ENERGEEZE + CBRE

ENERGEEZE + CBRE

+ arcsine

Outline

- **Introduction**
- Problem
- Solution overview
- Data modeling
- Backend services
- Frontend
- Demo
- Competitive advantages

arcsine

Outline

- Introduction
- **Problem**
- Solution overview
- Data modeling
- Backend services
- Frontend
- Demo
- Competitive advantages

Problem

- Predict future energy load in a given area
- Predict estimated monthly energy bill for a given property
- Goal:
    - Increase profits on energy purchases
    - Increase customer satisfaction on property purchase/lease

Outline

- Introduction
- Problem
- **Solution overview**
- Data modeling
- Backend services
- Frontend
- Demo
- Competitive advantages

arcsine

Watt’s Up?

(Watts Usage Prediction)

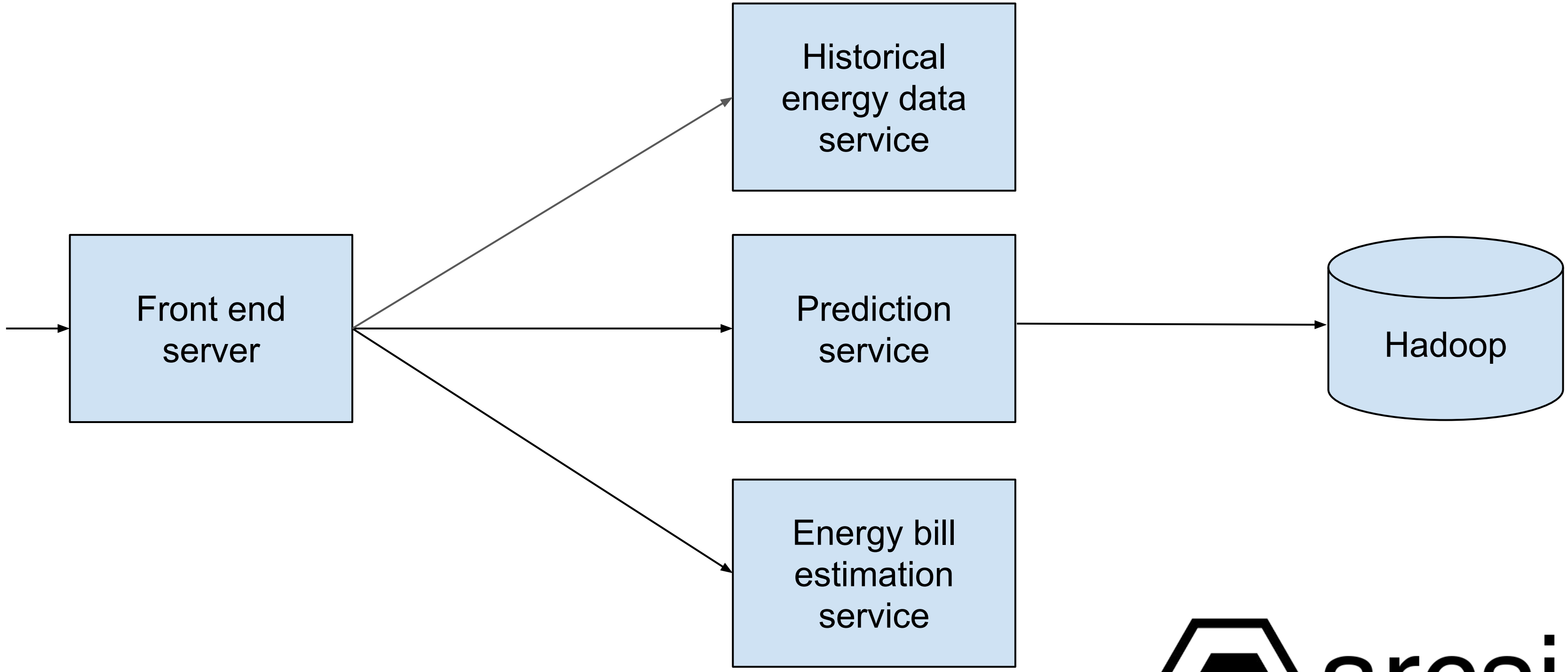
arcsine

Watt’s Up - Three Stages

- Data modeling
  - Energy usage prediction
  - Energy bill for property
- Services
  - Energy usage prediction
  - Energy bill for property
- Interactive web front end
  - Talks to services
  - Data visualization

arcsine

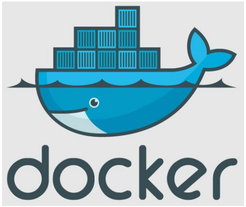
Architecture – Microservices


Historical energy data service
Front end server
Prediction service
Hadoop
Energy bill estimation service

arcsine

NGINX

gunicorn


docker

B

Flask
web development,
one drop at a time

scikit
learn

Spark

APACHE
HBASE

Outline

- Introduction
- Problem
- Solution overview
- **Data modeling**
  - **Energy usage**
  - Property energy bill
- Backend services
- Frontend
- Demo
- Competitive advantages

arcsine

Spark

- Computing framework on Hadoop
- Distributed data gathering
  - ERCOT energy data
  - NOAA GHCN-Daily weather data
- Distributed data processing

APACHE HBASE

- Distributed, scalable big data store
- Storage of weather data

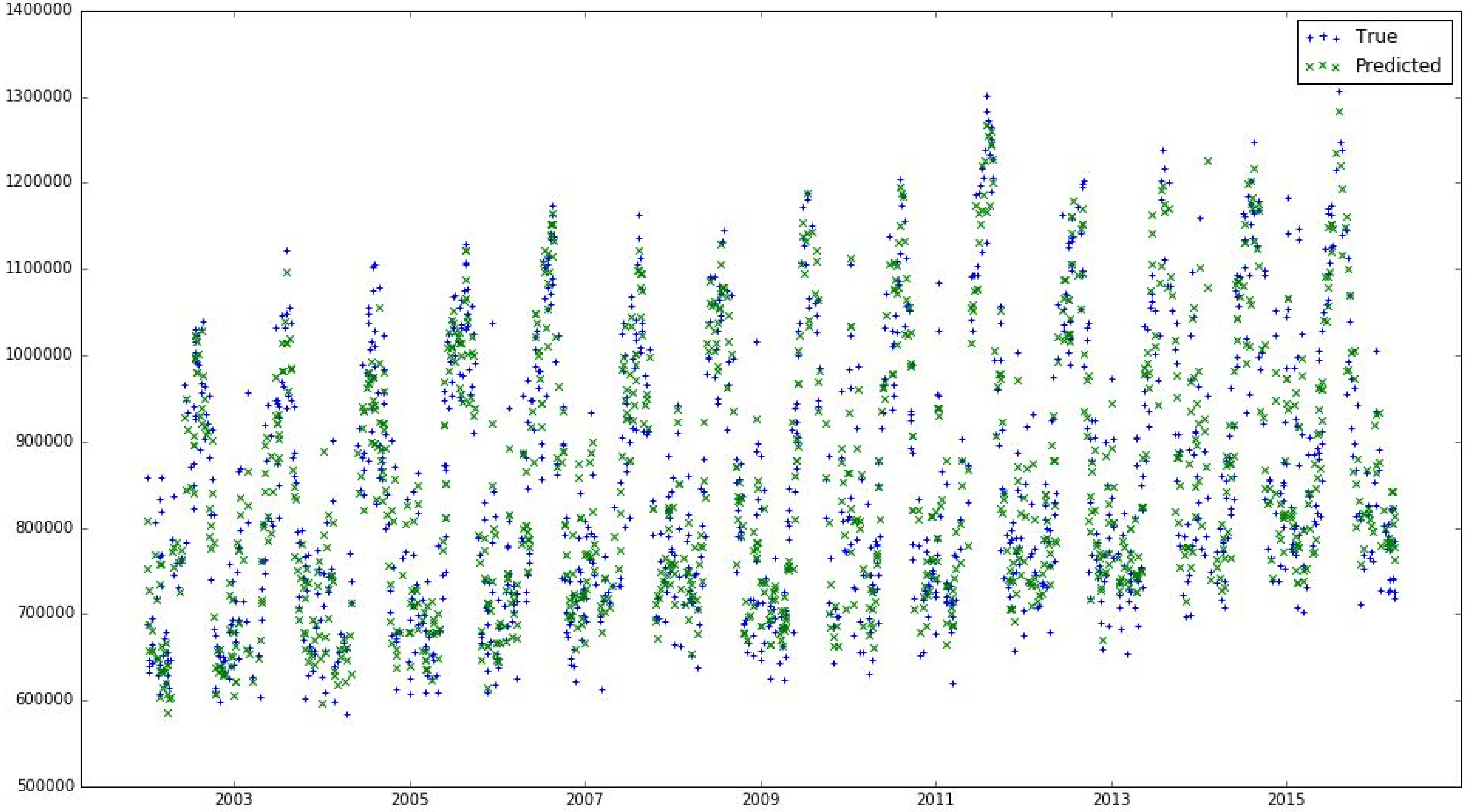
scikit learn

- Fit and evaluate models

Energy usage model

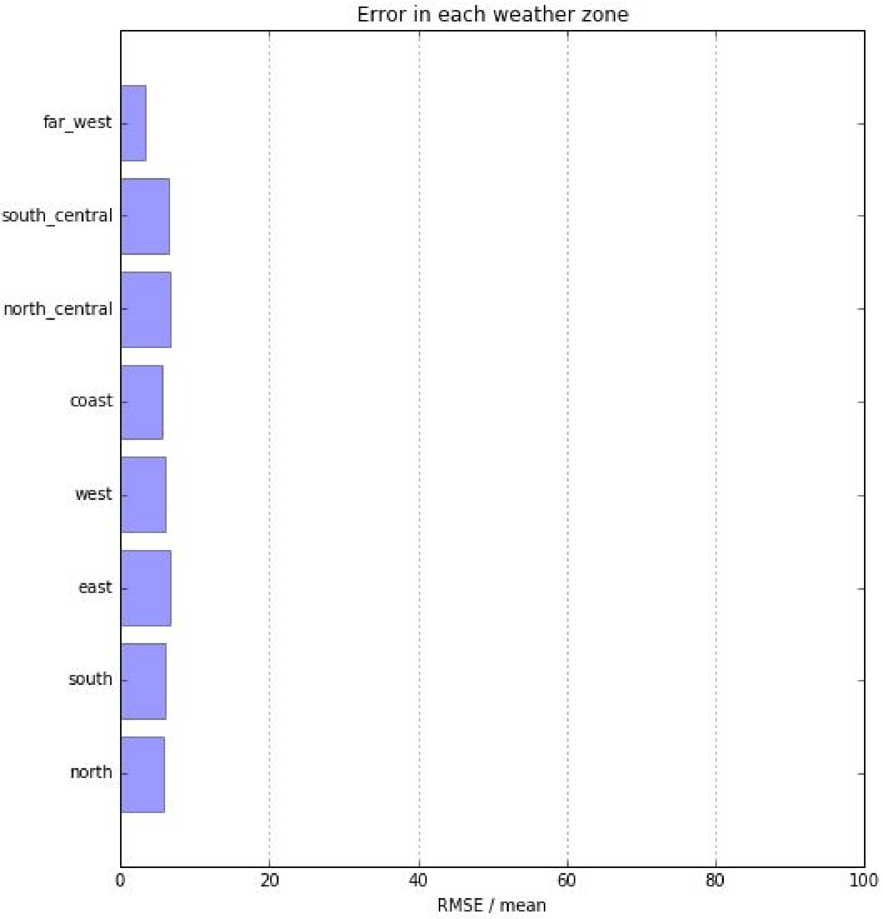
- Criteria
  - Date
  - Weather
    - Max/min/avg temp
    - Precipitation, snow depth
    - Wind speed
    - Water evaporation
    - Total sunshine
  - Holidays

- Target
  - Energy load


1400000
1300000
1200000
1100000
1000000
900000
800000
700000
600000
500000
2003
2005
2007
2009
2011
2013
2015
True
Predicted

Accuracy methodology

- 80/20 train/test split
- RMSE of predicted vs. actual (over the test set)
- Divide RMSE by mean of test values
  - Normalize to a percentage



Weather zone	Error (RMSE / mean)
North	5.94%
South	5.59%
East	6.66%
West	6.33%
Coast	5.71%
North Central	7.11%
South Central	7.20%
Far West	3.17%

County-level predictions

- No data on actuals
- Multiply zone-level prediction by population percentage of county

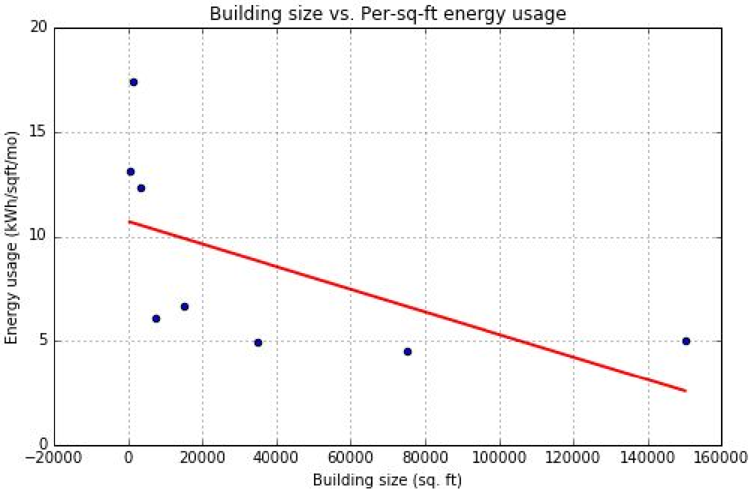
Outline

- Introduction
- Problem
- Solution overview
- **Data modeling**
  - Energy usage
  - **Property energy bill**
- Backend services
- Frontend
- Demo
- Competitive advantages

arcsine

Energy bill model

- Model energy usage as function of building size
- Least squares regression
- Approximation-generalization tradeoff



Data: Building Performance Database, Lawrence Berkeley National Laboratory, United States Department of Energy

arcsine

Outline

- Introduction
- Problem
- Solution overview
- Data modeling
- **Backend services**
- Frontend
- Demo
- Competitive advantages

Energy usage prediction service

```
$ curl -X POST -d'{"location": {"county": "McLennan"}, "date": "2015-02-14"}' $URL
{
  "error": {},
  "result": {
    "date": "2015-02-14",
    "location": {
      "county": "mclennan"
    },
    "value": 237301.16503092414
  },
  "unit": "MWh"
}
```

Energy bill service

```
$ curl -X POST -d'{"square_feet": 1000}' $URL
{
  "error": {},
  "result": 933.4258796934,
  "unit": "USD"
}
```

Historical data service

```
$ curl $URL/south/2002-01-01
{
  "date": "2002-01-01",
  "unit": "MWh",
  "usage": 53817.17388275522,
  "weather_zone": "south"
}
```

docker

- All services and the frontend deployed as containers
- Independently scalable, testable, maintainable
- Solves dependency conflicts
- Starting the application is a one-liner:

```
$ docker-compose up
```

APACHE HBASE

- Run-time access of weather data

Flask

web development,
one drop at a time

- Python micro-framework
- Basis for microservices

NGINX

- Lightweight, scalable
- HTTP server
- Reverse proxy

gunicorn

- Lightweight, scalable
- Python WSGI server

arcsine

Outline

- Introduction
- Problem
- Solution overview
- Data modeling
- Backend services
- **Frontend**
- Demo
- Competitive advantages

arcsine

NGINX

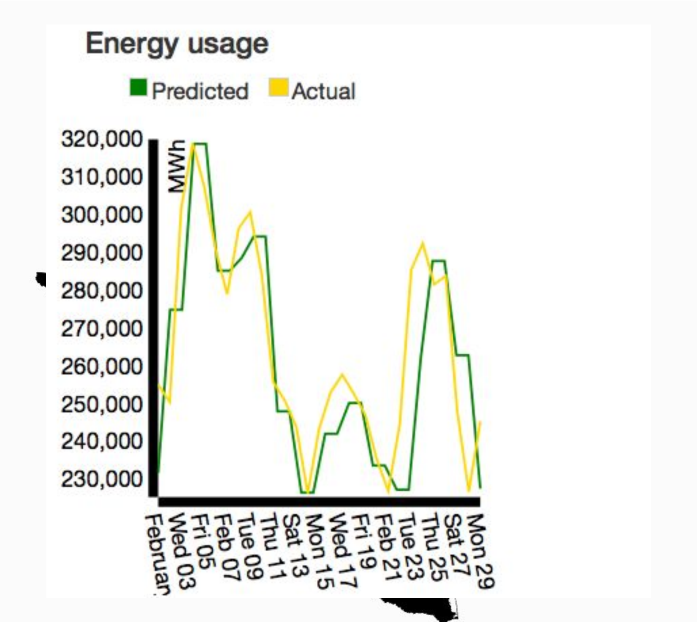
Flask
web development,
one drop at a time

gunicorn

jQuery

B

- Data visualization in JavaScript
- Our uses:
  - Geographical map
  - Actuals vs. predictions



Outline

- Introduction
- Problem
- Solution overview
- Data modeling
- Backend services
- Frontend
- **Demo**
- Competitive advantages

arcsine

Outline

- Introduction
- Problem
- Solution overview
- Data modeling
- Backend services
- Frontend
- Demo
- **Competitive advantages**

arcsine

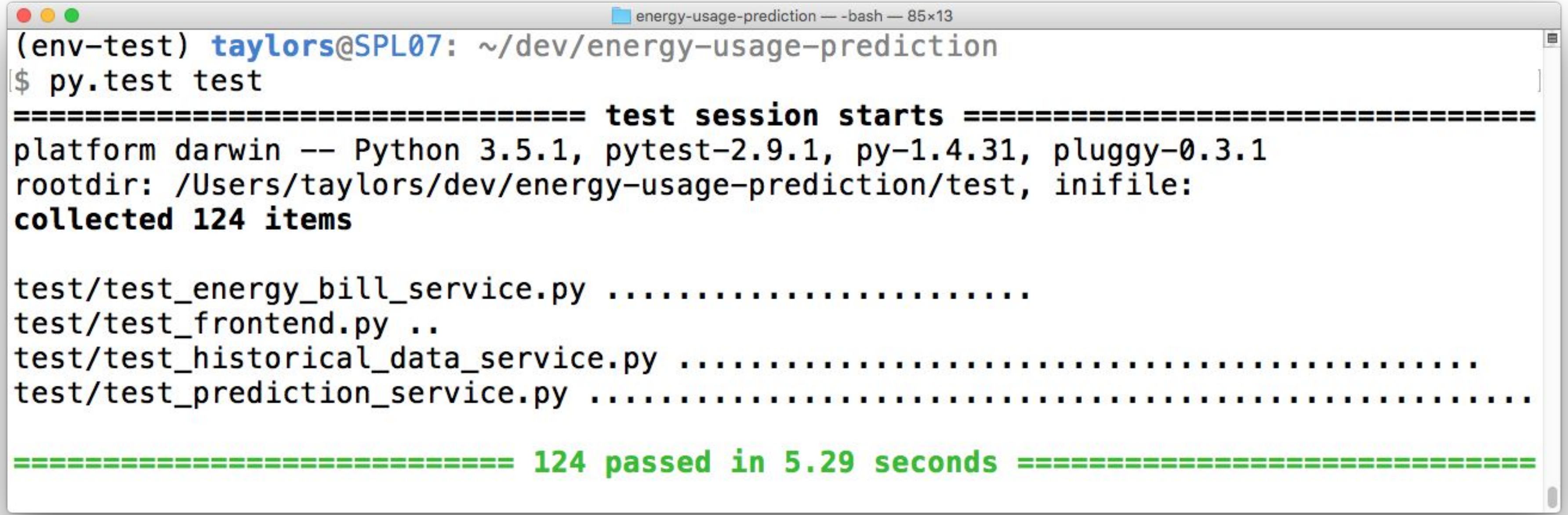
Architecture – Microservices

- High cohesion
- Low coupling
- Testable
- Maintainable
- Scalable

arcsine

pytest

Automated testing



```
(env-test) taylors@SPL07: ~/dev/energy-usage-prediction
$ py.test test
================================ test session starts ================================
platform darwin -- Python 3.5.1, pytest-2.9.1, py-1.4.31, pluggy-0.3.1
rootdir: /Users/taylors/dev/energy-usage-prediction/test, inifile:
collected 124 items

test/test_energy_bill_service.py ........................
test/test_frontend.py ..
test/test_historical_data_service.py .............................................
test/test_prediction_service.py .....................................................

============================ 124 passed in 5.29 seconds =============================
```

arcsine

Why us?

- Flexible
  - Loose coupling
- Scalable
  - Scale services independently
  - Hadoop
- Maintainable
  - Well-tested
  - Microservices easy to understand, change

arcsine

ENERGEEZE + CBRE

+ arcsine