Modelling Fashion @ wehkamp

About wehkamp

About Wehkamp

1952 - founded by Herman Wehkamp

2006 - transition to online

2010 - all sales through Digital Channels

Facts
- 180.000 products
- 1.850 different brands
- Largest automated Warehouse in Europe (Zwolle, The Netherlands)
- Same Day Delivery at large scale
- Content authority with Vloggers
- And much more...

Largest online Department Store in NL

Digital Development at Wehkamp

Approx 80 FTE engineers

Agile Teams own the Frontend Ecosystem

Customer Facing Technology Stack

- Innovation, full stack development
- Running operations (*DevOps/SRE*)
- Microservices at a Large Scale (*from parts to a whole*)
- Data Engineering capability
- Open Source, Scala, Java, Akka, Kafka
- Visibility in the Community
- And much more...

We love Technology and Reliable Propagation of Change

Innovation is in our DNA

Problem statement

IBM Coremetrics

Strategy

Technology Strategy

Make for competitive advantage

⇒ Roll our own Recommendations

Buy commodity functionalities

⇒ Google Analytics Premium for analytics

Recommender Item
item

Collaborative Filtering

Movie ratings

	Amy	Jef	Mike	Chris	Ken
The Piano	–	–	+		+
Pulp Fiction	–	+	+	–	+
Clueless	+		–	+	–
Cliffhanger	–	–	+	–	+
Fargo	–	+	+	–	?

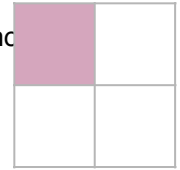
Co-occurrence

Item Item recommendation

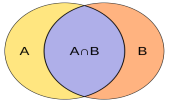
Score other items based on (non) co-occurrence

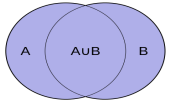
- Raw co-occurrence

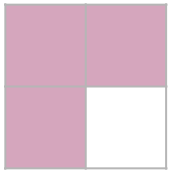
  recommend item that co-occurs mo



- Jaccard

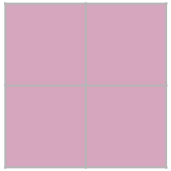






- Log likelihood ratio

  recommend anomalous co-occurrence;
  suppress popular items



			$\sum_{row}$
	12	73	85
	51	5334	5385
$\sum_{column}$	63	5407	5470

Evaluation

Mean Reciprocal Rank

First item in Session S
(Item_{S1})

1

2

3
Item_{S2}

4

5

Score for session S $\frac{1}{3}$

Total score $$\text{MRR} = \frac{1}{|\text{sessions}|} \sum_{S \in \text{sessions}} \frac{1}{\text{rank}(\text{Item}_{S2} | \text{Item}_{S1})}$$

Recommender - Compute

Collect events

- Custom definable events
- Writes Avro to HDFS
  no log file parsing
- Kafka
- In flight IP2geo lookup
- Scriptable (groovy)

Tag - send event

```
  <script src="//divolte-nl.wehkamp.com/divolte.js"></script>
  <script>
    divolte.signal("pageView", {"registrationId": "12345678"});
  </script>
</body>
```

Mapping - convert to avro

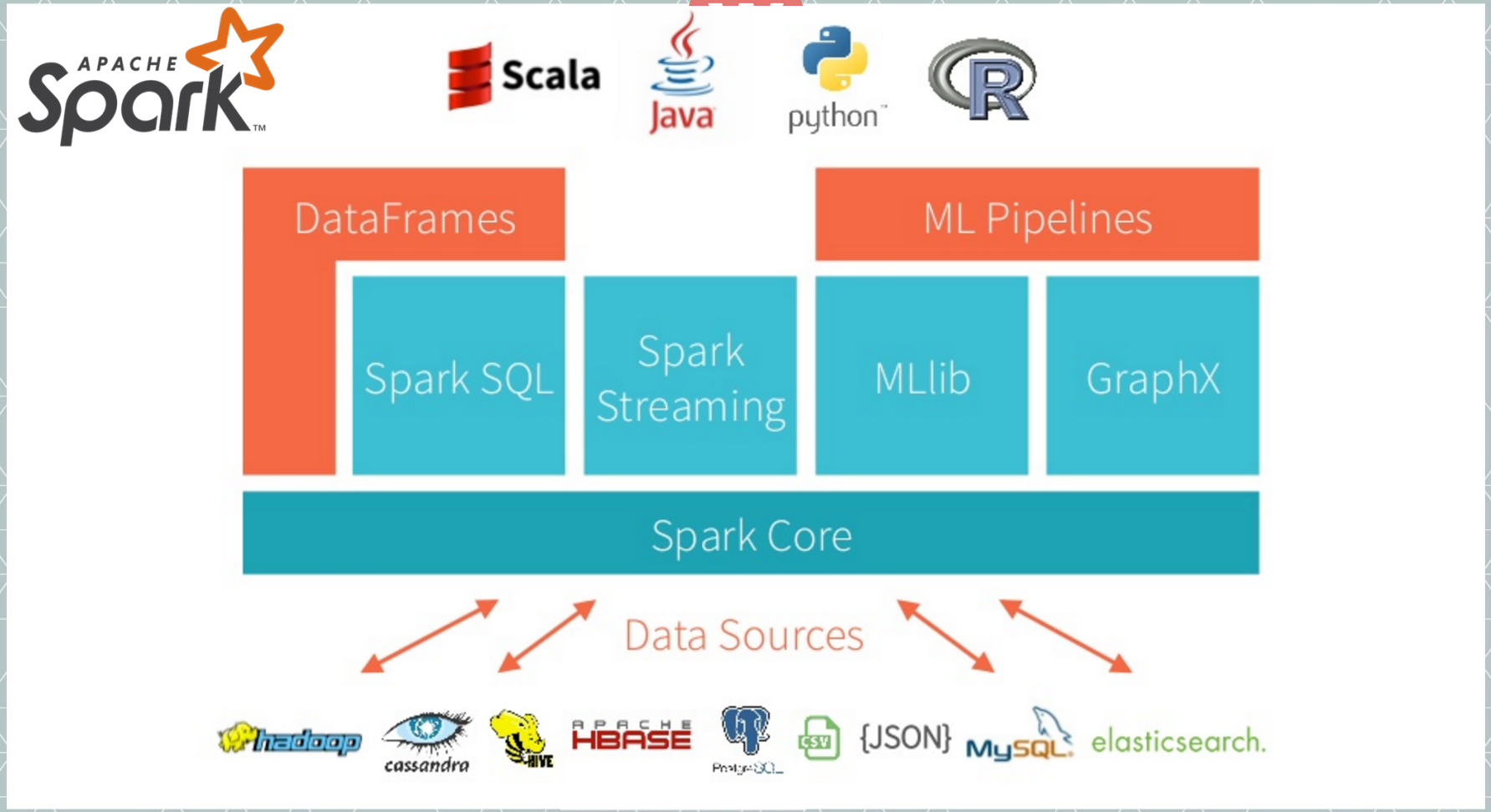
```
mapping {
  map clientTimestamp() onto 'timestamp'
  map location() onto 'location'

  def u = parse location() to uri
  section {
    when u.path().equalTo('/checkout') apply {
      map 'checkout' onto 'pageType'
      exit()
    }
    map 'normal' onto 'pageType'
  }
}
```

http://divolte.io/

Compute


```
val textFile = sc.textFile("hdfs://...")
val counts = textFile.flatMap(line => line.split(" "))
                 .map(word => (word, 1))
                 .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
```

cluster computing framework


APACHE Spark
Scala
Java
python
R
DataFrames
ML Pipelines
Spark SQL
Spark Streaming
MLlib
GraphX
Spark Core
Data Sources
hadoop
cassandra
HIVE
APACHE HBASE
PostgreSQL
CSV
{JSON}
MySQL
elasticsearch.

Airflow

Airflow

workflow management platform

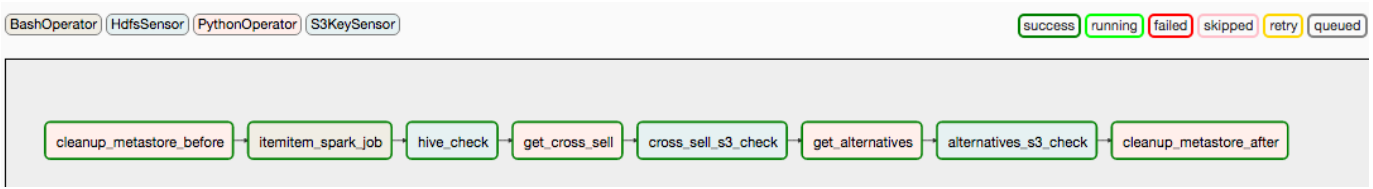
- Scheduling
- Data pipelines (DAG)

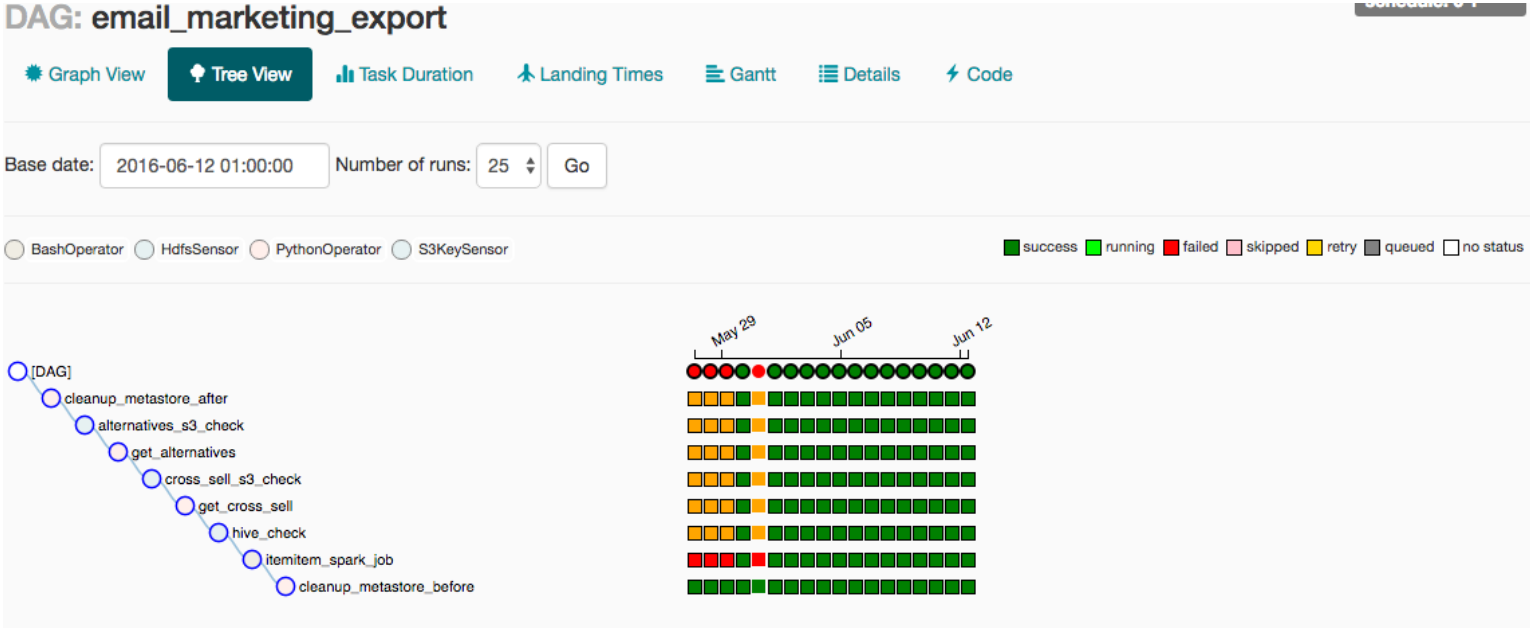
Dag definition (python)

```
dag = DAG('my_dag', start_date=datetime(2016, 1, 1))

# sets the DAG explicitly
explicit_op = DummyOperator(task_id='op1', dag=dag)

# deferred DAG assignment
deferred_op = DummyOperator(task_id='op2')
deferred_op.dag = dag

# inferred DAG assignment
inferred_op = DummyOperator(task_id='op3')
inferred_op.set_upstream(deferred_op)
```

http://airflow.apache.org/

Airflow


BashOperator
HdfsSensor
PythonOperator
S3KeySensor
success
running
failed
skipped
retry
queued
cleanup_metastore_before
itemitem_spark_job
hive_check
get_cross_sell
cross_sell_s3_check
get_alternatives
alternatives_s3_check
cleanup_metastore_after


DAG: email_marketing_export
Graph View
Tree View
Task Duration
Landing Times
Gantt
Details
Code
Base date: 2016-06-12 01:00:00
Number of runs: 25
Go
BashOperator
HdfsSensor
PythonOperator
S3KeySensor
success
running
failed
skipped
retry
queued
no status
May 29
Jun 05
Jun 12
[DAG]
cleanup_metastore_after
alternatives_s3_check
get_alternatives
cross_sell_s3_check
get_cross_sell
hive_check
itemitem_spark_job
cleanup_metastore_before

Airflow

Operators

```
itemitem_spark_job = BashOperator(
        task_id='itemitem_spark_job',
        bash_command="""spark-submit \
  --master yarn-cluster \
  --driver-memory 4g \
  /artifacts/itemitem-assembly.jar \
  --algorithm {{ params.algorithm }} \
  --number_of_recommendations {{ params.nr_recommendations }} \
  ...
  --cassandraKeyspace {{ params.cassandra_keyspace }} \
  --cassandraTable {{ params.cassandra_table }} \
  --saveToCassandra
""",
        params=SPARK_PARAMS,
        dag=dag)
```

Hooks

```
s3 = S3Hook(S3_CONN_ID)
s3.load_file(
    filename=LOCALTMP + finalname,
    key='sri/' + finalname,
    bucket_name=cfg.s3_bucket['cdw_exchange'])
```

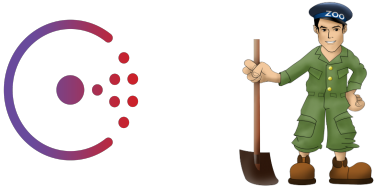
Sensors

```
wait_for_output = HdfsSensor(
    task_id="wait_for_output",
    filepath="sri-{{ tomorrow_ds_nodash }}/
_SUCCESS",
    dag=dag)
```

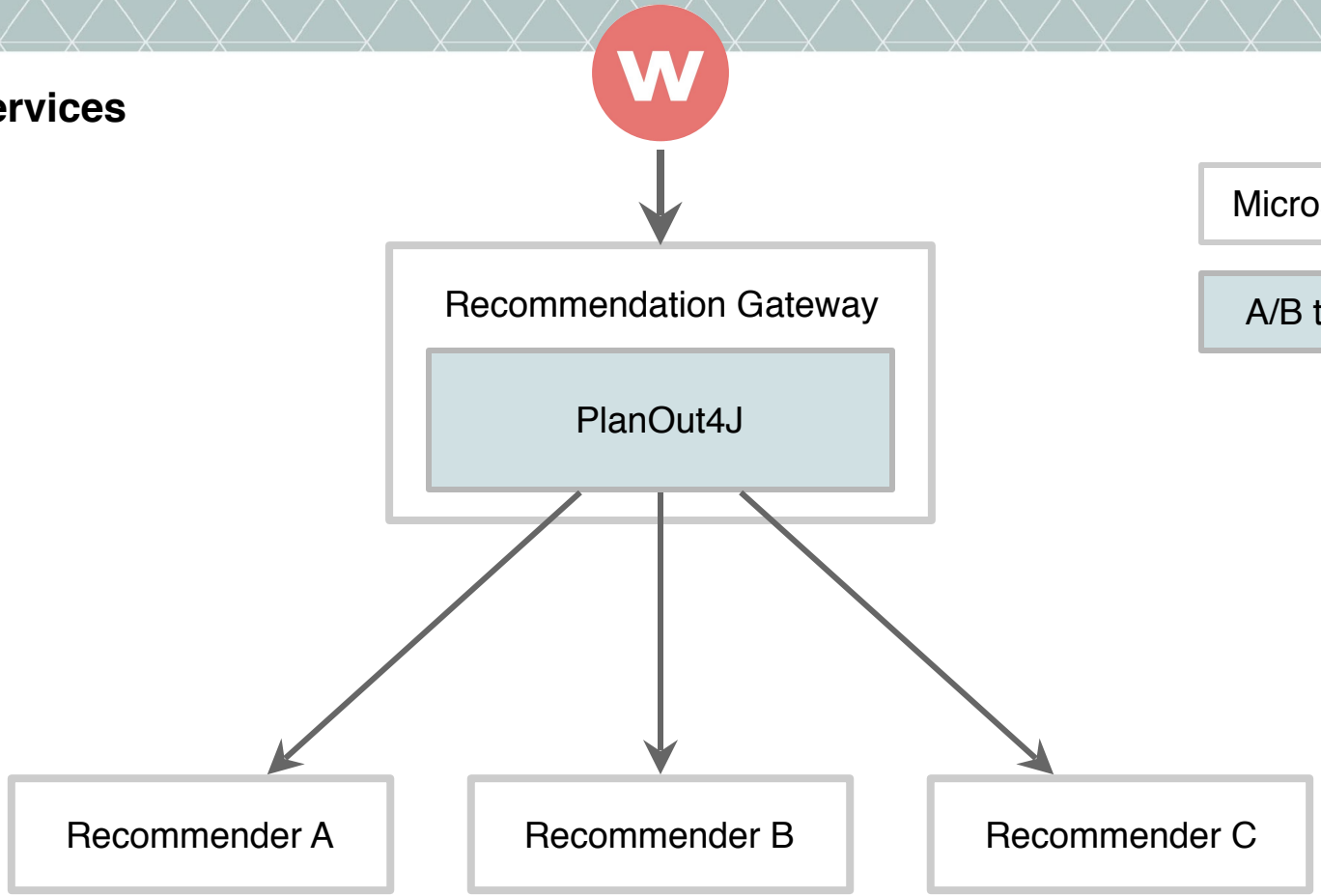
Recommender - Serve

Serve - Microservices

- Reactive Microservices architecture
- Scalable & Resilient Infrastructure
- Blend of SaaS & Wehkamp proprietary services
- Services expose REST API's over HTTP/JSON
- Channel Apps consume API's
- Open for integration, internally and externally
- Support for Multi-instances e.g, countries



Microservices



Storage - NoSQL

- Fault-tolerant
- Scalable
- Flexible read/write performance tuning

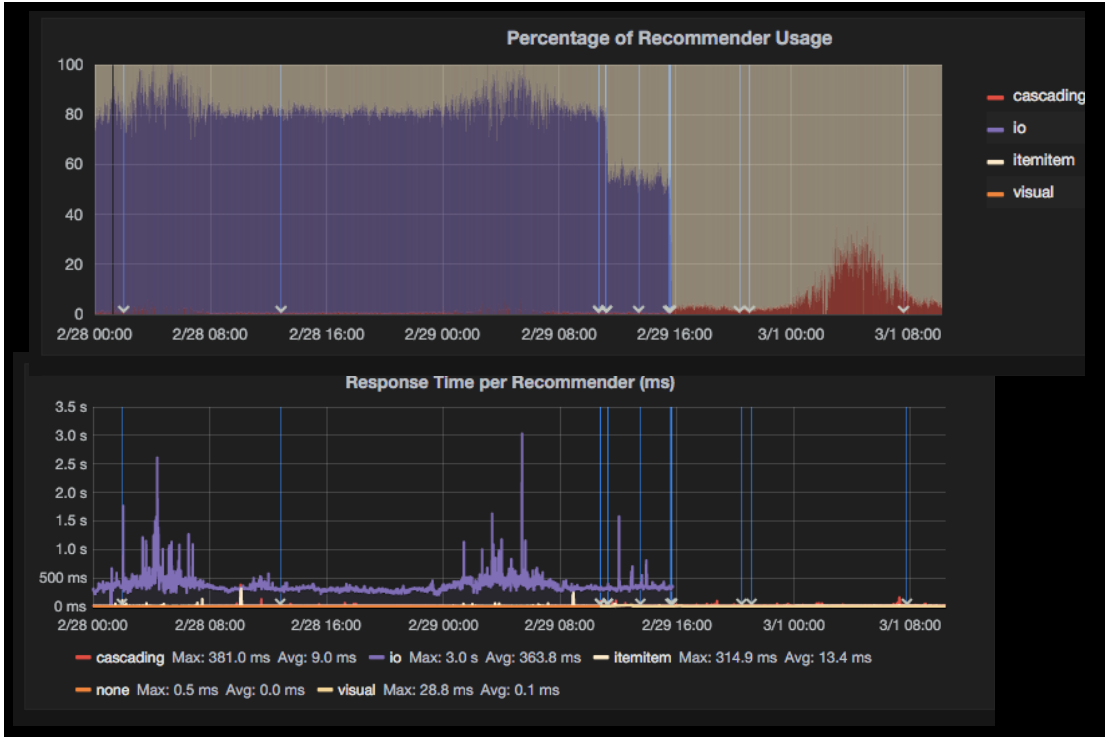
```
CREATE TABLE itemitem (
  product_id TEXT,
  rank INT,
  distance_score DOUBLE,
  related_product_id TEXT,
   ...
  PRIMARY KEY (product_id, rank)
) WITH CLUSTERING ORDER BY (rank ASC)
```

Partition Key

Top 5

```
SELECT distance_score, related_product_id
FROM itemitem WHERE product_id = '$productId' LIMIT 5;
```

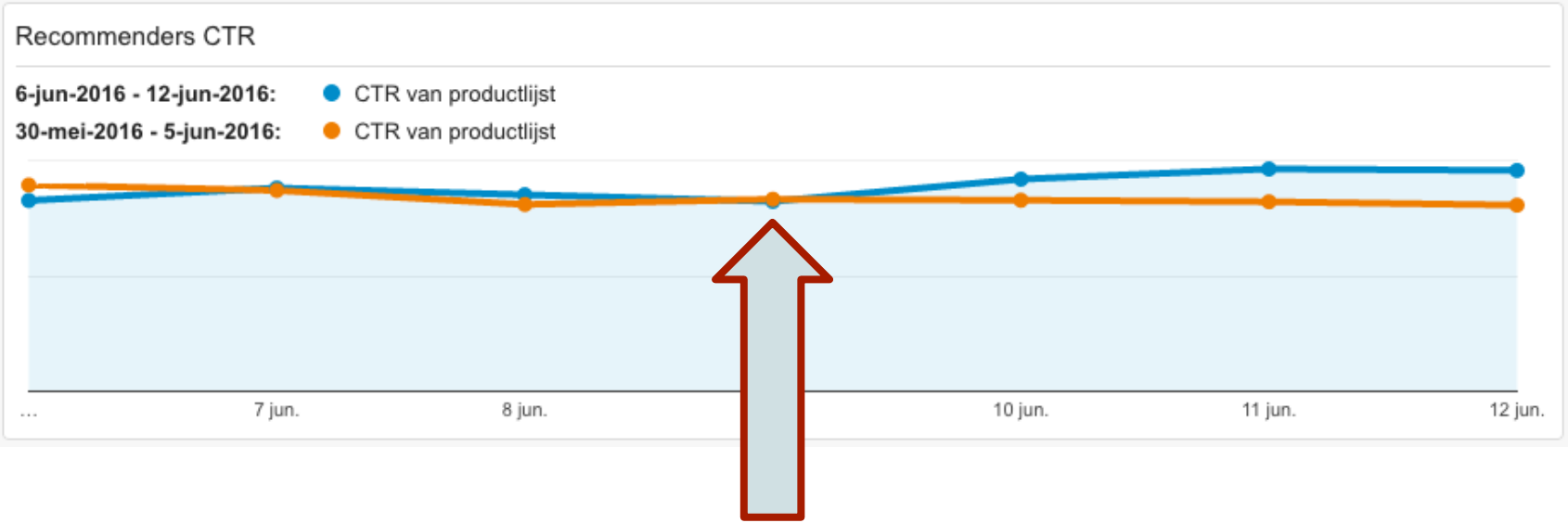
Exit Intelligent Offer



Exit Intelligent Offer

- Conversion improved
- Response times much better
- Controlled roll-out
  A/B testing infrastructure

Tunable



New version of algorithm

Beyond Collaborative Filtering

Content based Recommendations

Visual Similarity

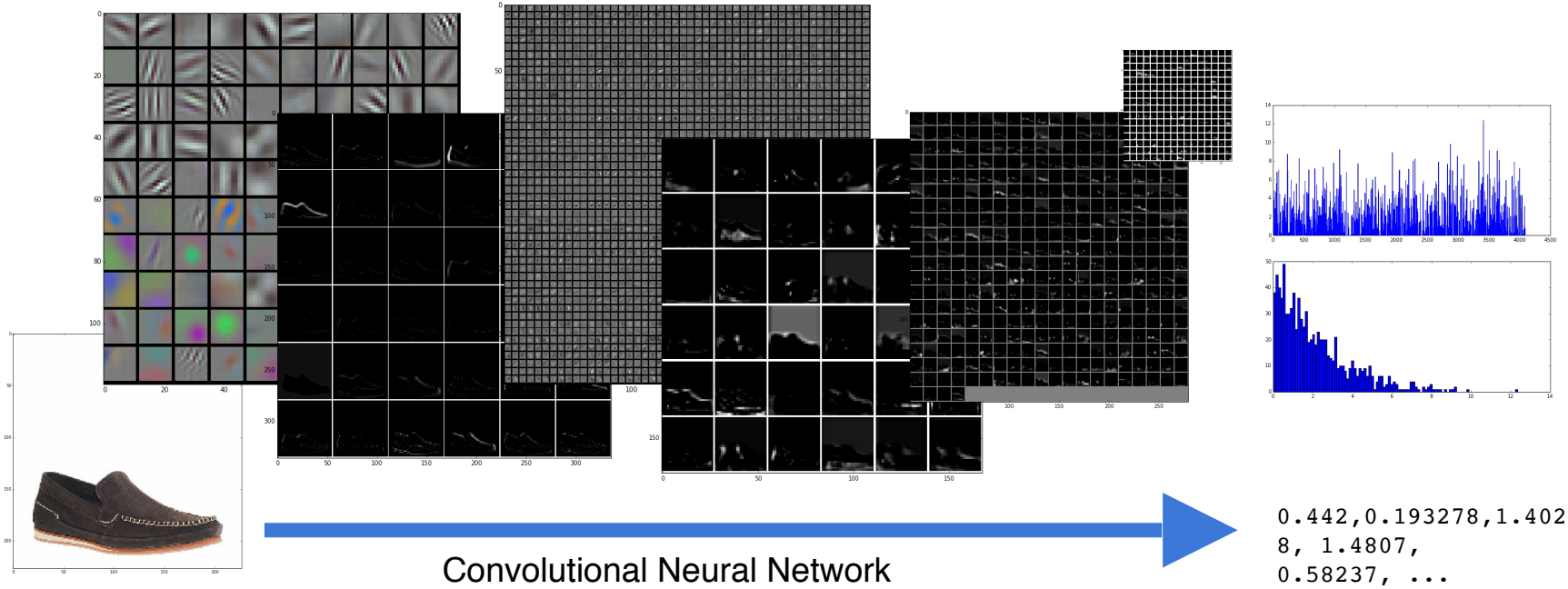
~

~

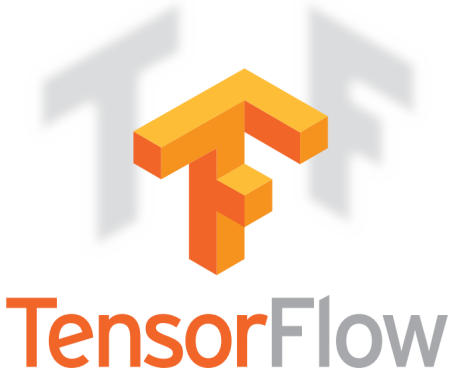
Items are close by visual inspection
no (meta) data needed

Visual similarity

Convolutional Neural Networks



Content based



Open source software library for numerical computation using data flow graphs.

Flexible architecture, runs on one or more CPU and GPUs on desktop, servers and mobile.

Developed by Google's brain team.

Generate feature vectors

Use deep convolutional network trained on ImageNet data (Large Scale Visual Recognition Challenge 2012)

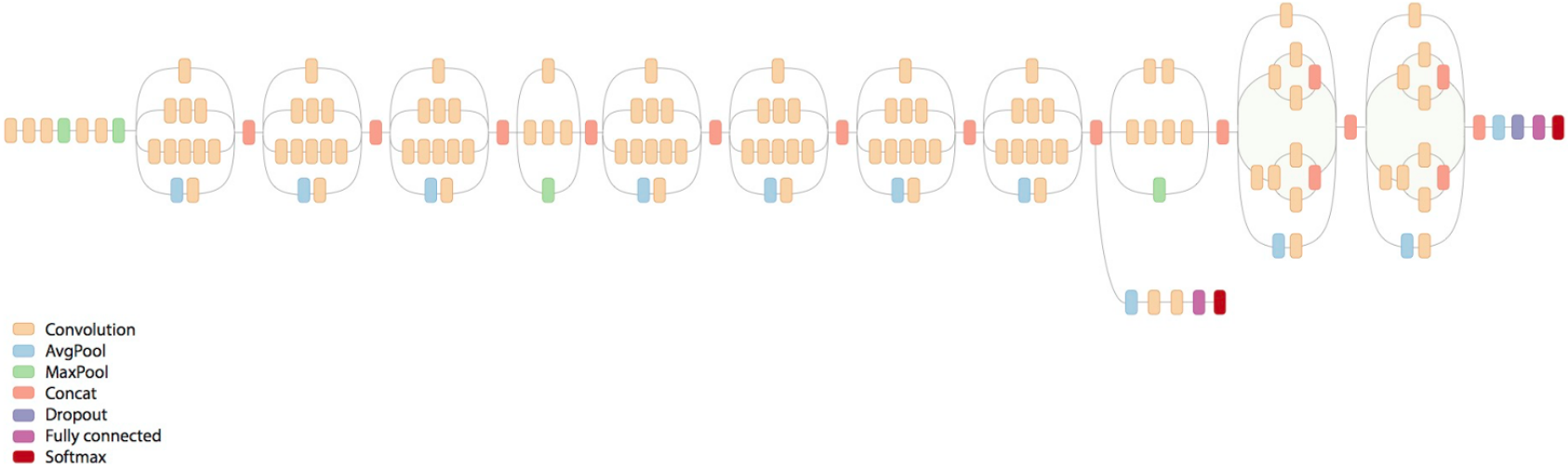
- Generates 2048 dimensional feature vector
- Euclidean distance measures (dis)similarity

Spark: find nearby images

Compute distance between images, find closest neighbor

- Scales with N images like $O(N^2)$ prohibitive for large image sets

Caffe Model(s)



https://github.com/tensorflow/models/tree/master/inception

Generating features with TF

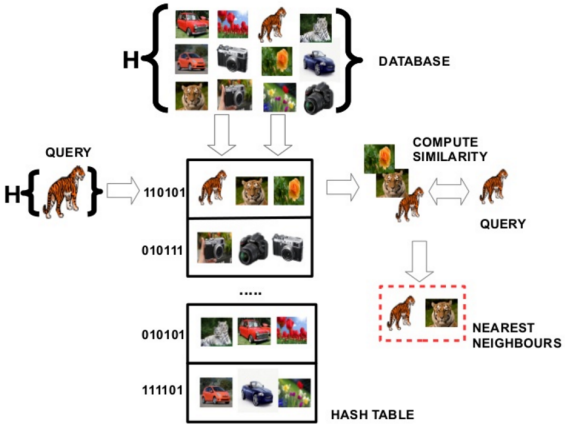
```
import tensorflow as tf
from tensorflow.python.platform import gfile

fname = “demo.jpg”

with gfile.FastGFile('data/network.pb', 'rb') as f:
  graph_def = tf.GraphDef()
  graph_def.ParseFromString(f.read())
  _ = tf.import_graph_def(graph_def, name='')

  pool3 = sess.graph.get_tensor_by_name('pool_3:0')

  image_data = gfile.FastGFile(fname, 'rb').read()

  pool3_features = sess.run(pool3, {'DecodeJpeg/contents:0': image_data})

  print pool3_features
```

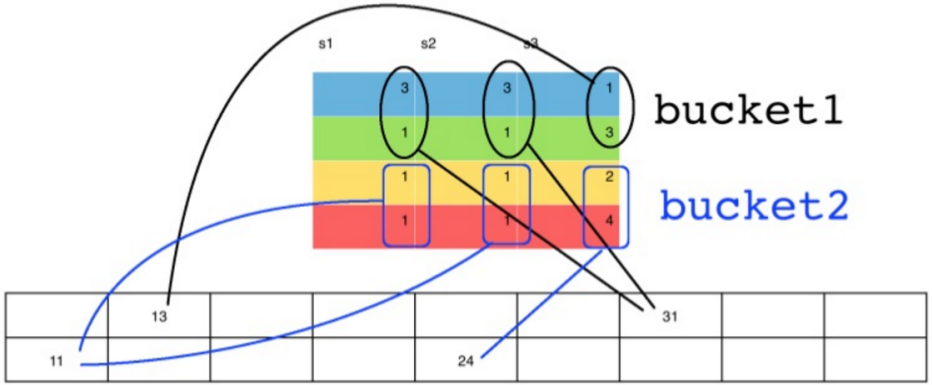
Locality Sensitive Hashing

Central idea

Vectors that are close will be close when projected to a (random) subspace.
Use “law of large numbers” to find vectors that are “probably” close - then calculate exact distance.

Say we use *K* random projections to {0, 1}. Then if *i* and *j* are not close, the probability of them having *K* identical projections is 2^{-K}.





Visual recommender demo

Visual recommender

Productnummer

351348

Original product

Fila New
Sneaker
351348

Recommended products

Fila New Sneaker	Fila New Sneaker	Fila New Sneaker	Fila New Sneaker	Fila New Sneaker
				
475746	396330	182113	351413	396331

We're hiring

werkenbij**wehkamp.nl**