SPARK+AI
SUMMIT EUROPE

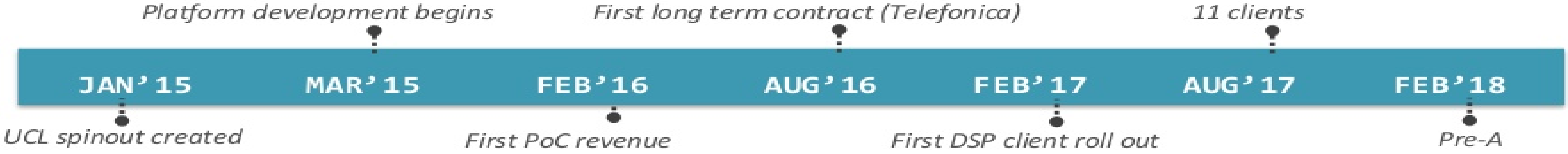
# OpenETL for Real-time Decision Making

Shuai Yuan, VP Data Science, MediaGamma

#SAISEnt7

# About us

- Spinout of UCL’s Computer Science department, specialising in computational advertising and electric commerce



JAN’15

UCL spinout created

MAR’15

*Platform development begins*

FEB’16

*First PoC revenue*

AUG’16

*First long term contract (Telefonica)*

FEB’17

*First DSP client roll out*

AUG’17

*11 clients*

FEB’18

*Pre-A*

- Proved our technology in the ad tech industry w/clients such as Beeswax & Telefonica
- Currently process over 3TB per day, containing tens of billions of daily user events, across tens of millions mobile profiles, spanning 5 countries.
- We work with DSPs/SSPs/exchanges & telcos w/over 85% accuracy & less than 10ms latency

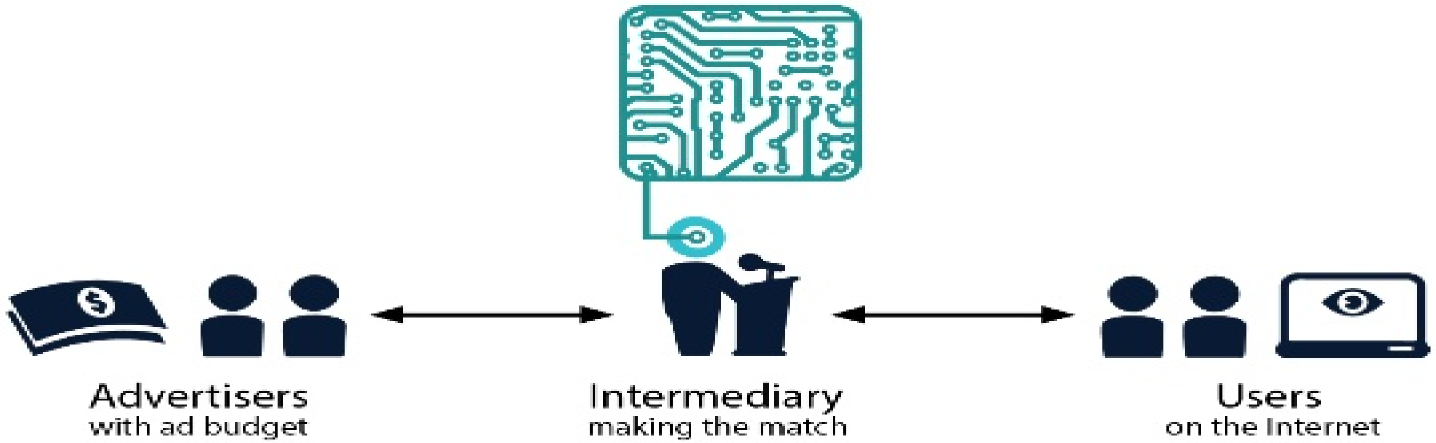
# What do we do

- FRAUD
  - Is the user human?
  - Up to 40% of ads are not shown to humans
- ACTIONS
  - How likely is the user to click on the ad or install an app or register?
- PRICING
  - How much should pay for this impression?
- RELEVANCE
  - Is this user my target audience?
  - How do I find more of the same users?

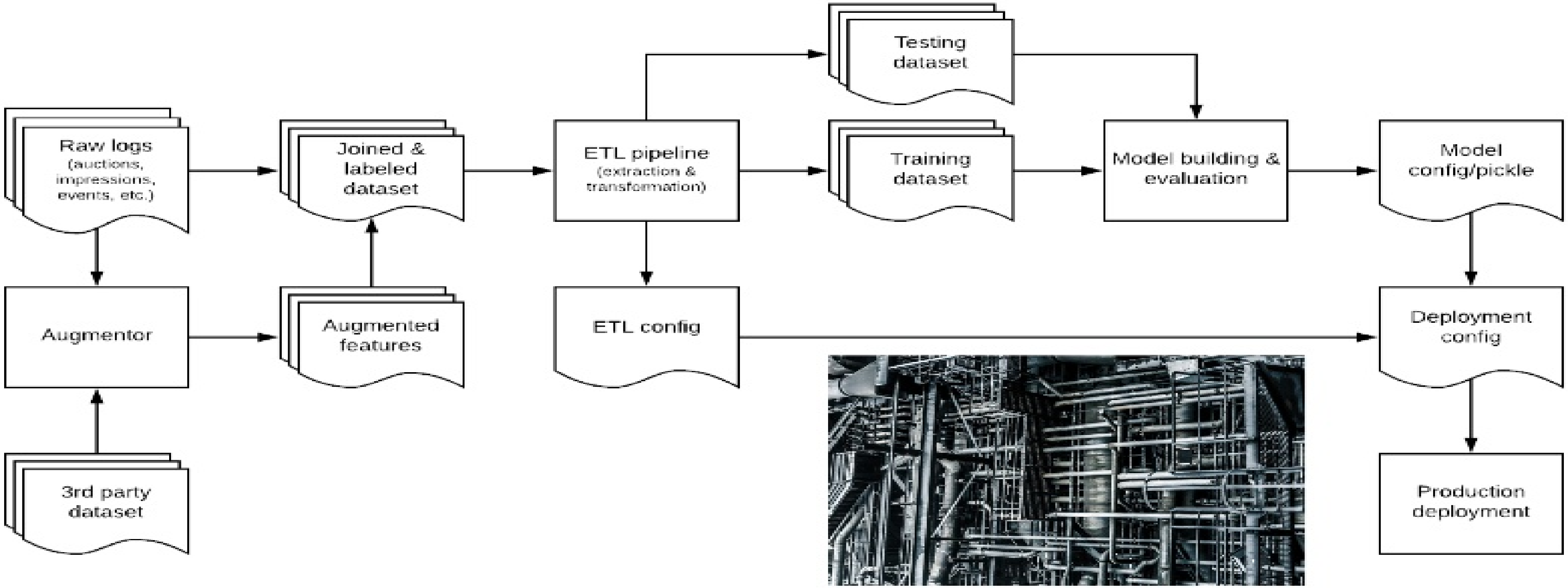
mediagamma

# Real-time Decision Making

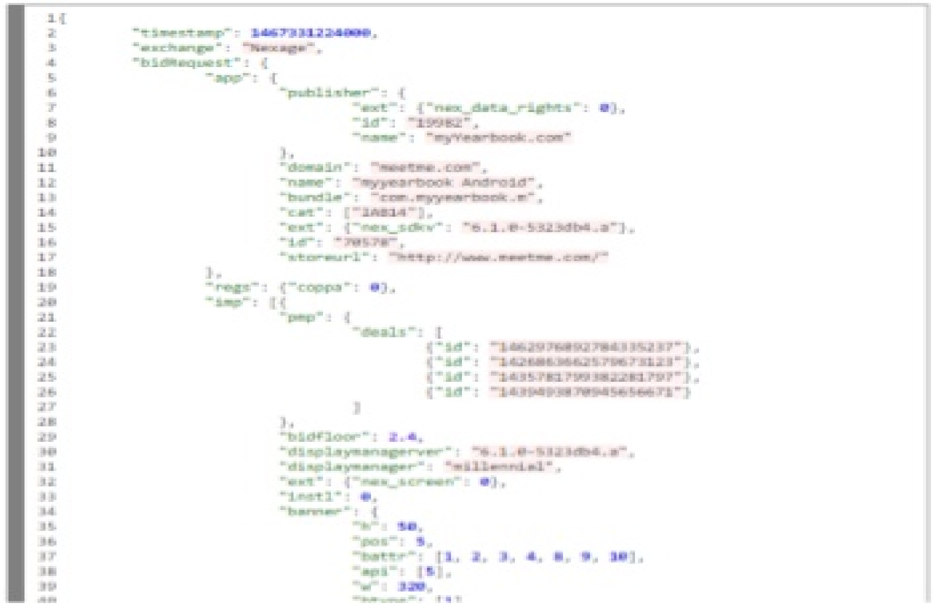
- Real-Time
  - Thousands of QPS
  - 99.9% response under 10ms
- Bidding
  - User response prediction (e.g., CTR prediction)
  - Bid price
- Optimisation
  - ROI
  - Volume (i.e., budget spent)



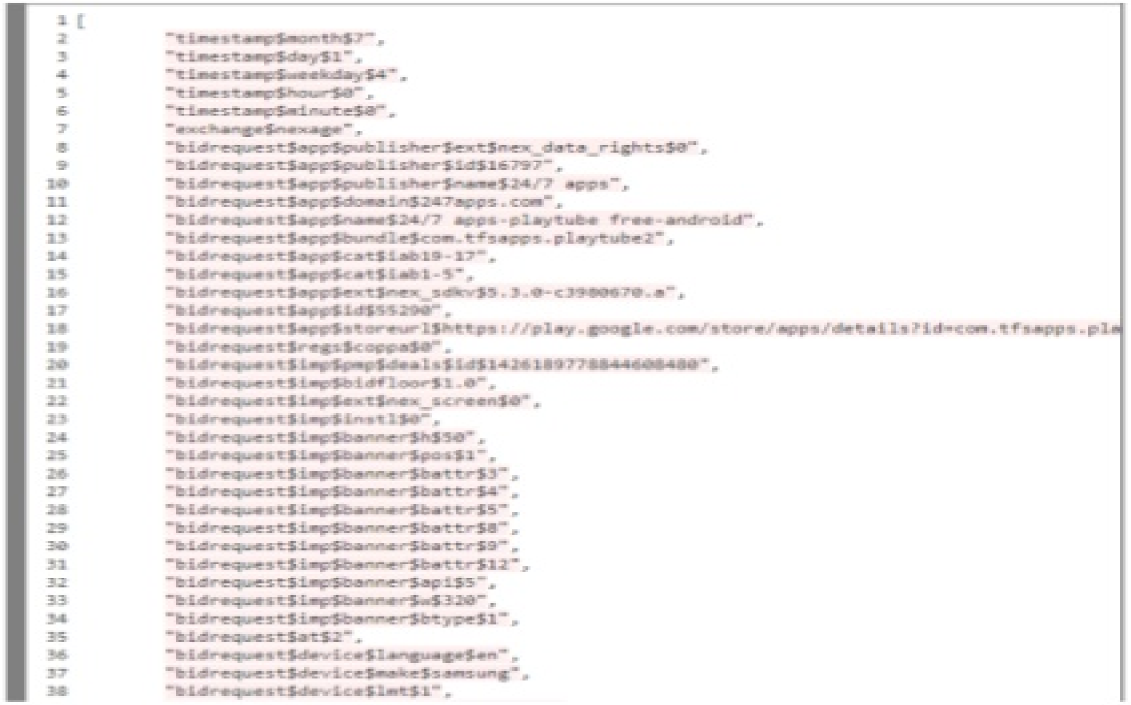
# An end-to-end pipeline



# Feature Engineering

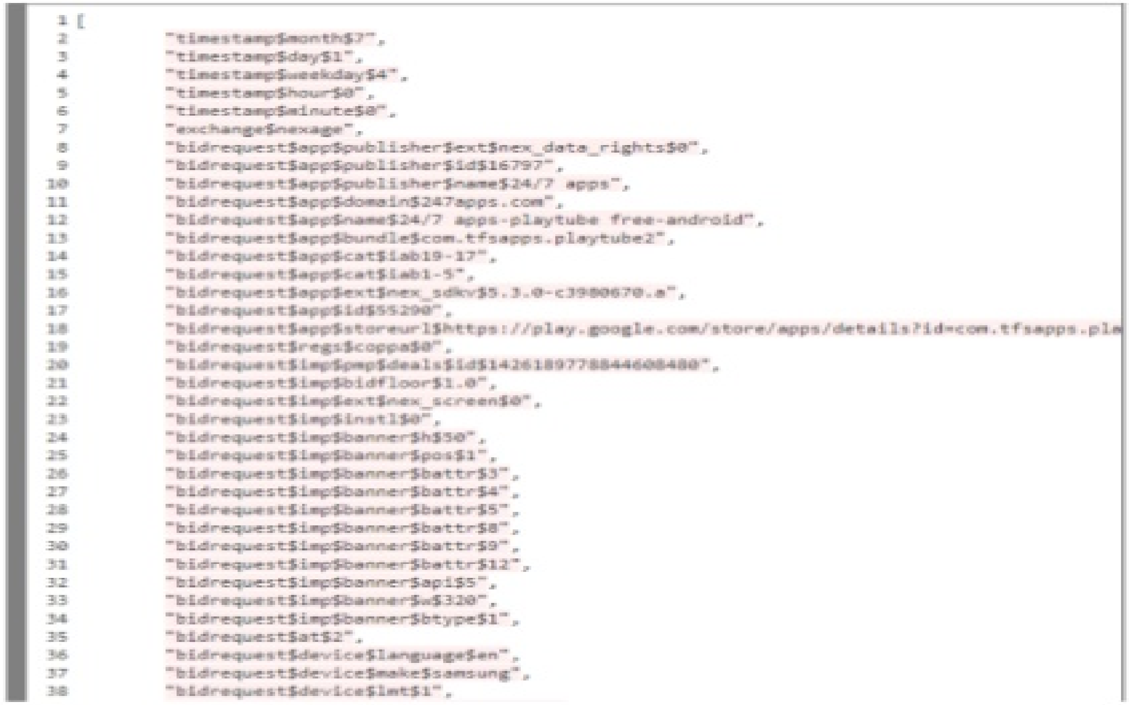


```
{
        "timestamp": 1467331224000,
        "exchange": "Nexage",
        "bidRequest": {
                "app": {
                        "publisher": {
                                "ext": {"nex_data_rights": 0},
                                "id": "19982",
                                "name": "myYearbook.com"
                        },
                        "domain": "meetme.com",
                        "name": "myyearbook Android",
                        "bundle": "com.myyearbook.m",
                        "cat": ["IAB14"],
                        "ext": {"nex_sdkv": "6.1.0-5323db4.a"},
                        "id": "70578",
                        "storeurl": "http://www.meetme.com/"
                },
                "regs": {"coppa": 0},
                "imp": [{
                        "pmp": {
                                "deals": [
                                        {"id": "1462976892784335237"},
                                        {"id": "1426863662579673123"},
                                        {"id": "1435781799382281797"},
                                        {"id": "1439493870945656671"}
                                ]
                        },
                        "bidfloor": 2.4,
                        "displaymanagerver": "6.1.0-5323db4.a",
                        "displaymanager": "millennial",
                        "ext": {"nex_screen": 0},
                        "instl": 0,
                        "banner": {
                                "h": 50,
                                "pos": 5,
                                "battr": [1, 2, 3, 4, 8, 9, 10],
                                "api": [5],
                                "w": 320,
                                "btype": [1]
```

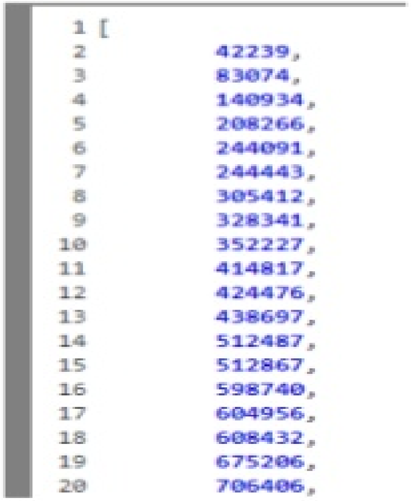


```
[
        "timestamp$month$7",
        "timestamp$day$1",
        "timestamp$weekday$4",
        "timestamp$hour$0",
        "timestamp$minute$0",
        "exchange$nexage",
        "bidrequest$app$publisher$ext$nex_data_rights$0",
        "bidrequest$app$publisher$id$16797",
        "bidrequest$app$publisher$name$24/7 apps",
        "bidrequest$app$domain$247apps.com",
        "bidrequest$app$name$24/7 apps-playtube free-android",
        "bidrequest$app$bundle$com.tfsapps.playtube2",
        "bidrequest$app$cat$iab19-17",
        "bidrequest$app$cat$iab1-5",
        "bidrequest$app$ext$nex_sdkv$5.3.0-c3980670.a",
        "bidrequest$app$id$55290",
        "bidrequest$app$storeurl$https://play.google.com/store/apps/details?id=com.tfsapps.pla
        "bidrequest$regs$coppa$0",
        "bidrequest$imp$pmp$deals$id$1426189778844608480",
        "bidrequest$imp$bidfloor$1.0",
        "bidrequest$imp$ext$nex_screen$0",
        "bidrequest$imp$instl$0",
        "bidrequest$imp$banner$h$50",
        "bidrequest$imp$banner$pos$1",
        "bidrequest$imp$banner$battr$3",
        "bidrequest$imp$banner$battr$4",
        "bidrequest$imp$banner$battr$5",
        "bidrequest$imp$banner$battr$8",
        "bidrequest$imp$banner$battr$9",
        "bidrequest$imp$banner$battr$12",
        "bidrequest$imp$banner$api$5",
        "bidrequest$imp$banner$w$320",
        "bidrequest$imp$banner$btype$1",
        "bidrequest$at$2",
        "bidrequest$device$language$en",
        "bidrequest$device$make$samsung",
        "bidrequest$device$lmt$1",
```

# Feature Engineering contd.



```
[
        "timestamp$month$7",
        "timestamp$day$1",
        "timestamp$weekday$4",
        "timestamp$hour$0",
        "timestamp$minute$0",
        "exchange$nexage",
        "bidrequest$app$publisher$ext$nex_data_rights$0",
        "bidrequest$app$publisher$id$16797",
        "bidrequest$app$publisher$name$24/7 apps",
        "bidrequest$app$domain$247apps.com",
        "bidrequest$app$name$24/7 apps-playtube free-android",
        "bidrequest$app$bundle$com.tfsapps.playtube2",
        "bidrequest$app$cat$iab19-17",
        "bidrequest$app$cat$iab1-5",
        "bidrequest$app$ext$nex_sdkv$5.3.0-c3980670.a",
        "bidrequest$app$id$55290",
        "bidrequest$app$storeurl$https://play.google.com/store/apps/details?id=com.tfsapps.pla
        "bidrequest$regs$coppa$0",
        "bidrequest$imp$pmp$deals$id$1426189778844608480",
        "bidrequest$imp$bidfloor$1.0",
        "bidrequest$imp$ext$nex_screen$0",
        "bidrequest$imp$instl$0",
        "bidrequest$imp$banner$h$50",
        "bidrequest$imp$banner$pos$1",
        "bidrequest$imp$banner$battr$3",
        "bidrequest$imp$banner$battr$4",
        "bidrequest$imp$banner$battr$5",
        "bidrequest$imp$banner$battr$8",
        "bidrequest$imp$banner$battr$9",
        "bidrequest$imp$banner$battr$12",
        "bidrequest$imp$banner$api$5",
        "bidrequest$imp$banner$w$320",
        "bidrequest$imp$banner$btype$1",
        "bidrequest$at$2",
        "bidrequest$device$language$en",
        "bidrequest$device$make$samsung",
        "bidrequest$device$lmt$1",
```

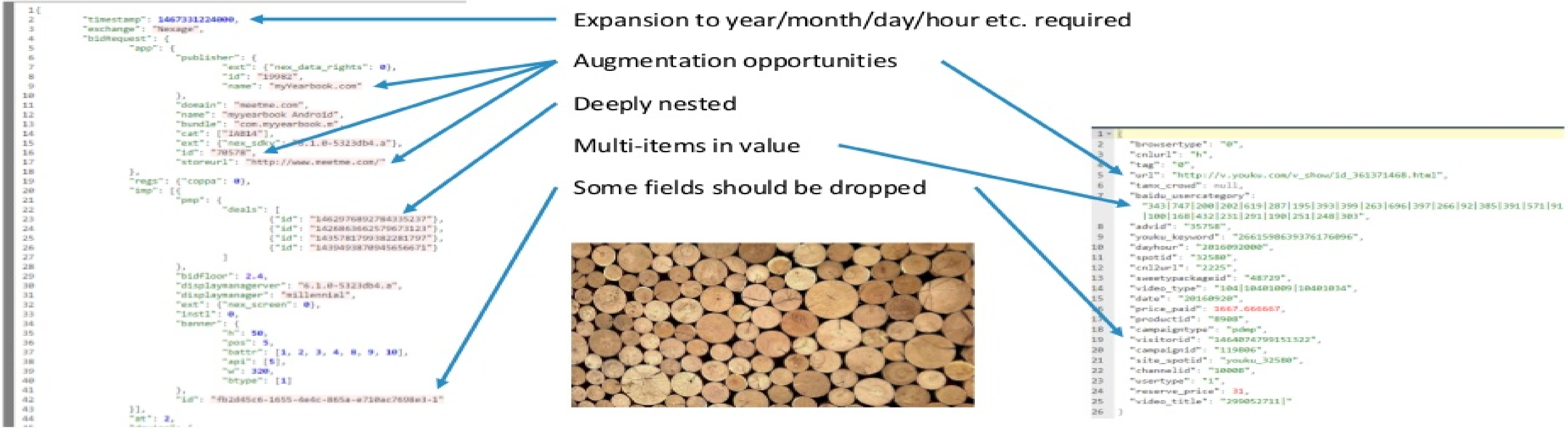


```
[
        42239,
        83074,
        140934,
        208266,
        244091,
        244443,
        305412,
        328341,
        352227,
        414817,
        424476,
        438697,
        512487,
        512867,
        598740,
        604956,
        608432,
        675206,
        706406,
```

mediagamma

# Challenge 1

- How to deal with arbitrary fields in unstructured logs?



Expansion to year/month/day/hour etc. required

Augmentation opportunities

Deeply nested

Multi-items in value

Some fields should be dropped

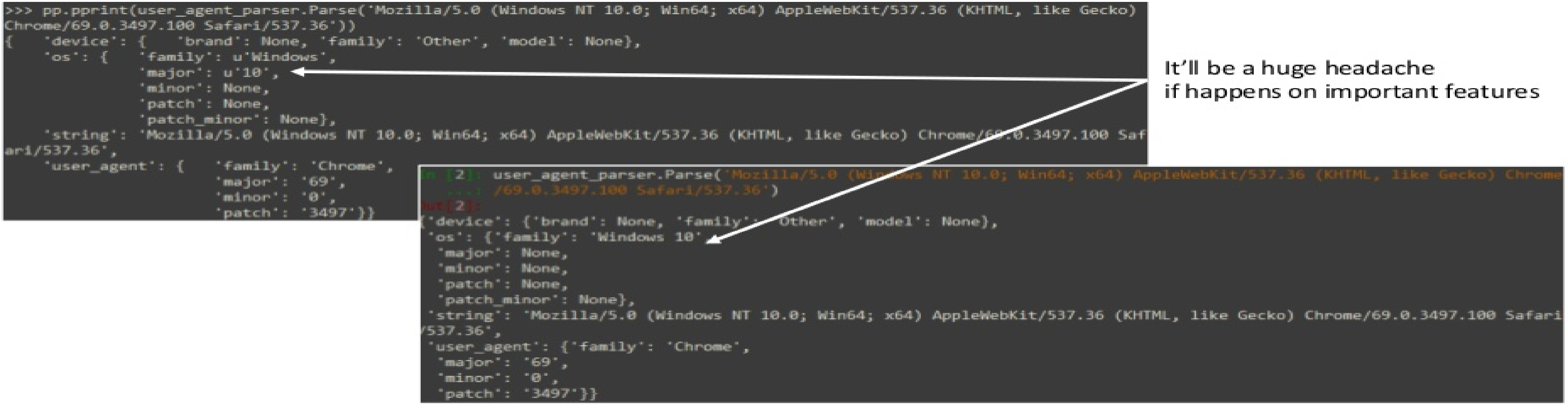
```
{
        "timestamp": 1467331224000,
        "exchange": "Nexage",
        "bidRequest": {
                "app": {
                        "publisher": {
                                "ext": {"nex_data_rights": 0},
                                "id": "19982",
                                "name": "myYearbook.com"
                        },
                        "domain": "meetme.com",
                        "name": "myyearbook Android",
                        "bundle": "com.myyearbook.m",
                        "cat": ["IAB14"],
                        "ext": {"nex_sdkv": "6.1.0-5323db4.a"},
                        "id": "70578",
                        "storeurl": "http://www.meetme.com/"
                },
                "regs": {"coppa": 0},
                "imp": [{
                        "pmp": {
                                "deals": [
                                        {"id": "1462976892784335237"},
                                        {"id": "1426863662579673123"},
                                        {"id": "1435781799382281797"},
                                        {"id": "1439493870945656671"}
                                ]
                        },
                        "bidfloor": 2.4,
                        "displaymanagerver": "6.1.0-5323db4.a",
                        "displaymanager": "millennial",
                        "ext": {"nex_screen": 0},
                        "instl": 0,
                        "banner": {
                                "h": 50,
                                "pos": 5,
                                "battr": [1, 2, 3, 4, 8, 9, 10],
                                "api": [5],
                                "w": 320,
                                "btype": [1]
                        },
                        "id": "fb2d45c6-1655-4e6c-865a-e710ac7698e3-1"
                }],
                "at": 2,
```

```
{
  "browsertype": "0",
  "cnlurl": "h",
  "tag": "0",
  "url": "http://v.youku.com/v_show/id_361371468.html",
  "tamx_crowd": null,
  "baidu_usercategory":
    "343|747|200|202|619|287|195|393|399|263|696|397|266|92|385|391|571|91
    |100|168|432|231|291|190|251|248|303",
  "advid": "35758",
  "youku_keyword": "2661598639376176096",
  "dayhour": "2016092000",
  "spotid": "32580",
  "cnl2url": "2225",
  "sweetypackageid": "48729",
  "video_type": "104|10401009|10401034",
  "date": "20160920",
  "price_paid": 3667.666667,
  "productid": "8908",
  "campaigntype": "pdmp",
  "visitorid": "1464074799151322",
  "campaignid": "119806",
  "site_spotid": "youku_32580",
  "channelid": "10008",
  "usertype": "1",
  "reserve_price": 31,
  "video_title": "299052711|"
}
```

mediagamma

# Challenge 2

- How to guarantee the feature extraction/augmentation consistency?



```
>>> pp.pprint(user_agent_parser.Parse('Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36'))
{   'device': {   'brand': None, 'family': 'Other', 'model': None},
    'os': {   'family': u'Windows',
              'major': u'10',
              'minor': None,
              'patch': None,
              'patch_minor': None},
    'string': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Saf
ari/537.36',
    'user_agent': {   'family': 'Chrome',
                      'major': '69',
                      'minor': '0',
                      'patch': '3497'}}
```

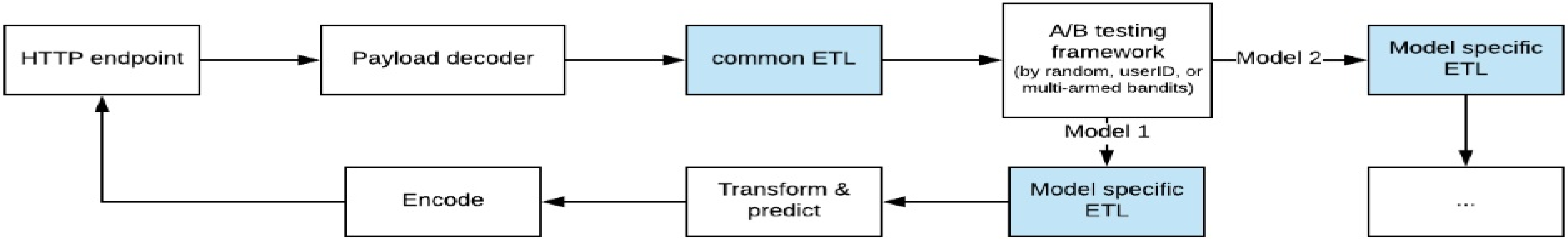
```
In [2]: user_agent_parser.Parse('Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
   ...: /69.0.3497.100 Safari/537.36')
Out[2]:
{'device': {'brand': None, 'family': 'Other', 'model': None},
 'os': {'family': 'Windows 10',
  'major': None,
  'minor': None,
  'patch': None,
  'patch_minor': None},
 'string': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari
/537.36',
 'user_agent': {'family': 'Chrome',
  'major': '69',
  'minor': '0',
  'patch': '3497'}}
```

It'll be a huge headache
if happens on important features

mediagamma

# Challenge 3

- How to make the ETL process portable?



# Challenge 4

- How to do it fast enough?
  - Hundreds of thousands of QPS
  - 10-15ms round trip time
  - Overhead for API & decoding (e.g., protobuf)
  - Cost?
  - It's common to implement the prediction functions in a different language (than python)

mediagamma

# OpenETL

- Tree traversal
  - A recursive function
  - Deals with both structured and unstructured input requests
- Libs + Configuration
  - Build libs for multiple programming language
  - Load configurations at runtime
  - Different levels of tests to guarantee consistency
- Micro services architecture; containerize:
  - I/O
  - Common ETL
  - Experiment control
  - Specific transformation & model & stacking

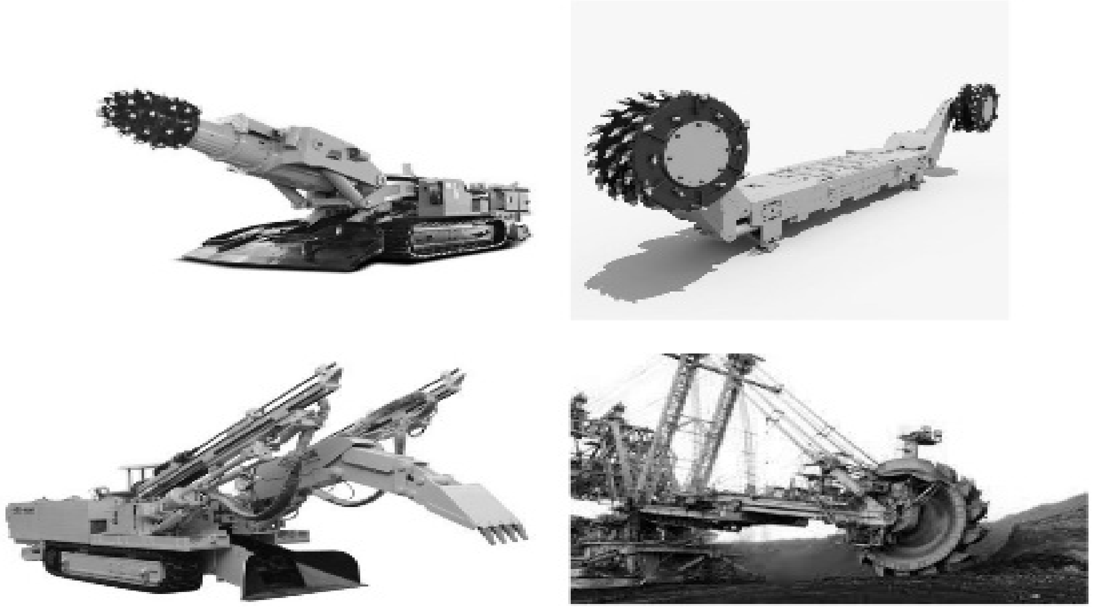
# Alternatives

- Featuretools
  - A framework to perform automated feature engineering. It excels at transforming temporal and relational datasets into feature matrices for machine learning.
  - Featuretools is intended to be run on datasets that can fit in memory on one machine.
- TransmogrifAI
  - An end-to-end AutoML library for structured data written in Scala that runs on top of Apache Spark. It was developed with a focus on accelerating machine learning developer productivity through machine learning automation, and an API that enforces compile-time type-safety, modularity, and reuse.
- bonobo
  - A lightweight Extract-Transform-Load (ETL) framework for Python 3.5+

- https://www.featuretools.com
- https://transmogrif.ai
- https://www.bonobo-project.org

# Extraction

- Operators
  - Object traverse
    - Lists & dicts
    - Optional depth limit
  - Split
  - Exclude
  - Augment
    - Internal & external datasource
  - Evaluate
    - Essentially `eval()`
    - E.g., converting timestamps



mediagamma

# Augmentation

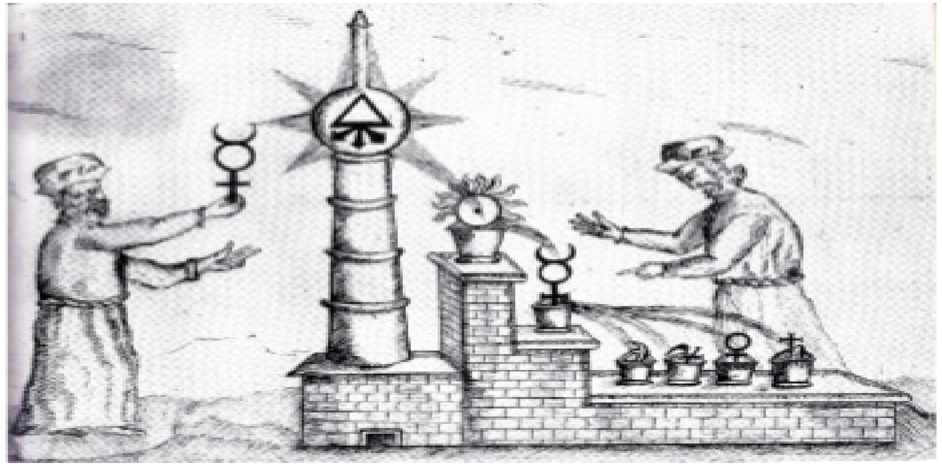
- Examples
  - doc2vec for a given corpus
  - Historical CTR/CVR
  - First-party user data (e.g., abandoned shopping cart value)
  - Time + location -> weather

- Integration
  - As dictionary
  - Real-time API

mediagamma

# Transformation

- Operators
  - CountVectorizer
  - HashingVectorizer
  - Bucketizer
  - MinMaxScaler
  - PolynomialFeatures



- If necessary, trained in Apache Spark
  - For many transformation `fit()` is expensive but `transform()` is cheap
  - E.g., `OpenETLCountVectorizer.copy_from_spark()`

- Rosicrucian Digest on Alchemy, https://www.rosicrucian.org/rosicrucian-digest-alchemy

mediagamma

# Optimisation

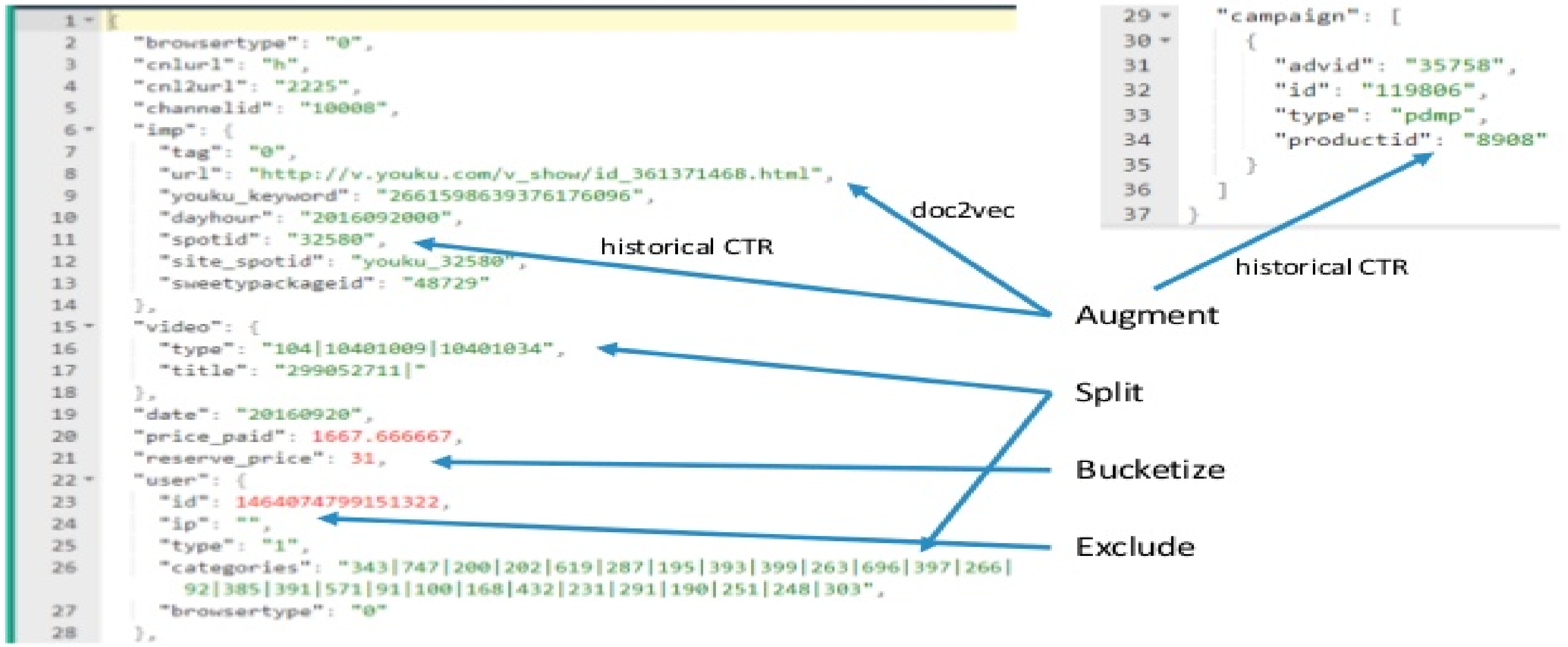
- Higher-level APIs to manipulate the ETL pipeline steps
  - Step selection in training -> step importance
  - Optional priority field
    (dropping steps/features when performance degrades)

- Cython for python, later other programming languages
  - Golang
  - Java

- WikiMedia Commons

# A Real-world Example

- Input
  - Customer defined
  - Text based
  - JSON format
  - Requires further processing

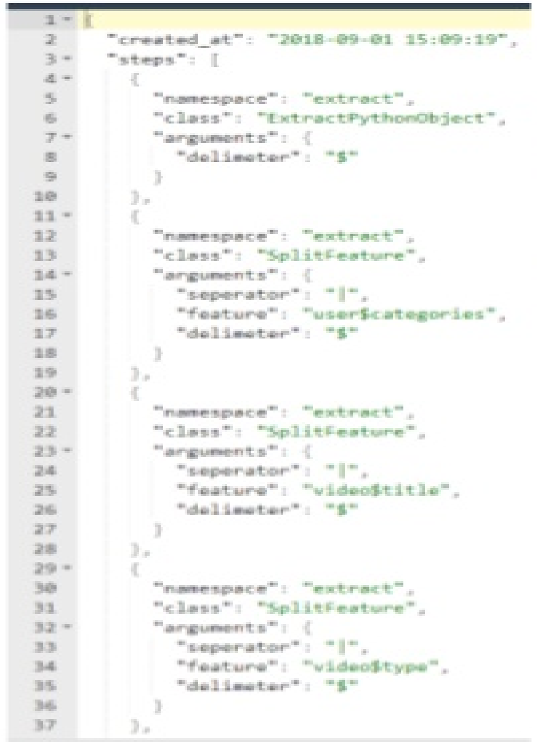


```
{
  "browsertype": "0",
  "cnlurl": "h",
  "cnl2url": "2225",
  "channelid": "10008",
  "imp": {
    "tag": "0",
    "url": "http://v.youku.com/v_show/id_361371468.html",
    "youku_keyword": "2661598639376176096",
    "dayhour": "2016092000",
    "spotid": "32580",
    "site_spotid": "youku_32580",
    "sweetypackageid": "48729"
  },
  "video": {
    "type": "104|10401009|10401034",
    "title": "299052711|"
  },
  "date": "20160920",
  "price_paid": 1667.666667,
  "reserve_price": 31,
  "user": {
    "id": 1464074799151322,
    "ip": "",
    "type": "1",
    "categories": "343|747|200|202|619|287|195|393|399|263|696|397|266|
      92|385|391|571|91|100|168|432|231|291|190|251|248|303",
    "browsertype": "0"
  },
```

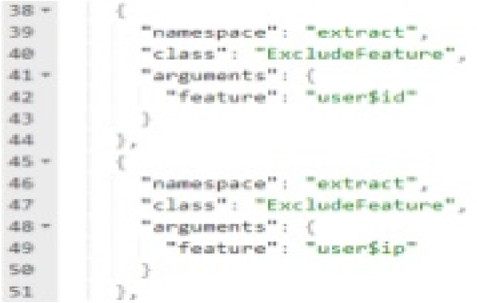
```
  "campaign": [
    {
      "advid": "35758",
      "id": "119806",
      "type": "pdmp",
      "productid": "8908"
    }
  ]
}
```

# A Real-world Example, contd.

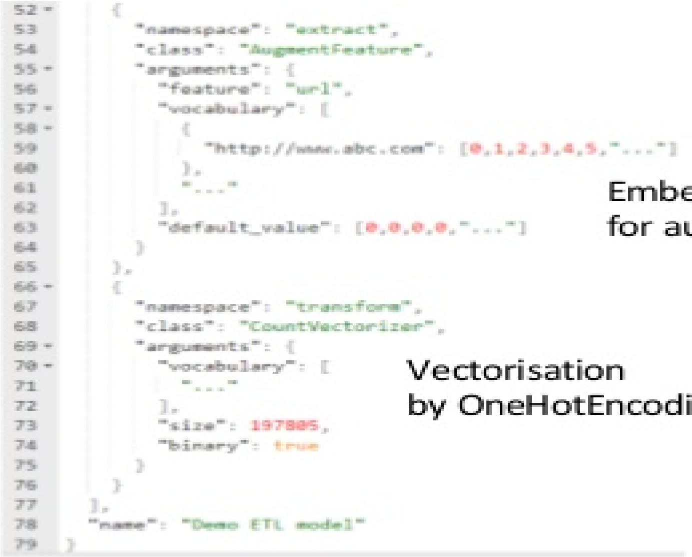
Feature extraction
by traversing
JSON/pyobj tree



```
{
  "created_at": "2018-09-01 15:09:19",
  "steps": [
    {
      "namespace": "extract",
      "class": "ExtractPythonObject",
      "arguments": {
        "delimeter": "$"
      }
    },
    {
      "namespace": "extract",
      "class": "SplitFeature",
      "arguments": {
        "seperator": "|",
        "feature": "user$categories",
        "delimeter": "$"
      }
    },
    {
      "namespace": "extract",
      "class": "SplitFeature",
      "arguments": {
        "seperator": "|",
        "feature": "video$title",
        "delimeter": "$"
      }
    },
    {
      "namespace": "extract",
      "class": "SplitFeature",
      "arguments": {
        "seperator": "|",
        "feature": "video$type",
        "delimeter": "$"
      }
    },
```



```
    {
      "namespace": "extract",
      "class": "ExcludeFeature",
      "arguments": {
        "feature": "user$id"
      }
    },
    {
      "namespace": "extract",
      "class": "ExcludeFeature",
      "arguments": {
        "feature": "user$ip"
      }
    },
```



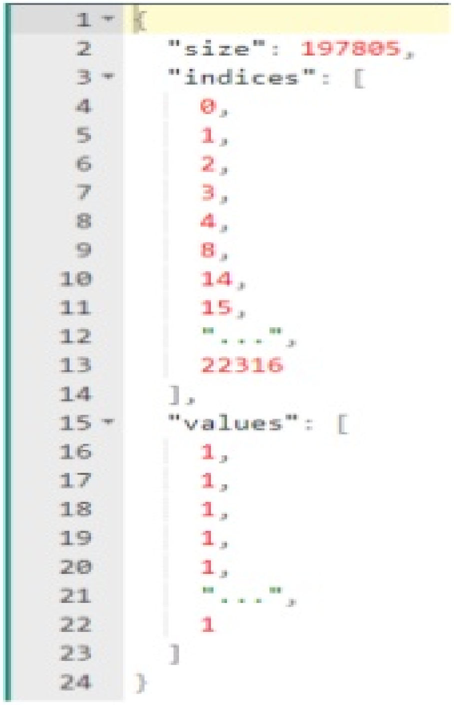
```
    {
      "namespace": "extract",
      "class": "AugmentFeature",
      "arguments": {
        "feature": "url",
        "vocabulary": [
          {
            "http://www.abc.com": [0,1,2,3,4,5,"..."]
          },
          "..."
        ],
        "default_value": [0,0,0,0,"..."]
      }
    },
    {
      "namespace": "transform",
      "class": "CountVectorizer",
      "arguments": {
        "vocabulary": [
          "..."
        ],
        "size": 197805,
        "binary": true
      }
    }
  ],
  "name": "Demo ETL model"
}
```

Embedded dictionary
for augmentation

Vectorisation
by OneHotEncoding

# A Real-world Example, contd.

- Output:
  - Dense / sparse vector: size, indices, values
  - JSON/CSV/Parquet
  - Optional “label” field
- Utilities for format conversion
  - org.apache.spark.ml.linalg.SparseVector
  - scipy.sparse.csr_matrix
  - tf.SparseTensor
  - etc.



```json
{
  "size": 197805,
  "indices": [
    0,
    1,
    2,
    3,
    4,
    8,
    14,
    15,
    "...",
    22316
  ],
  "values": [
    1,
    1,
    1,
    1,
    1,
    "...",
    1
  ]
}
```

# Thank you!

- Questions?

- We are hiring!

- Shuai Yuan, VP Data Science, MediaGamma
- shuai.yuan@mediagamma.com