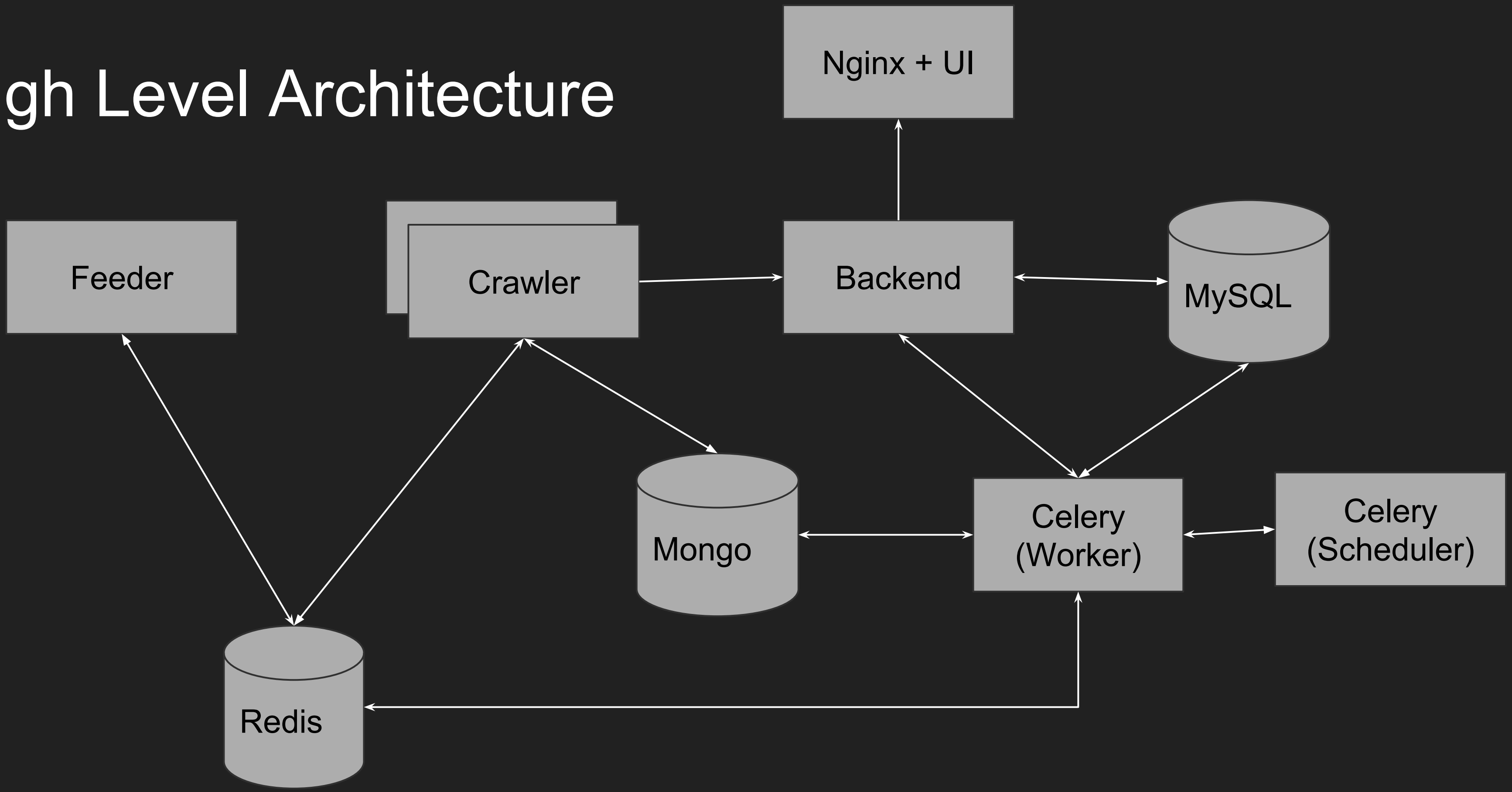
Used Car Service

jwyx88003@gmail.com

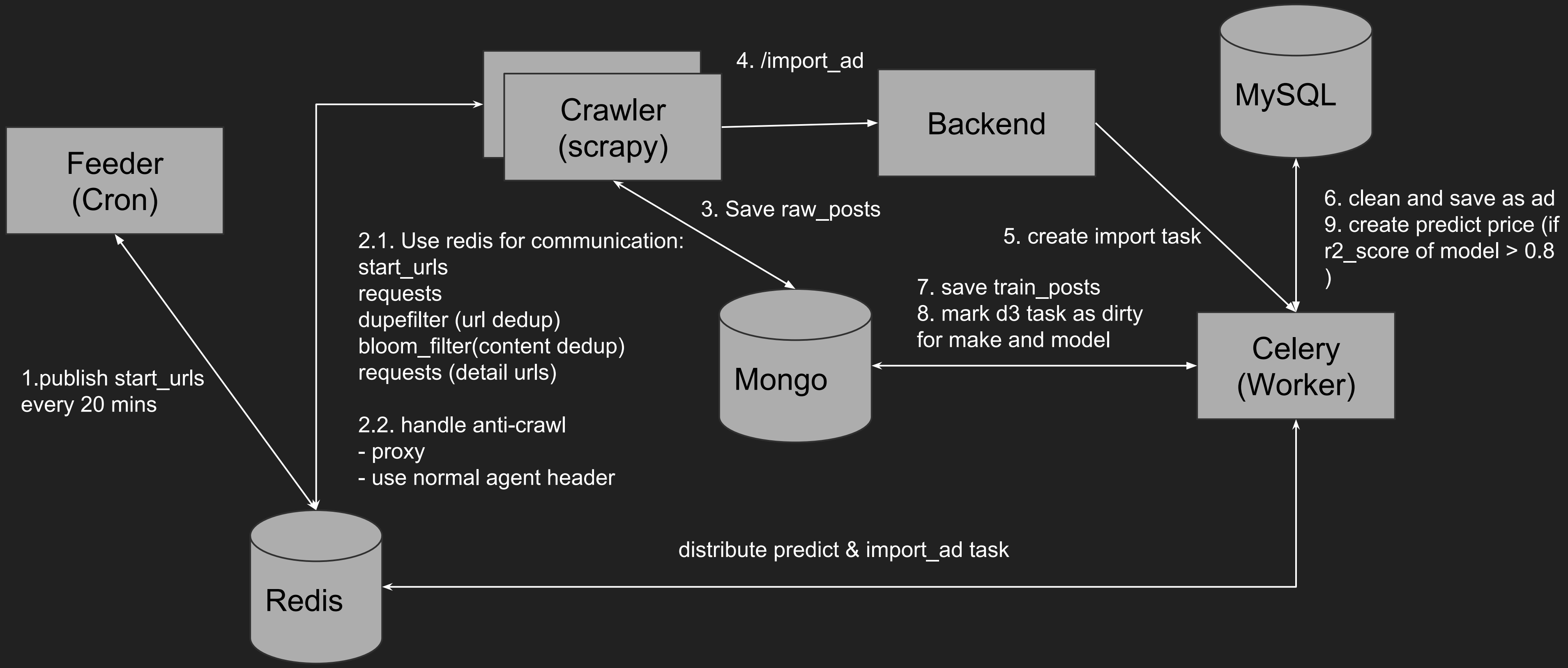
Requirements

- Crawl used car ads from craigslist
- Extract structure information and save to database
- Visualize data (e.g. count, price, etc) group by make and model
- Apply regression to generate predict price for new ads

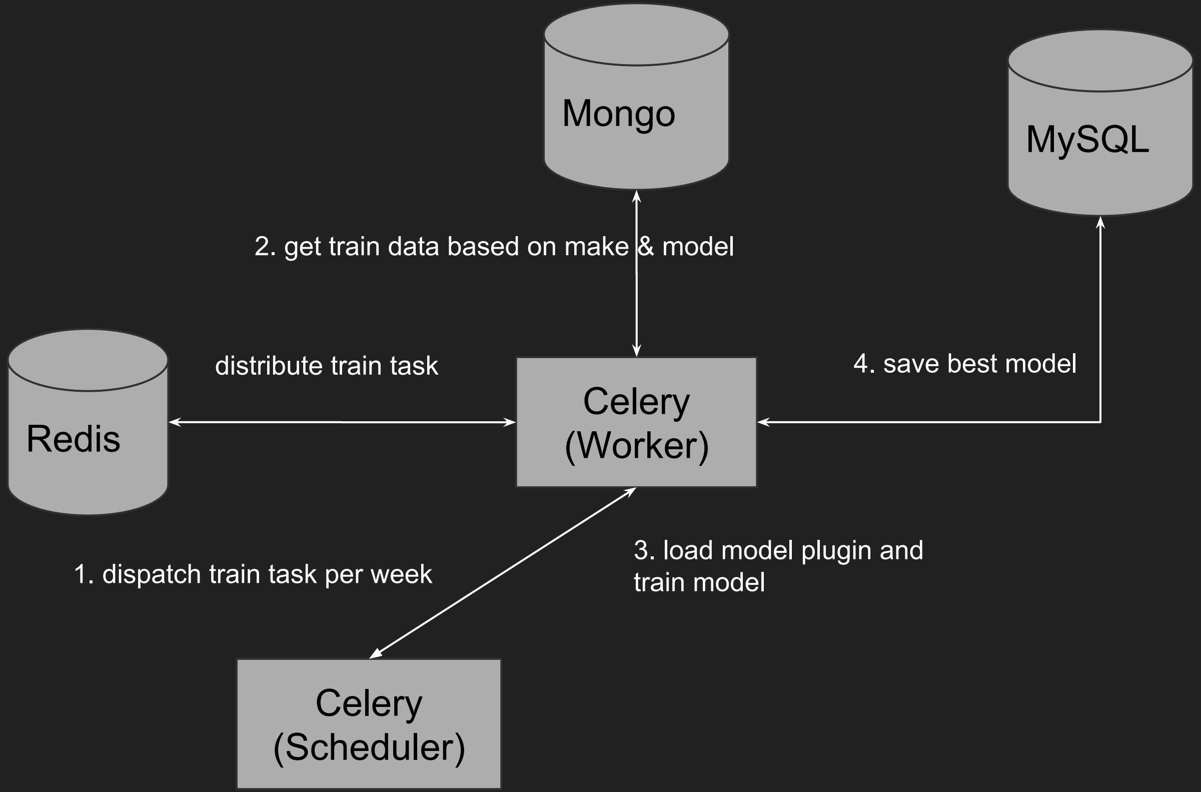
High Level Architecture



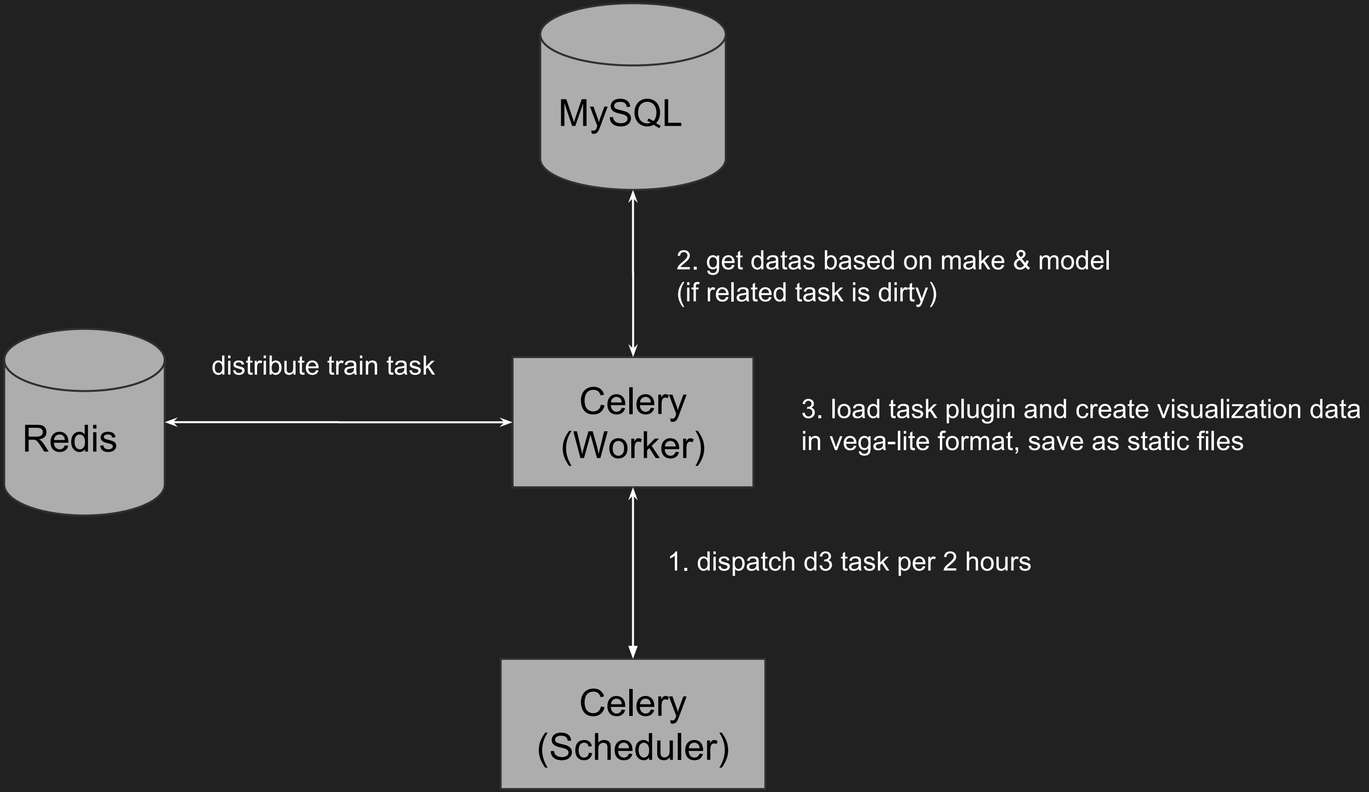
Workflow: Crawl and import post



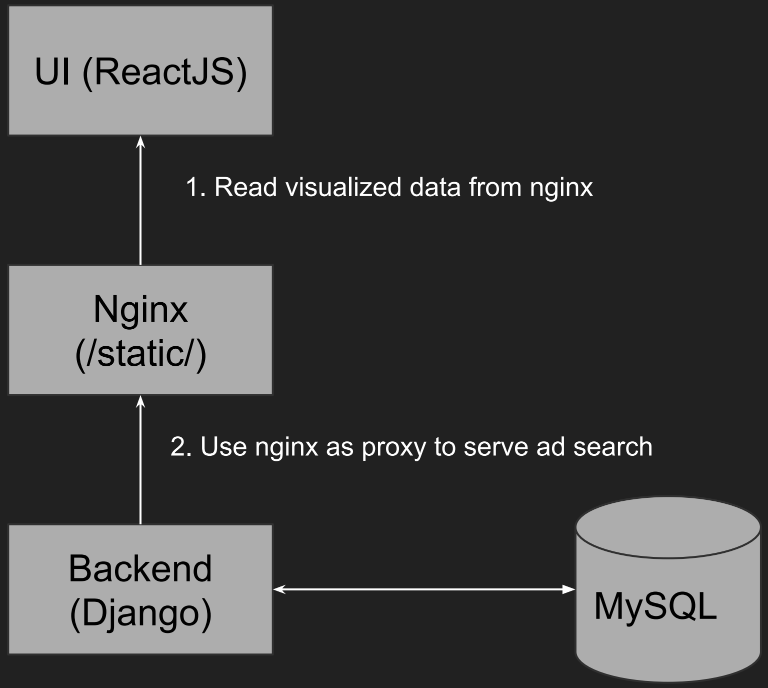
Workflow: Train model


Mongo
MySQL
2. get train data based on make & model
Redis
distribute train task
Celery (Worker)
4. save best model
1. dispatch train task per week
3. load model plugin and train model
Celery (Scheduler)

Workflow: Data Visualization


MySQL
2. get datas based on make & model
(if related task is dirty)
distribute train task
Redis
Celery
(Worker)
3. load task plugin and create visualization data
in vega-lite format, save as static files
1. dispatch d3 task per 2 hours
Celery
(Scheduler)

Workflow: UI


UI (ReactJS)
1. Read visualized data from nginx
Nginx (/static/)
2. Use nginx as proxy to serve ad search
Backend (Django)
MySQL

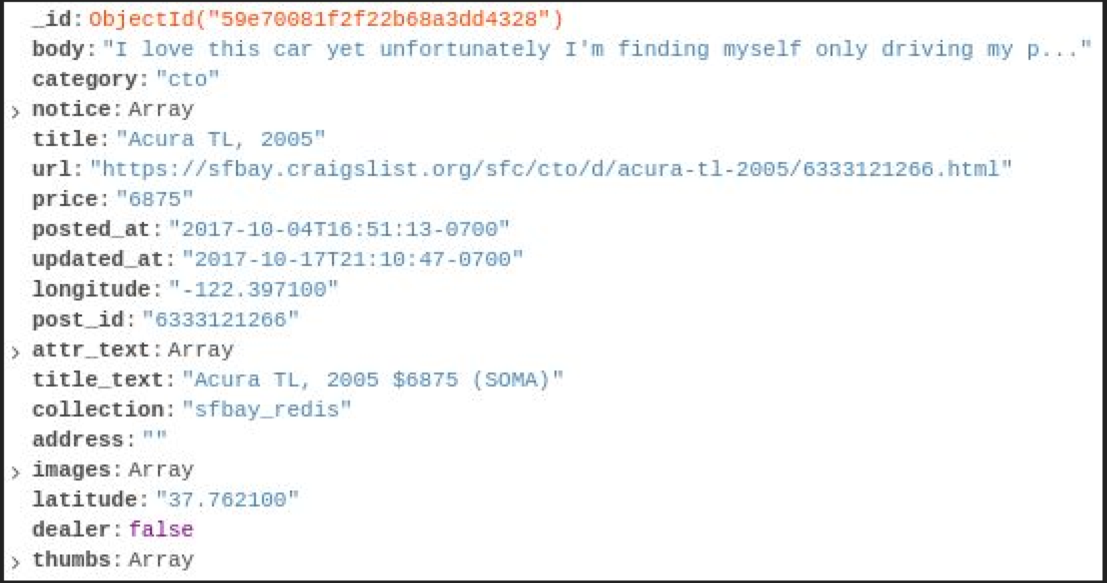
Data Models

MySQL:

- ad:
  - id (PK), url, make_id (FK), model_id (FK), year, price, title_status, odometer, size, category
  - color, condition, drive, fuel, transmission, latitude, longitude
  - dealer, cylinders, posted_at, post_url, predict_price, predicted_at, predict_info
- make:
  - id, name
- carmodel:
  - id, name, make_id (FK), predict_model

Data Model (CONT.)

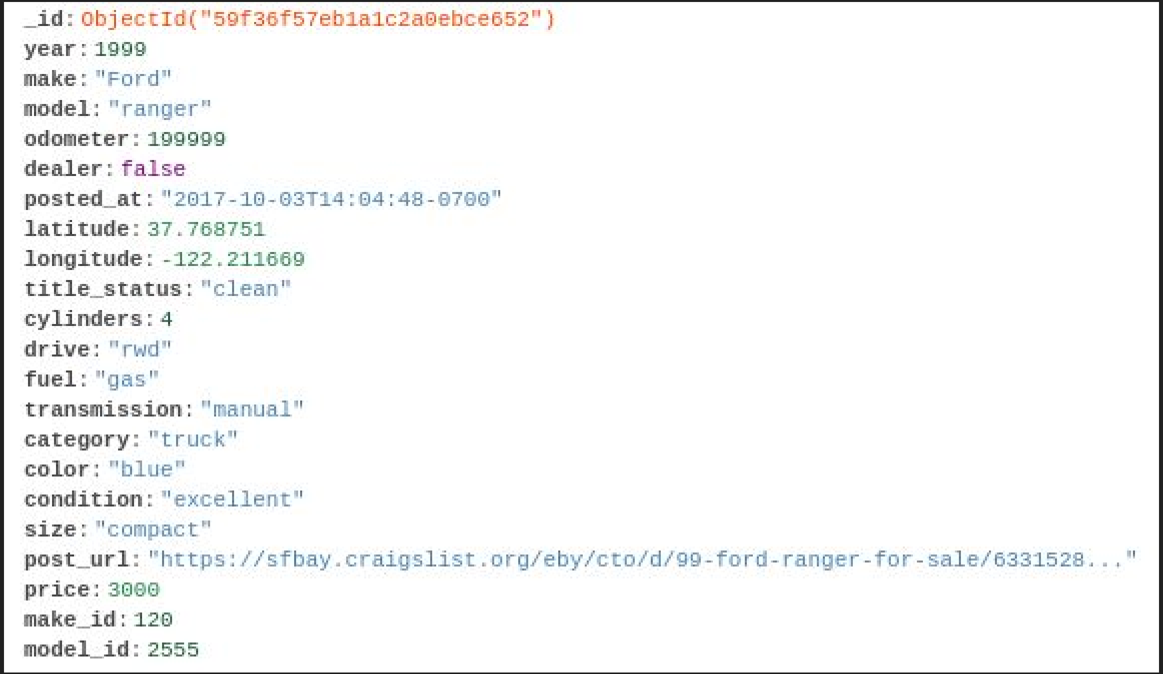
MongoDB: sfbay_redis



```
  _id: ObjectId("59e70081f2f22b68a3dd4328")
  body: "I love this car yet unfortunately I'm finding myself only driving my p..."
  category: "cto"
› notice: Array
  title: "Acura TL, 2005"
  url: "https://sfbay.craigslist.org/sfc/cto/d/acura-tl-2005/6333121266.html"
  price: "6875"
  posted_at: "2017-10-04T16:51:13-0700"
  updated_at: "2017-10-17T21:10:47-0700"
  longitude: "-122.397100"
  post_id: "6333121266"
› attr_text: Array
  title_text: "Acura TL, 2005 $6875 (SOMA)"
  collection: "sfbay_redis"
  address: ""
› images: Array
  latitude: "37.762100"
  dealer: false
› thumbs: Array
```

Data Model (CONT.)

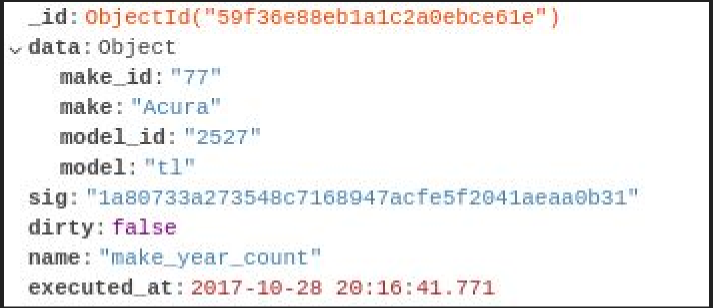
MongoDB: sfbay_train



```
_id: ObjectId("59f36f57eb1a1c2a0ebce652")
year: 1999
make: "Ford"
model: "ranger"
odometer: 199999
dealer: false
posted_at: "2017-10-03T14:04:48-0700"
latitude: 37.768751
longitude: -122.211669
title_status: "clean"
cylinders: 4
drive: "rwd"
fuel: "gas"
transmission: "manual"
category: "truck"
color: "blue"
condition: "excellent"
size: "compact"
post_url: "https://sfbay.craigslist.org/eby/cto/d/99-ford-ranger-for-sale/6331528..."
price: 3000
make_id: 120
model_id: 2555
```

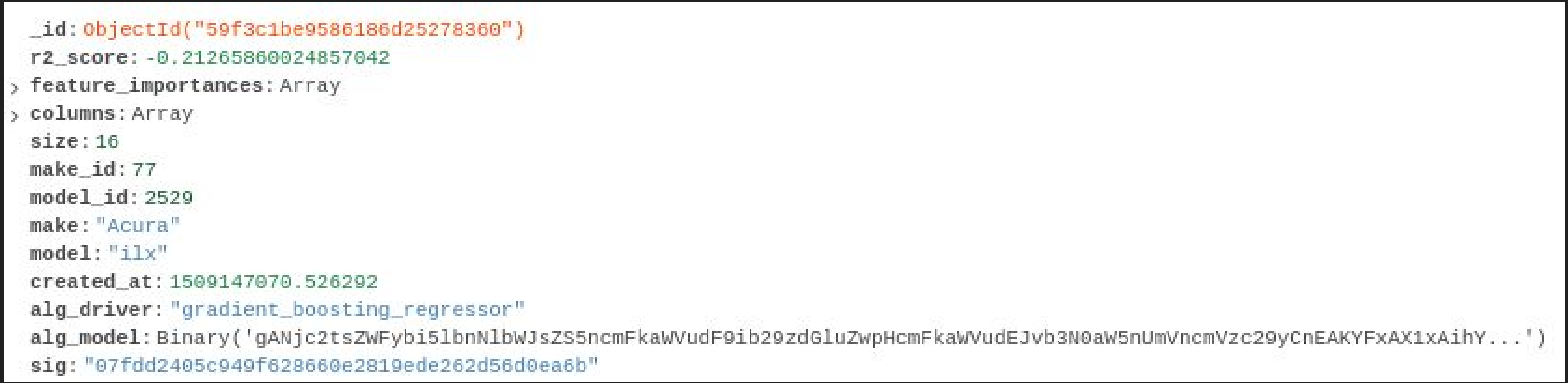
Data Model (CONT.)

MongoDB: sfbay_d3_task



```
 _id: ObjectId("59f36e88eb1a1c2a0ebce61e")
⌄data: Object
    make_id: "77"
    make: "Acura"
    model_id: "2527"
    model: "tl"
 sig: "1a80733a273548c7168947acfe5f2041aeaa0b31"
 dirty: false
 name: "make_year_count"
 executed_at: 2017-10-28 20:16:41.771
```

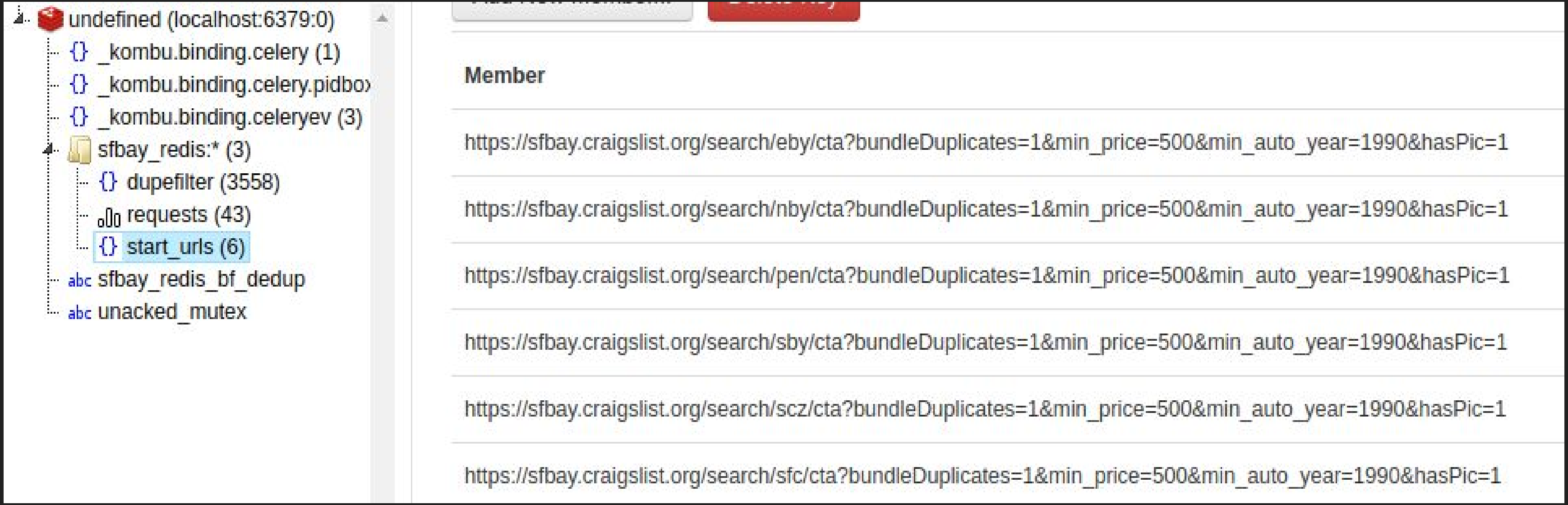
MongoDB: sfbay_model



```
 _id: ObjectId("59f3c1be9586186d25278360")
 r2_score: -0.21265860024857042
› feature_importances: Array
› columns: Array
 size: 16
 make_id: 77
 model_id: 2529
 make: "Acura"
 model: "ilx"
 created_at: 1509147070.526292
 alg_driver: "gradient_boosting_regressor"
 alg_model: Binary('gANjc2tsZWFybi5lbnNlbWJsZS5ncmFkaWVudF9ib29zdGluZwpHcmFkaWVudEJvb3N0aW5nUmVncmVzc29yCnEAKYFxAX1xAihY...')
 sig: "07fdd2405c949f628660e2819ede262d56d0ea6b"
```

Data Model (CONT.)

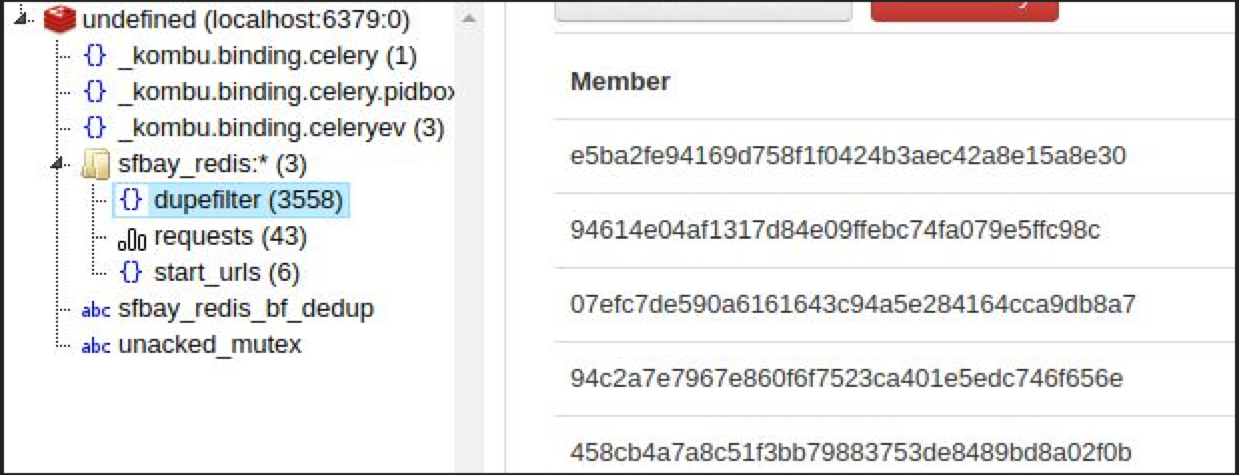
Redis: start_urls


undefined (localhost:6379:0)
{} _kombu.binding.celery (1)
{} _kombu.binding.celery.pidbo›
{} _kombu.binding.celeryev (3)
sfbay_redis:* (3)
{} dupefilter (3558)
requests (43)
{} start_urls (6)
abc sfbay_redis_bf_dedup
abc unacked_mutex

Member
https://sfbay.craigslist.org/search/eby/cta?bundleDuplicates=1&min_price=500&min_auto_year=1990&hasPic=1
https://sfbay.craigslist.org/search/nby/cta?bundleDuplicates=1&min_price=500&min_auto_year=1990&hasPic=1
https://sfbay.craigslist.org/search/pen/cta?bundleDuplicates=1&min_price=500&min_auto_year=1990&hasPic=1
https://sfbay.craigslist.org/search/sby/cta?bundleDuplicates=1&min_price=500&min_auto_year=1990&hasPic=1
https://sfbay.craigslist.org/search/scz/cta?bundleDuplicates=1&min_price=500&min_auto_year=1990&hasPic=1
https://sfbay.craigslist.org/search/sfc/cta?bundleDuplicates=1&min_price=500&min_auto_year=1990&hasPic=1

Data Model (CONT.)

Redis: dupefilter (url dedup)

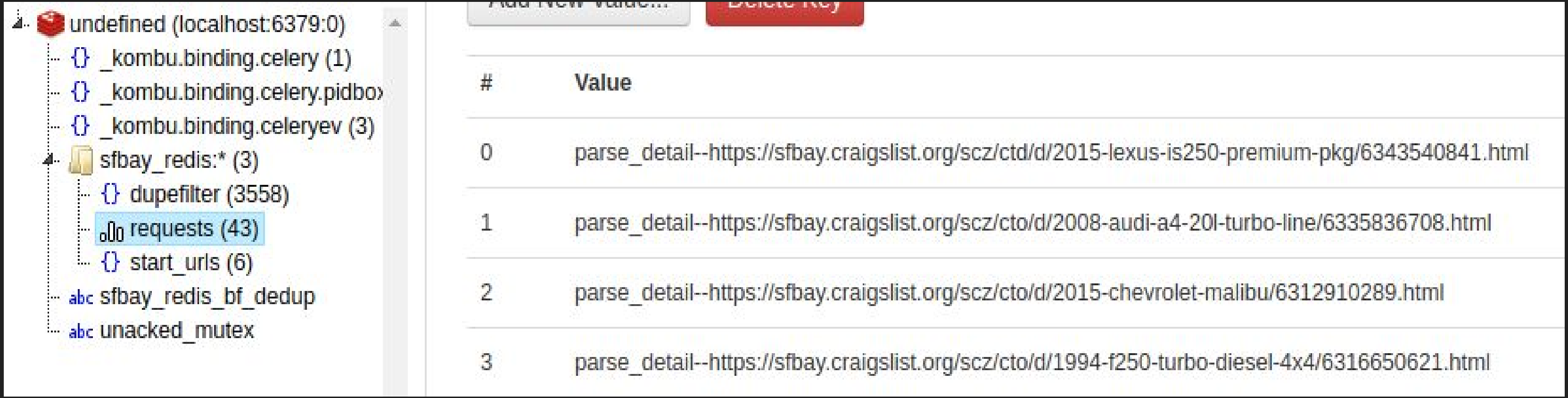


```
undefined (localhost:6379:0)
  _kombu.binding.celery (1)
  _kombu.binding.celery.pidbox
  _kombu.binding.celeryev (3)
  sfbay_redis:* (3)
    dupefilter (3558)
    requests (43)
    start_urls (6)
  sfbay_redis_bf_dedup
  unacked_mutex
```

Member
e5ba2fe94169d758f1f0424b3aec42a8e15a8e30
94614e04af1317d84e09ffebc74fa079e5ffc98c
07efc7de590a6161643c94a5e284164cca9db8a7
94c2a7e7967e860f6f7523ca401e5edc746f656e
458cb4a7a8c51f3bb79883753de8489bd8a02f0b

Data Model (CONT.)

Redis: requests


undefined (localhost:6379:0)
_kombu.binding.celery (1)
_kombu.binding.celery.pidbox
_kombu.binding.celeryev (3)
sfbay_redis:* (3)
dupefilter (3558)
requests (43)
start_urls (6)
sfbay_redis_bf_dedup
unacked_mutex

#	Value
0	parse_detail--https://sfbay.craigslist.org/scz/ctd/d/2015-lexus-is250-premium-pkg/6343540841.html
1	parse_detail--https://sfbay.craigslist.org/scz/cto/d/2008-audi-a4-20l-turbo-line/6335836708.html
2	parse_detail--https://sfbay.craigslist.org/scz/cto/d/2015-chevrolet-malibu/6312910289.html
3	parse_detail--https://sfbay.craigslist.org/scz/cto/d/1994-f250-turbo-diesel-4x4/6316650621.html

APIs

- POST /import_ad/: {“post_url”: “xxx”}
- CRUD /ads/
- CRUD /makes/
- CRUD /models/

Deployment

- Use docker and docker-compose
  - Four infra containers: MySQL, MongoDB, Redis, Frontend (nginx)
  - Three app containers: Backend, Feeder, Crawler
  - Establish overlay networks between containers and use hostname to communicate
- Deploy at one VM
  - 2 GB memory and 2 CPU
  - Memory usage tuning
    - Use custom mysql configuration to decrease memory usage
    - Set max memory of celery worker (one worker and use 500 MB): --concurrency=1 --max-memory-per-child=500000

Design Consideration

- Storage choice:
  - MySQL: save structure data and leverage rich query feature
  - MongoDB: save semi-structure data
    - Raw post from craigslist
    - Train data
    - Train model
    - D3 Task
  - Redis: fast, in memory and offer rich data structure
    - Celery: use as broker for task
    - Crawler: use as request queue, bitset and dedup set
    - Feeder: use as start_urls queue
  - File System: save static files
    - Data visualization data in Vega-Lite format
    - Client side code

Design Consideration (CONT.)

- Extensibility
  - Use plugin mechanism
    - Train model task
    - D3 task to generate different visualization data
- Scalability
  - Scale to N replica
    - Crawler share redis
    - Celery worker
    - Backend server
  - Use offline jobs if real-time calculation is not required
    - Train model
    - D3 task
  - Remove unnecessary calculation
    - Use dirty flag of D3 task

Q & A