



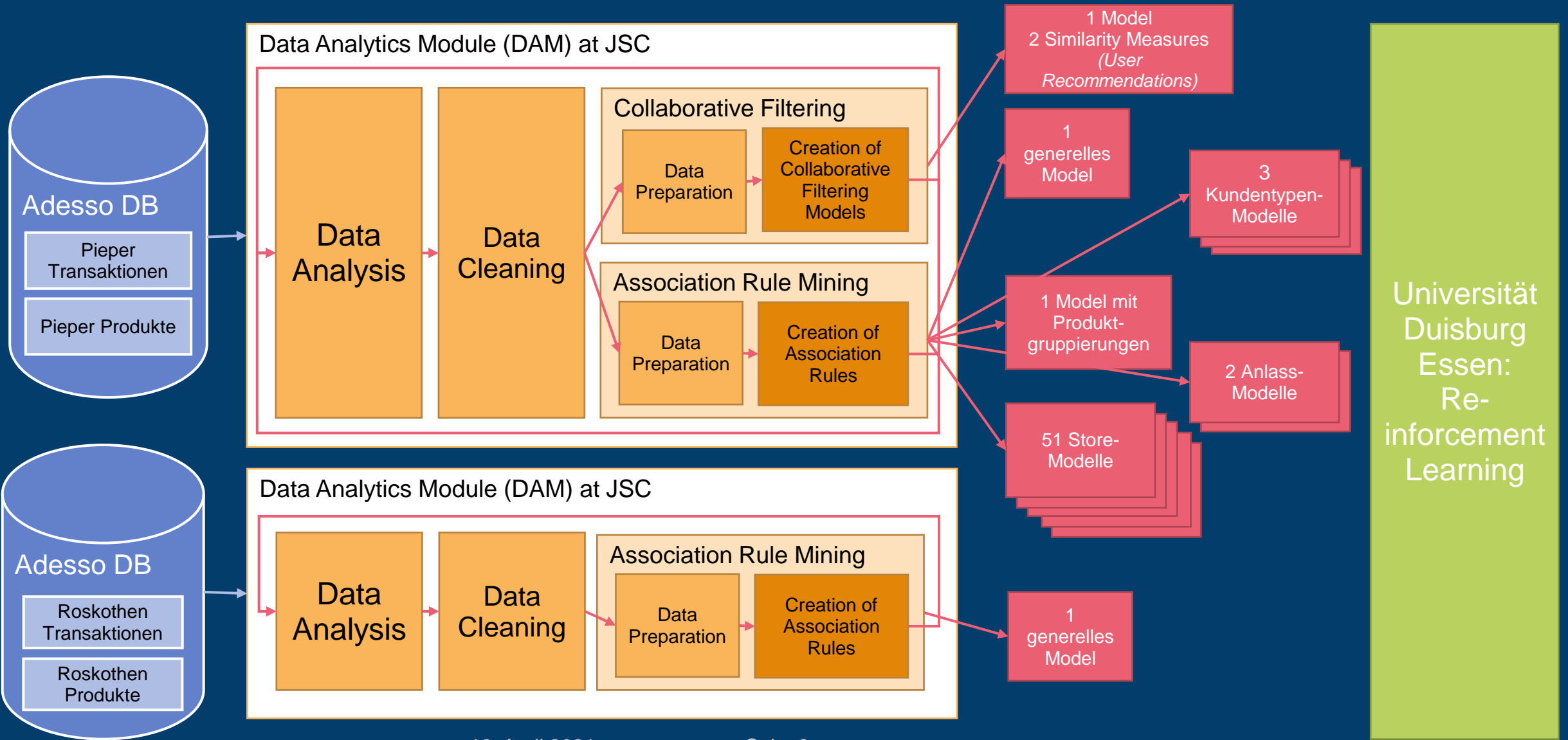
ON4OFF

KONSORTIALMEETING - PROGRESS

19.04.2021 | BARAKAT, CHADI

| BUSCH, JOSEFINE SOPHIE

RECOMMENDER MODELLE – AKTUELLER STAND



Universität
Duisburg
Essen:
Re-
inforcement
Learning

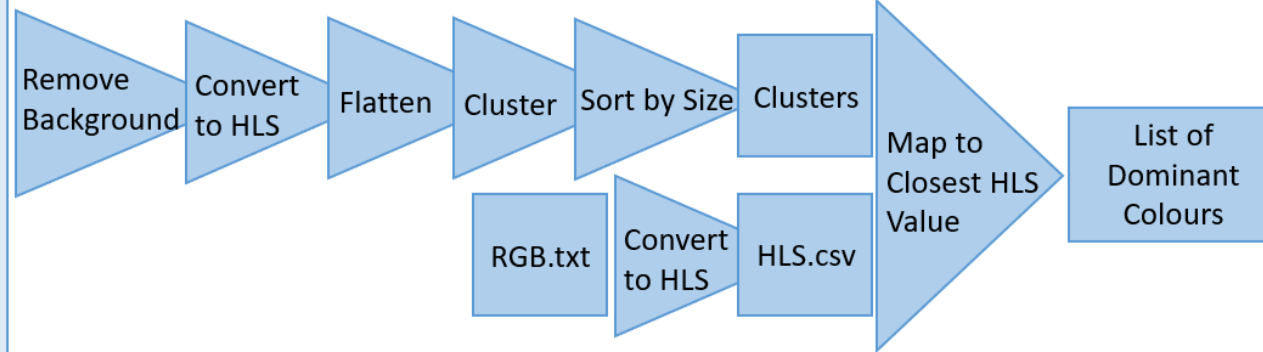
Adesso
Database

Data Analytics Module (DAM) at JSC

Adesso
Services

Image Tagging Service

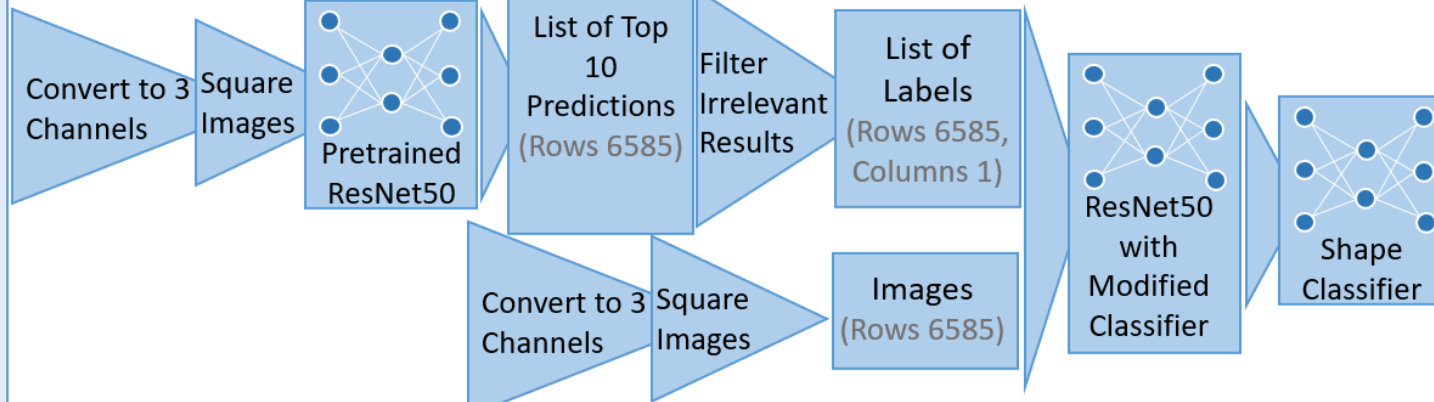
Colour Detection Module



Images
(Rows 6585)

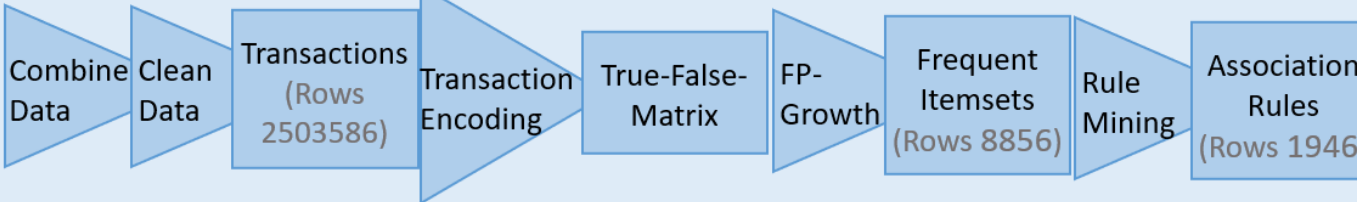
Colour
Tagging
Interface

Shape Detection Module



Shape
Tagging
Interface

Recommender Service



Transactions
(Rows
8139215)

Products
(Rows
8139215)

Recommender
Interface

KUNDENBEZOGENE EMPFEHLUNGEN

Auf Basis von Collaborative Filtering

```
localhost:5010/cf_recommend_cusotmer?clientID=pieper&customerNo=206075
```

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

ids:

- 0: "65282"
- 1: "29685"
- 2: "968"
- 3: "8016"
- 4: "182116"
- 5: "60880"
- 6: "176370"
- 7: "58948"
- 8: "19410"
- 9: "2501"
- 10: "8163"
- 11: "92075"
- 12: "46546"
- 13: "14597"
- 14: "49105"
- 15: "33633"
- 16: "41813"
- 17: "7411"
- 18: "60925"
- 19: "1808"
- 20: "30232"

RITUALS
The Ritual of Sakura
Körperpeeling

€ 14,90
€ 5,96 / 100 g

inkl. MwSt
zzgl. Versand

IN DEN WARENKORB

[1]

ARTDECO
Collection BEAUTY MEETS FASHION ...
Perfect Mat Lipstick

ab € 10,70
€ 267,50 / 100 g

inkl. MwSt
zzgl. Versand

VARIANTE WÄHLEN

[2]

[1] <https://www.pieper.de/search?sSearch=body+scrub+sakura>
[2] <https://www.pieper.de/search?sSearch=artdeco%2Bperfect%2Bmat&p=1>

HINDERNISSE DURCH BIG DATA

- 100.000 Belege benutzt als Testdaten:
 - One-Hot-Encoding-Matrix (für cross-matching) hat mehr als 250 MB
- Gesamter Datensatz hat 5 Millionen Belege
 - über 200.000 einzigartige Produkte
 - mehr als 100.000 Kunden IDs
 - Die Matrix würde exponentiell größer sein
 - Nicht möglich mit dieser Menge

```
In [16]: indices.shape
Out[16]: (12929, 2)

In [14]: tf.Session().run(sparse_tensor)
Out[14]: SparseTensorValue(indices=array([[ 0,  0],
      [ 1,  1],
      [ 2,  1],
      ...,
      [3537, 5862],
      [5625, 1588],
      [ 324, 5863]]), values=array([1, 1, 1, ..., 1, 1, 1]), dense_shape=array([5626, 5864]))

In [19]: test = pd.DataFrame(columns=range(5626), index=range(5864))

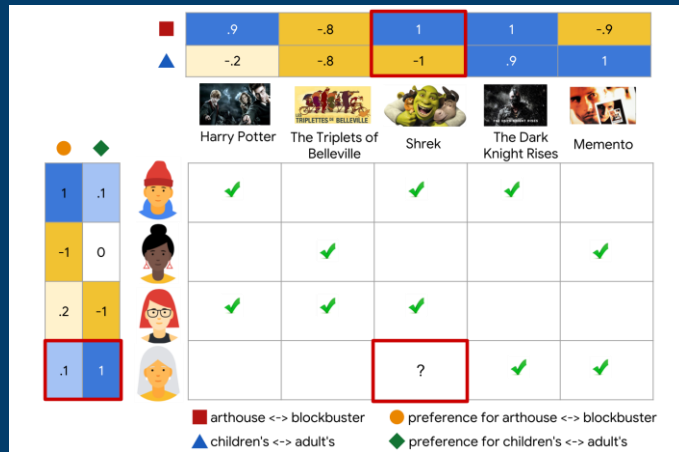
In [30]: print(test.memory_usage().sum()/1024/1024, "MB")
251.700439453125 MB

In [21]: test2 = pd.DataFrame(columns=range(12929), index=range(2))

In [28]: print(test2.memory_usage().sum()/1024/1024, "MB")
0.1974029541015625 MB
```


COLLABORATIVE FILTERING

Wie es mit unseren Daten funktioniert



```

# Build the CF model and train it.
model = build_model(ratings, embedding_dim=10, init_stddev=0.5)
model.train(num_iterations=1000, learning_rate=10.)
    
```

