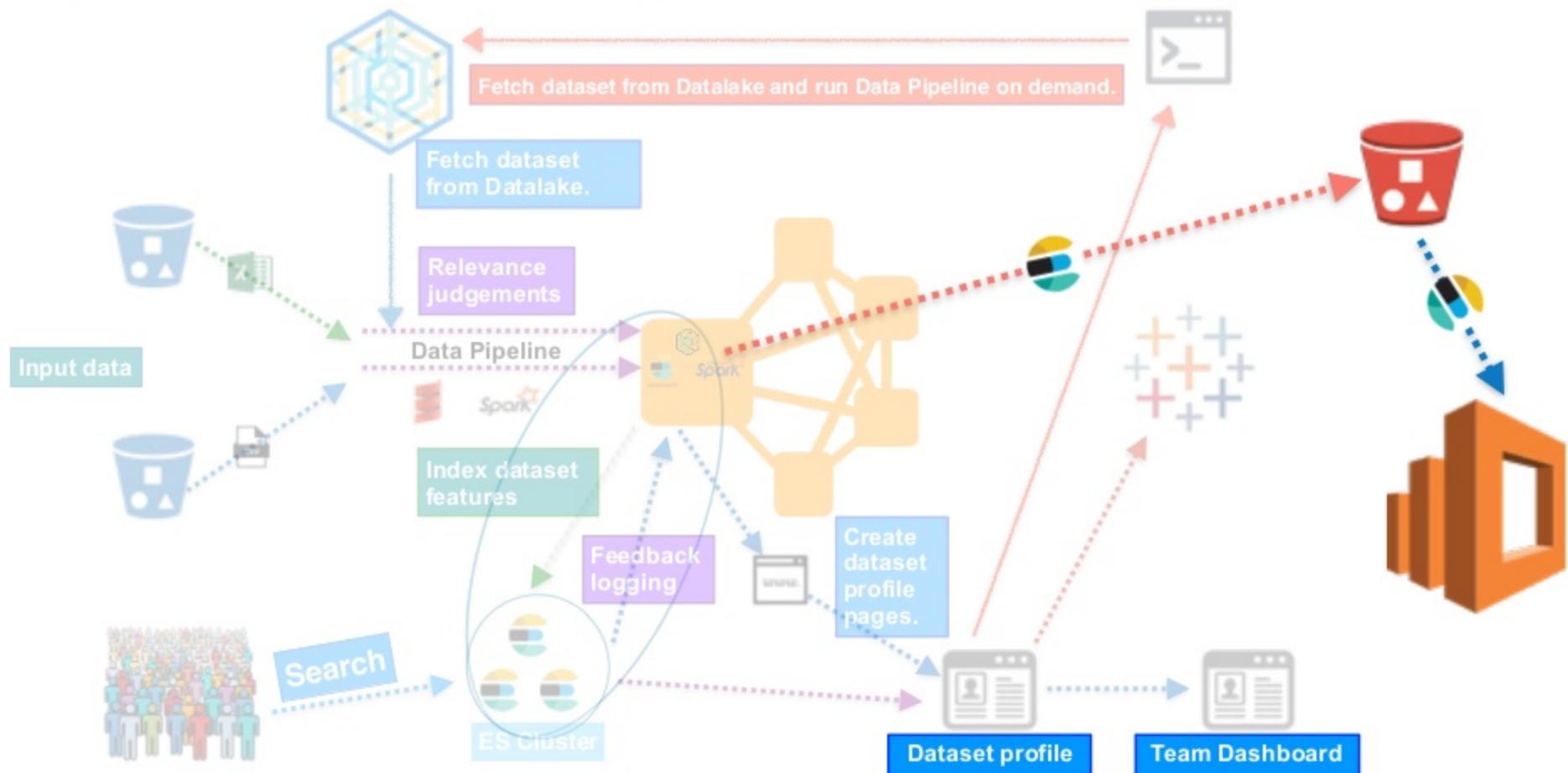
# Organizing, Indexing and Ranking Datasets



# Organizing, Indexing and Ranking Datasets



# Organizing, Indexing and Ranking Datasets



# How do you rank datasets?

# Ranking datasets

- Extraction of “*relevance judgements*” can be built into data pipeline for specific datasets immediately after they are generated.

# Ranking datasets

- Extraction of “*relevance judgements*” can be built into data pipeline for specific datasets immediately after they are generated.
  - Used to produce a *ranking function* for datasets.



# Ranking datasets

- Extraction of “*relevance judgements*” can be built into data pipeline for specific datasets immediately after they are generated.
  - Used to produce a *ranking function for datasets*.
  - Leveraged for training



# Ranking datasets

- Extraction of “*relevance judgements*” can be built into data pipeline for specific datasets immediately after they are generated.
  - Used to produce a *ranking function for datasets*.
  - Leveraged for training
  - And to bootstrap a dataset rank model



# Ranking datasets

- Extraction of “*relevance judgements*” can be built into data pipeline for specific datasets immediately after they are generated.
  - Used to produce a *ranking function for datasets*.
  - Leveraged for training
  - And to bootstrap a dataset rank model
  - (a posteriori vs. post hoc (Halevy et al., 2016).)



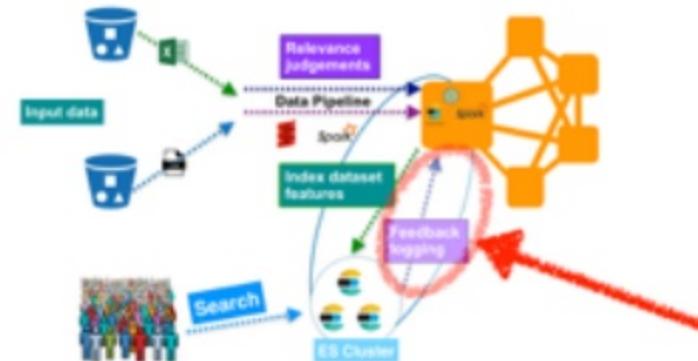
# Ranking datasets

- Click-through data provides implicit feedback useful to adjust initial relevance judgements.



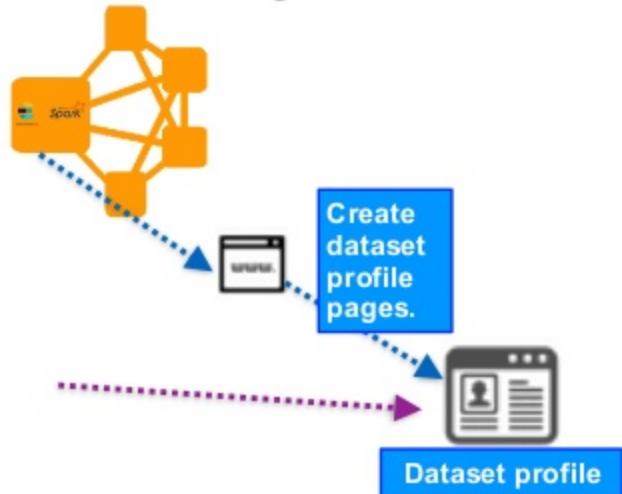
# Ranking datasets

- Click-through data provides implicit feedback useful to adjust initial relevance judgements.
  - Leveraging click-through data on **dataset profile pages**.



# Ranking datasets

- Click-through data provides implicit feedback useful to adjust initial relevance judgements.
  - Leveraging click-through data on **dataset profile pages**.



# A posteriori vs. Post-hoc

- Alon et al (2016) advocate finding data in a *post-hoc* manner by collecting and aggregating metadata after datasets are created or updated.
- We propose a so-called “*a posteriori*” approach where metadata is generated as part of running pipelines using Spark.

# A posteriori vs. Post-hoc

- Alon et al (2016) advocate indexing

after the fact

- We prefer indexing

immediately after the fact

# Pros

- “*Relevance judgements*” can be extracted and leveraged to bootstrap a ranked dataset index.
  - In a feedback loop leveraging click-through data on Dataset profile pages.
- More granular metrics available to evaluate metadata regeneration.

# Cons

- Offline model development is disconnected and only indirectly part of feedback using click-through data.
- Looking at trees instead of the forest.
- Need to replay indexing pipeline when things change (per data pipeline).

# Demo!



# Demo Scenario

- Movies represent datasets
- TMDB movie pages represent dataset profile pages.
- Marketing team (also called WAR team) interested in War movies (datasets).

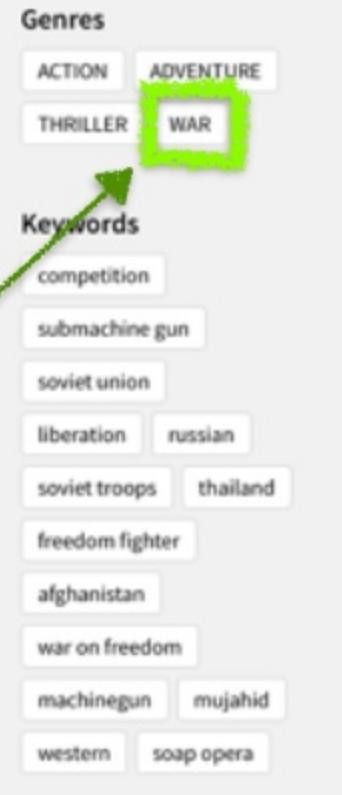
# Movie 1

<https://www.themoviedb.org/movie/7555-rambo>



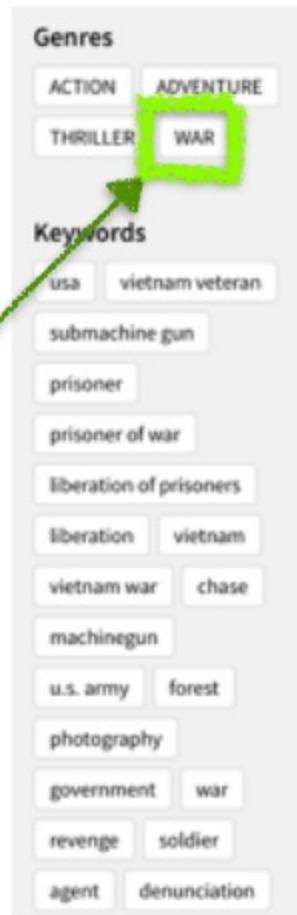
# Movie 2

<https://www.themoviedb.org/movie/1370-rambo-iii>



# Movie 3

<https://www.themoviedb.org/movie/1369-rambo-first-blood-part-ii>



# judgement file

```
# grade (0-4)    queryid docId      title
#
# Add your keyword strings below, the feature script will
# Use them to populate your query templates
#
# qid:4: Drama
# qid:5: Comedy
# qid:6: War
# qid:8: Action
#
# https://sourceforge.net/p/lemur/wiki/RankLib%20File%20Format/
#
#
0      qid:6 # 7555   ← Rambo
4      qid:6 # 1370   ← Rambo III
4      qid:6 # 1369   ← Rambo: First Blood Part II
```

# What have we seen?

- How to rank datasets on Elasticsearch using LTR.
- Extract relevance judgements immediately after datasets are generated in Spark.
- Demo: Dataset Search with Spark and Elasticsearch LTR.

# Next Steps (1)

- **Describe Datasets in a structured [schema.org way](#)** using Data Catalog Vocabulary [2].
- **Build a knowledge graph** and use GraphX to extract insights. (Useful e.g. for column concept determination (Deng et al. 2013)).
- **Build topic models based on structured Datasets** using Glint to perform scalable topic model extraction in Spark (Jagerman and Eickhoff, 2016) [1].

[1] <https://spark-summit.org/eu-2016/events/glint-an-asynchronous-parameter-server-for-spark/>

# References

- Alon Y. Halevy, Flip Korn, Natalya Fridman Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. Goods: Organizing google's datasets. In Fatmañzcan, Georgia Koutrika, and Sam Madden, editors, Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016, pages 795–806. ACM, 2016. ISBN 978-1-4503-3531-7. doi: <http://doi.acm.org/10.1145/2882903.2903730>.
- Katja Hofmann. Fast and Reliable Online Learning to Rank for Information Retrieval. PhD thesis, Informatics Institute, University of Amsterdam, May 2013.
- Rolf Jagerman and Carsten Eickhoff. Web-scale topic models in spark: An asynchronous parameter server. CoRR, abs/1605.07422, 2016. URL <http://arxiv.org/abs/1605.07422>.
- Dong Deng, Yu Jiang, Guoliang Li, Jian Li, and Cong Yu. Scalable column concept de- termination for web tables using large knowledge bases. PVLDB, 6(13):1606–1617, 2013. doi: <http://www.vldb.org/pvldb/vol6/p1606-li.pdf>.
- Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. Multileave gradient descent for fast online learning to rank. In *WSDM 2016: The 9th International Conference on Web Search and Data Mining*, pages 457-466. ACM, February 2016.
- Sreeram Balakrishnan, Alon Y. Halevy, Boulos Harb, Hongrae Lee, Jayant Madhavan, Afshin Rostamizadeh, Warren Shen, Kenneth Wilder, Fei Wu 0003, and Cong Yu. Ap- plying webtables in practice. In CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings. www.cidrdb.org, 2015.

# Q&A

# Thank You.

Email: [ocastaneda@paypal.com](mailto:ocastaneda@paypal.com)

Twitter: [@oscar\\_castaneda](https://twitter.com/@oscar_castaneda)