Innovationlab Big Data Science

Energeeks - ASHRAE Great Energy Predictor III

Adrian Uffmann
Dario Lepke
Erjona Dervishi
Tobias Weber

February 21, 2020

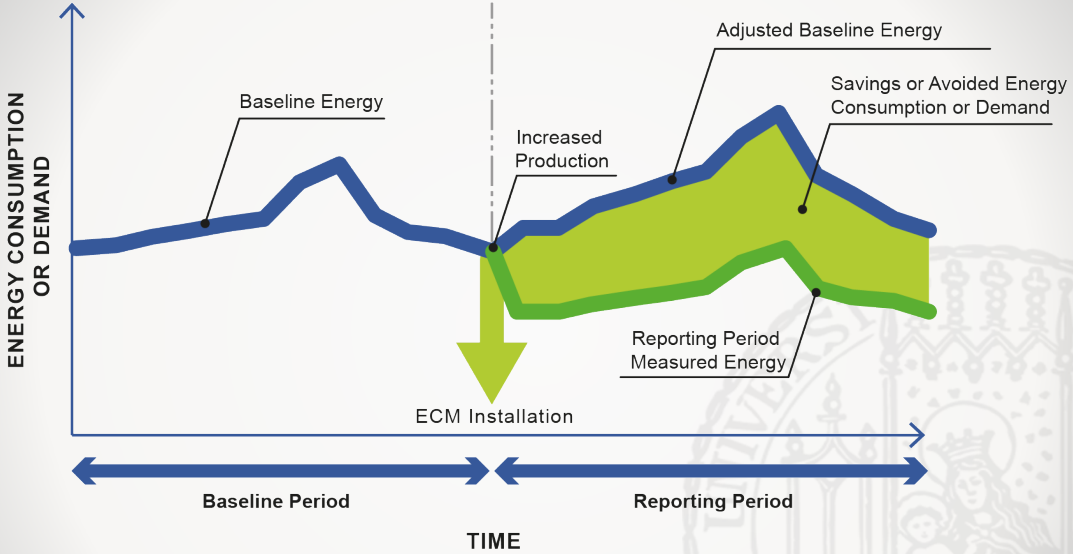
Institut for Statistics
Ludwig-Maximilians-University of Munich

Agenda

1. Phase I - Kaggle Challenge
2. Phase II - App Development
3. Retrospective

Phase I - Kaggle Challenge

The Challenge: ASHRAE Great Energy Predictor III



The Dataset

- Training Data
  - Contains $\approx$ 20 mio. rows
  - Timespan: 1 year (2016)
- Test Data
  - Contains $\approx$ 40 mio. rows
  - Timespan: 2 years (2017 & 2018)
- Target
  - Hourly readings from four different meters (kWh)
    $\rightarrow$ electricity, chilledwater, steam, hotwater
- Features:
  - train.csv: building_id, timestamp, meter, meter_reading
  - building_metadata.csv: site_id, building_id, primary_use, square_feet, year_built, floor_count
  - weather_train.csv: site_id, timestamp, air_temperature, wind_direction, ...

Dataframe.head()

```
In [5]: train.head()
```

Out[5]:

	building_id	meter	timestamp	meter_reading
0	0	0	2016-01-01 00:00:00	0.0
1	1	0	2016-01-01 00:00:00	0.0
2	2	0	2016-01-01 00:00:00	0.0
3	3	0	2016-01-01 00:00:00	0.0
4	4	0	2016-01-01 00:00:00	0.0

```
In [4]: metadata.head()
```

Out[4]:

	site_id	building_id	primary_use	square_feet	year_built	floor_count
0	0	0	Education	7432	2008.0	NaN
1	0	1	Education	2720	2004.0	NaN
2	0	2	Education	5376	1991.0	NaN
3	0	3	Education	23685	2002.0	NaN
4	0	4	Education	116607	1975.0	NaN

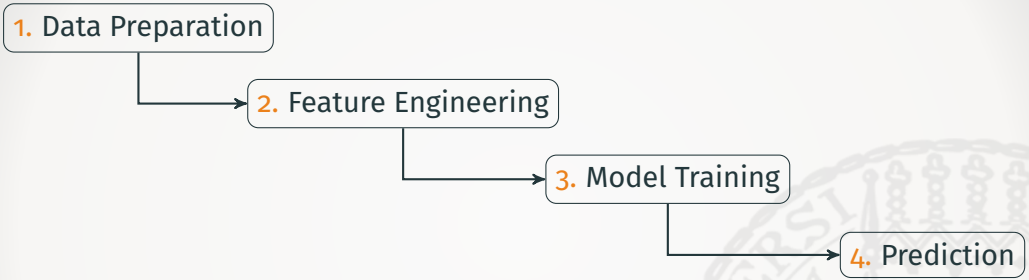
```
In [7]: weather.head()
```

Out[7]:

	site_id	timestamp	air_temperature	cloud_coverage	dew_temperature	precip_depth_1_hr	sea_level_pressure	wind_direction	wind_speed
0	0	2016-01-01 00:00:00	25.0	6.0	20.0	NaN	1019.7	0.0	0.0
1	0	2016-01-01 01:00:00	24.4	NaN	21.1	-1.0	1020.2	70.0	1.5
2	0	2016-01-01 02:00:00	22.8	2.0	21.1	0.0	1020.2	0.0	0.0
3	0	2016-01-01 03:00:00	21.1	2.0	20.6	0.0	1020.1	0.0	0.0
4	0	2016-01-01 04:00:00	20.0	2.0	20.0	-1.0	1020.0	250.0	2.6

Our Project

Pipeline Overview

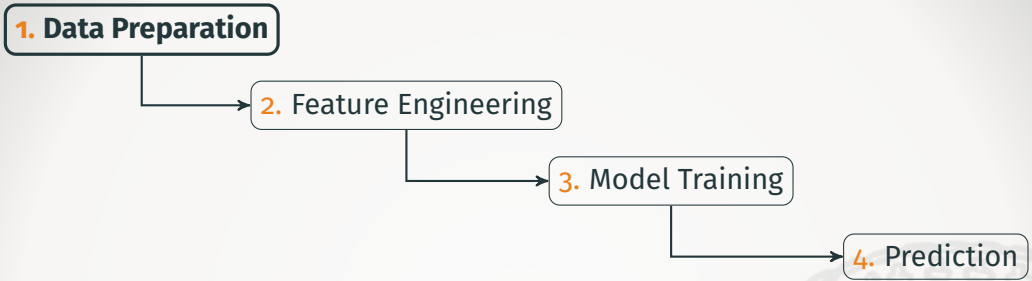


Project Structure

Using Cookiecutter: `https://drivendata.github.io/cookiecutter-data-science/`

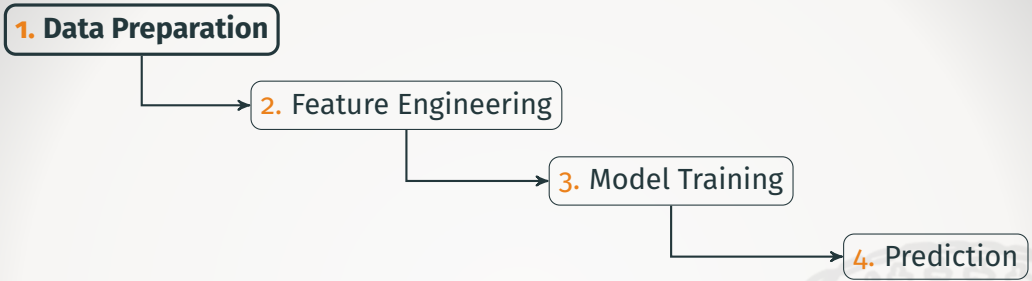
```
+-- data
|   +-- external                <- Data from third party sources.
|   +-- interim                 <- Intermediate data that has been transformed.
|   +-- processed               <- The final, canonical data sets.
|   +-- raw                     <- The original, immutable data dump.
|
+-- src
    +-- data
    |   +-- make_dataset.py     <- Data preparation.
    |
    +-- features
    |   +-- build_features.py   <- Feature engineering.
    |
    +-- models
        +-- predict_model.py    <- Prediction.
        +-- train_model.py      <- Model Training.
```

Pipeline Overview



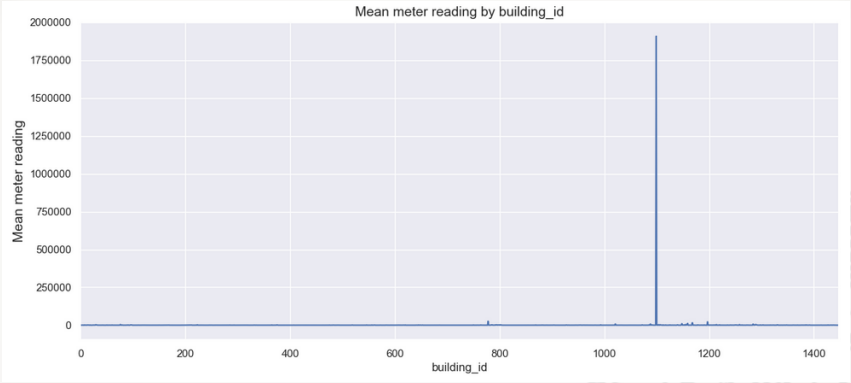
- Impute missing data.
- Exclude faulty readings.
- Align timestamps.
- Merge data frames.

Pipeline Overview

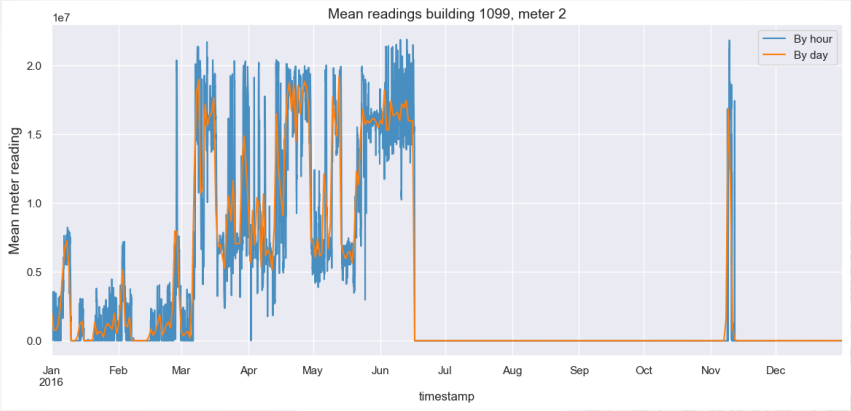


- Impute missing data.
- **Exclude faulty readings.**
- Align timestamps.
- Merge data frames.

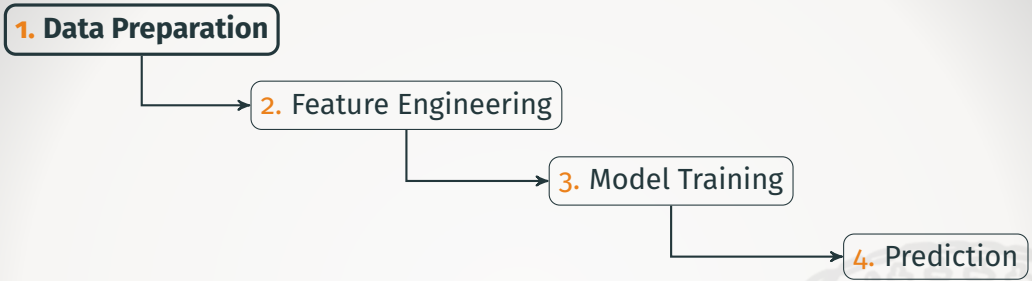
Exclude faulty readings



Exclude faulty readings

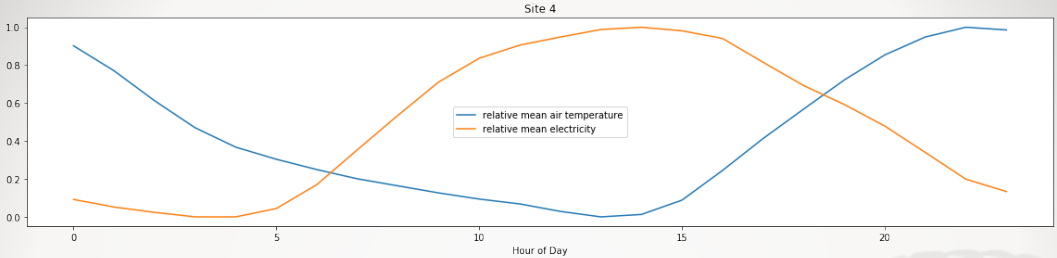


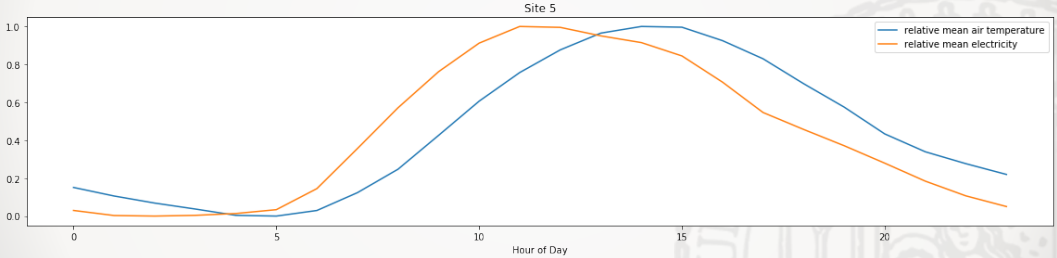
Pipeline Overview



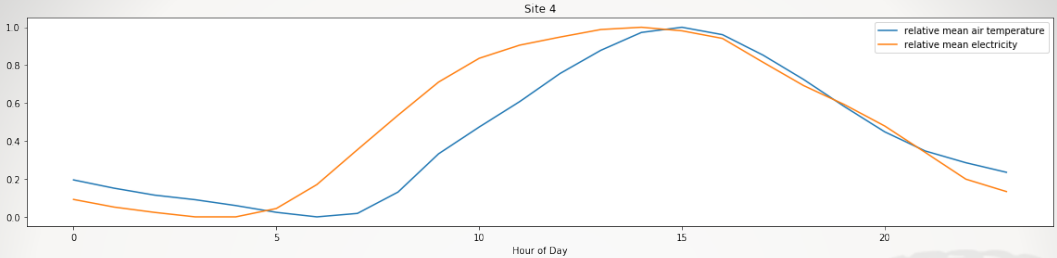
- Impute missing data.
- Exclude faulty readings.
- **Align timestamps.**
- Merge data frames.

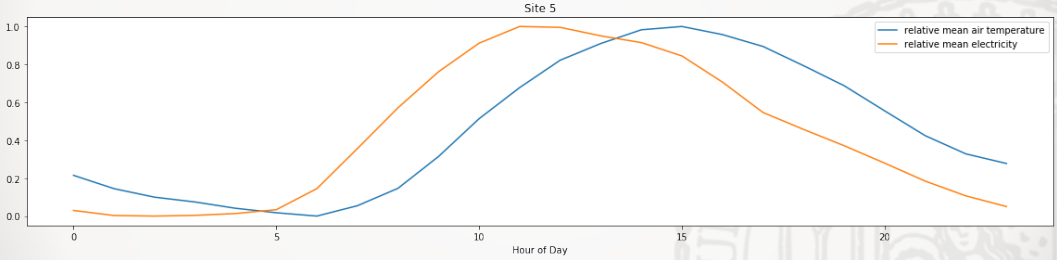
Without Timestamp-Alignment



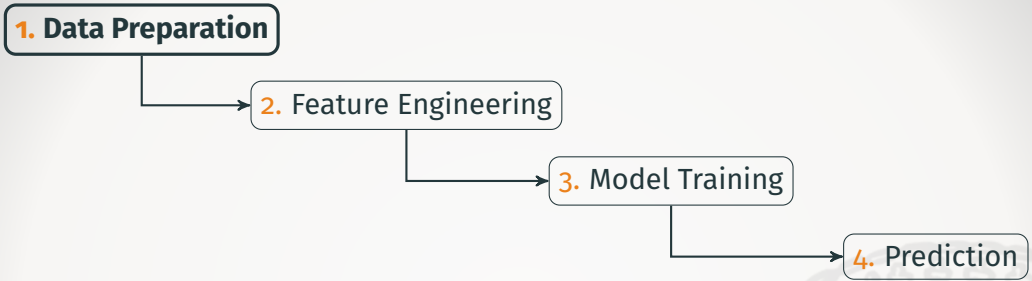


With Timestamp-Alignment



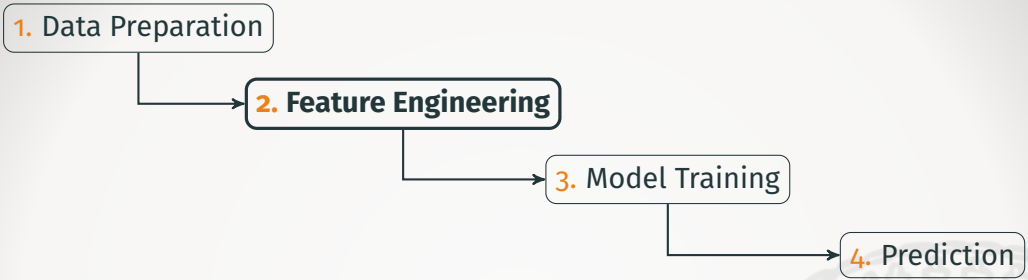


Pipeline Overview



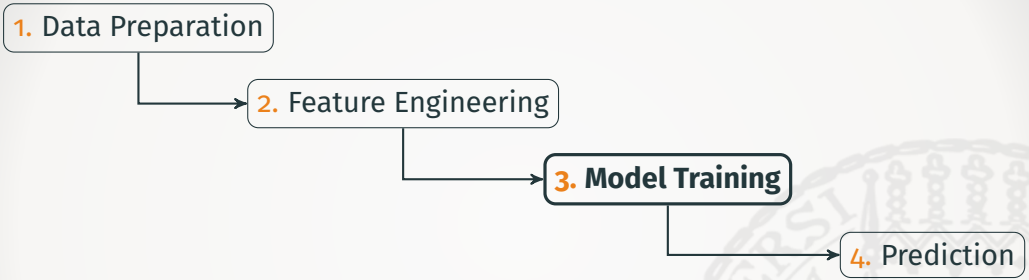
- Impute missing data.
- Exclude faulty readings.
- Align timestamps.
- Merge data frames.

Pipeline Overview



- Encode categorical data.
- Transform year_built into age.
- Logarithmic scaling of square_feet.
- Add lag features.
- Encode cyclic data.

Pipeline Overview



Finding the best model...

Framework

- Focus on Boosting
- Best experiences with LightGBM
- But also tried XGBoost & CatBoost

Finding the best model...

Framework

- Focus on Boosting
- Best experiences with LightGBM
- But also tried XGBoost & CatBoost

Hyperparameter

- Finding optimal setting via Hyperopt
- High number of leaves and low estimators.

Finding the best model...

Framework

- Focus on Boosting
- Best experiences with LightGBM
- But also tried XGBoost & CatBoost

Hyperparameter

- Finding optimal setting via Hyperopt
- High number of leaves and low estimators.

CV Strategy

- Evaluated varying split sizes
- Additionally: Training per meter or per building

Finding the best model...

Framework

- Focus on Boosting
- Best experiences with LightGBM
- But also tried XGBoost & CatBoost

Hyperparameter

- Finding optimal setting via Hyperopt
- High number of leaves and low estimators.

CV Strategy

- Evaluated varying split sizes
- Additionally: Training per meter or per building

Validation

- Local CV
- Daily submissions to challenge

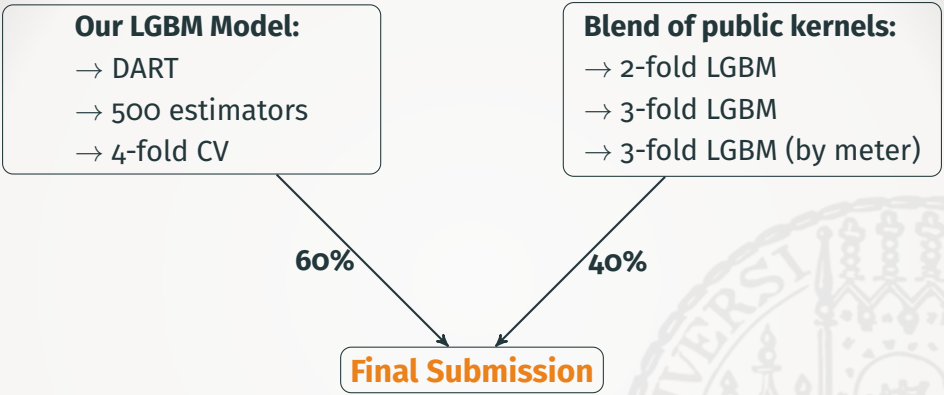
Leaks & a broken Leaderboard

- Identification of sites and buildings via timestamp
- Public availability of energy consumption

→ Scraping of 1 mio. test labels

→ Data Science competition ⇒ Web Scraping competition

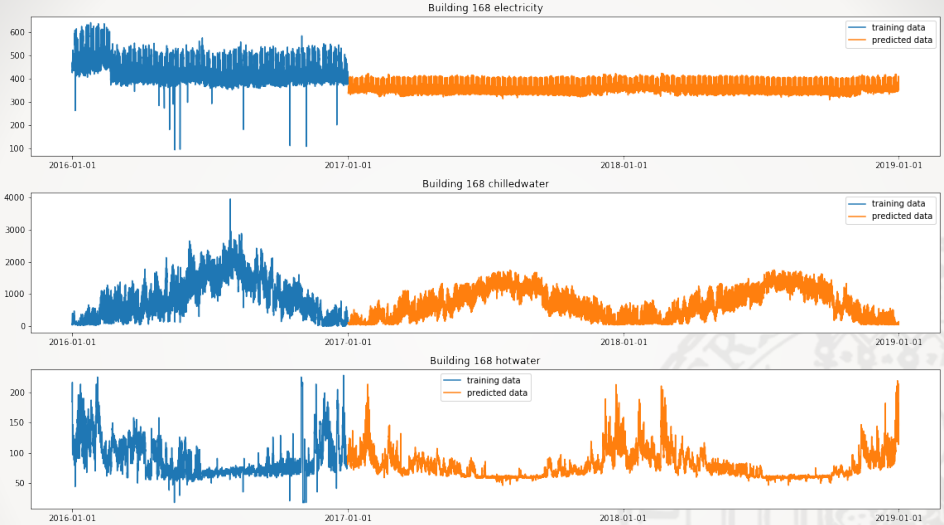
Final Model - Stacking



Pipeline Overview

1. Data Preparation
2. Feature Engineering
3. Model Training
4. **Prediction**

Final Model - Prediction



Leaderboard Placement

Public Leaderboard:

→ Place: 159th

→ Score: 0.953

→ Top 4.3%

Leaderboard Placement

Public Leaderboard:

→ Place: 159th

→ Score: 0.953

→ Top 4.3%

Private Leaderboard:

→ Place: 507th

→ Score: 1.287

→ Top 14%

Leaderboard Placement

Public Leaderboard:

→ Place: 159th

→ Score: 0.953

→ Top 4.3%

Private Leaderboard:

→ Place: 507th

→ Score: 1.287

→ Top 14%

Why did we do so bad? :(

- Private set contains only 11% of whole test set
- No leaked buildings were included in the private set at all

→ Very biased sampling from test pool

→ Focus on non-leaked buildings would have been key

Phase II - App Development

App: Basic Idea

- Embed pipeline and model into a web-application
- Users can manage and add buildings
  - → User gives building metadata and location
  - → App predicts and visualizes future energy consumption
- Get weather forecasts from a weather API
- Also: Make our predictions directly available via REST API

Weather API

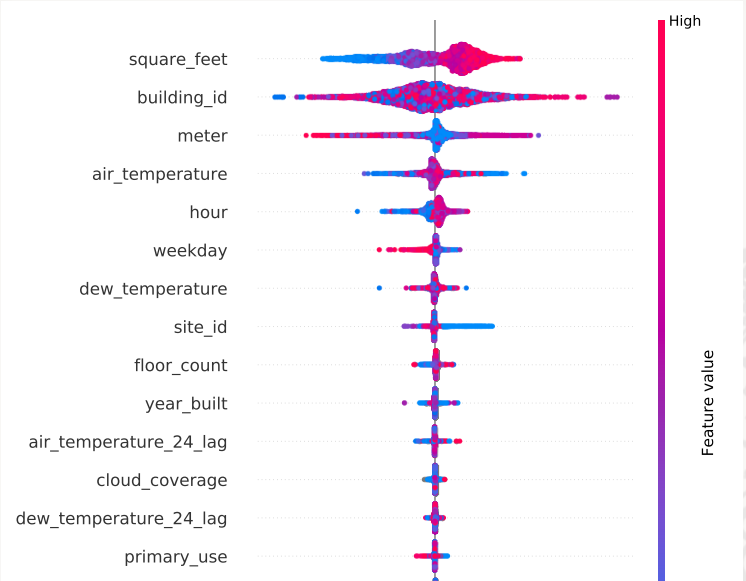
- Criteria: Forecast length and resolution, free calls per day/minute
- We use OpenWeathermap
  - → 5 day forecast, 3 hour resolution, 60 calls/minute
- Cache API calls and refresh after 60 seconds

Challenge Model vs App Model

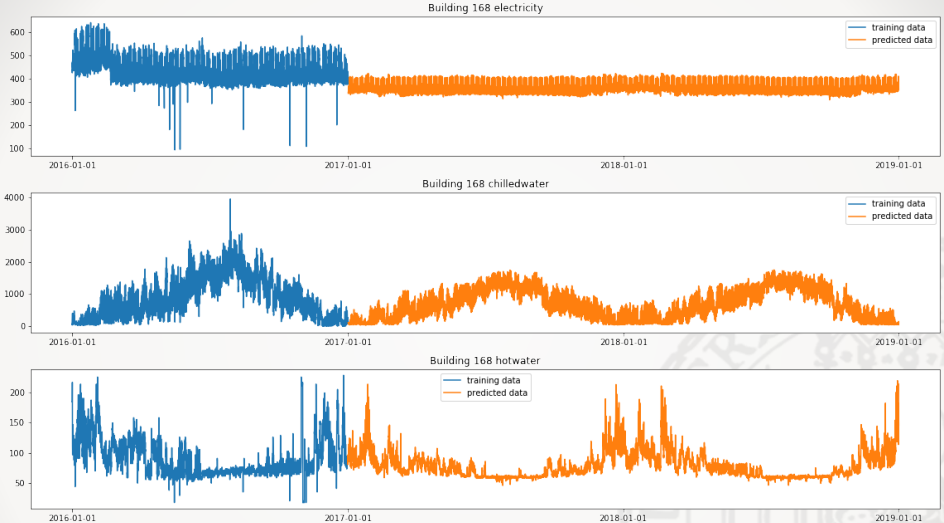
Challenge Model vs App Model

	Challenge	App
Building_id	Yes	No
Site_id	Yes	No
Weather	Historical	Forecast

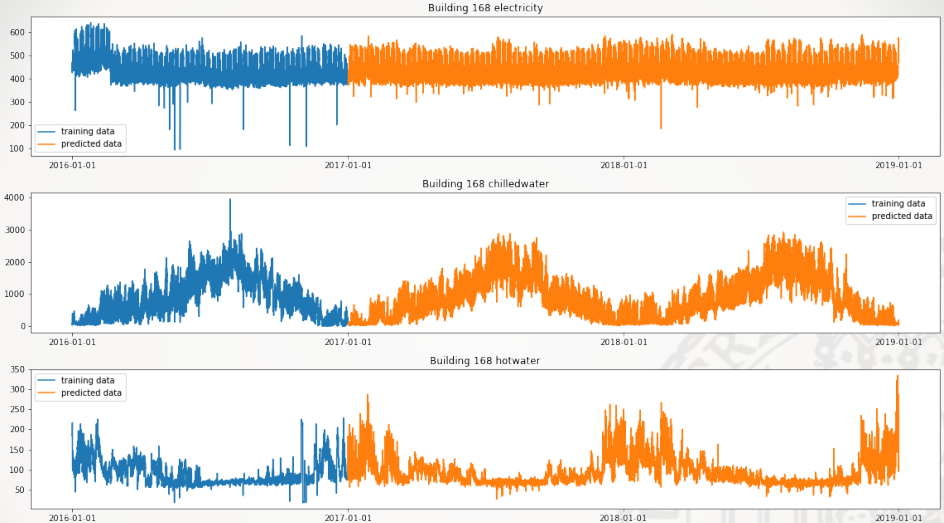
SHAP Values for the Challenge Model



Prediction with site_id and building_id



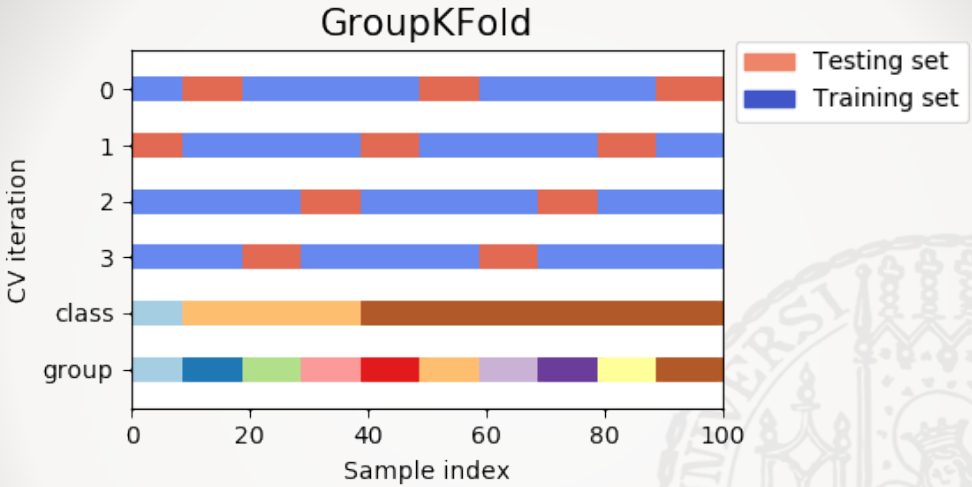
Prediction without site_id and building_id



Challenge Model vs App Model

	Challenge	App
Building_id	Yes	No
Site_id	Yes	No
Weather	Historical	Forecast
Public LB	1.065	1.111
Private LB	1.316	1.337
CV	KFold	GroupKFold

Grouped Kfold Cross-Validation



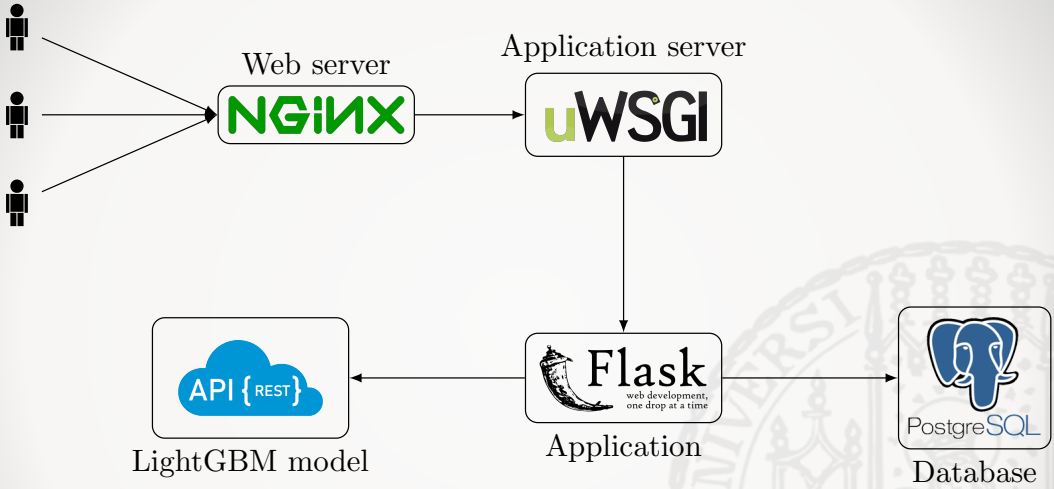
Model: Kaggle Challenge vs App

	Challenge	App
Public LB	1.065	1.111
Private LB	1.316	1.337
CV	KFold	GroupKFold
CV-Result	0.81	2.3

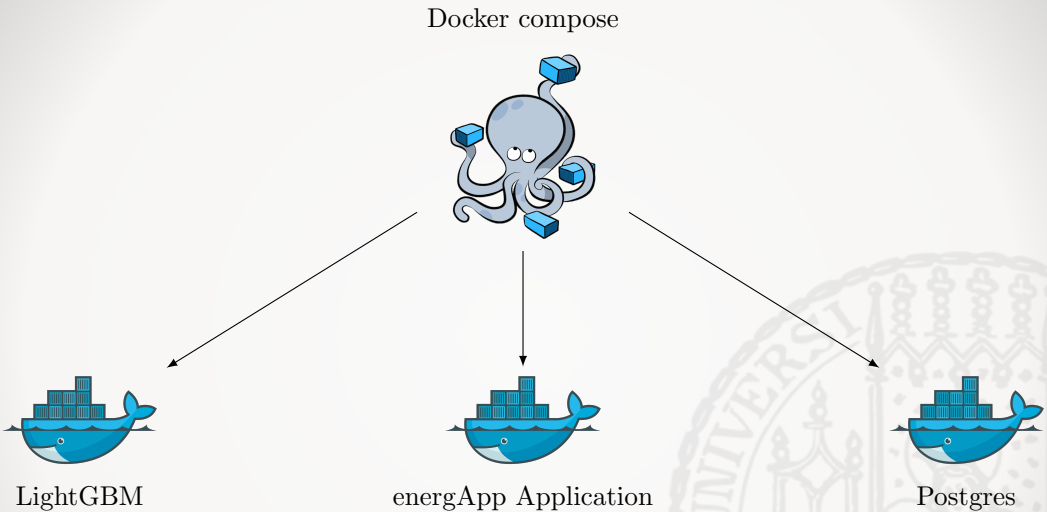
- Take results with grain of salt, not directly comparable:
  - LB and KFold Scores measure performance on known buildings and sites
  - GroupKFold measures performance on buildings not seen in training data
- Takeaway: Generalisation on unknown buildings is difficult task
- For optimal results retrain on own buildings and use app for managing and visualization

App Architecture

App Architecture & Software Stack

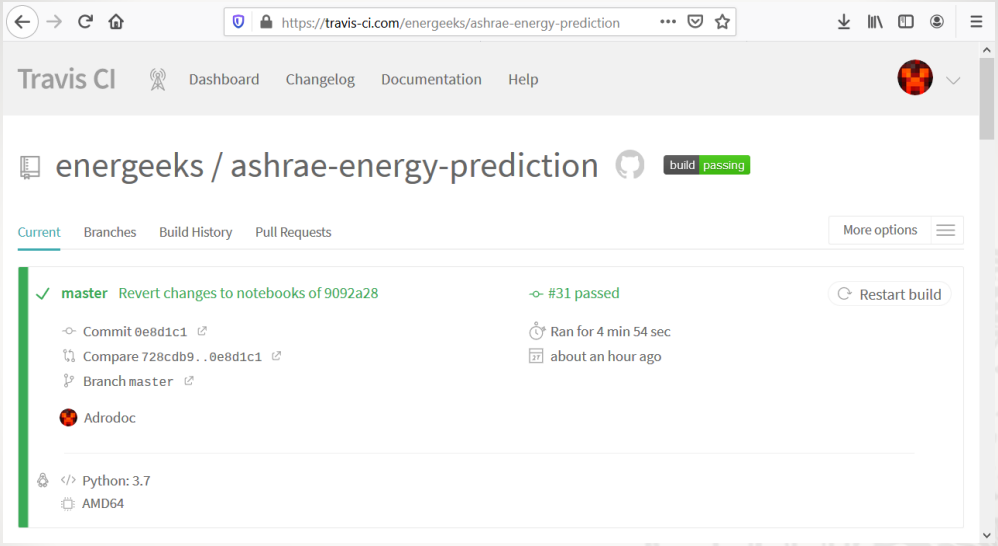


Deployment as Microservices via Docker



Automated Testing

Travis


https://travis-ci.com/energeeks/ashrae-energy-prediction
Travis CI
Dashboard
Changelog
Documentation
Help
energeeks / ashrae-energy-prediction
build passing
Current
Branches
Build History
Pull Requests
More options
master Revert changes to notebooks of 9092a28
#31 passed
Restart build
Commit 0e8d1c1
Compare 728cdb9..0e8d1c1
Branch master
Ran for 4 min 54 sec
about an hour ago
Adrodoc
Python: 3.7
AMD64

Unit-Tests

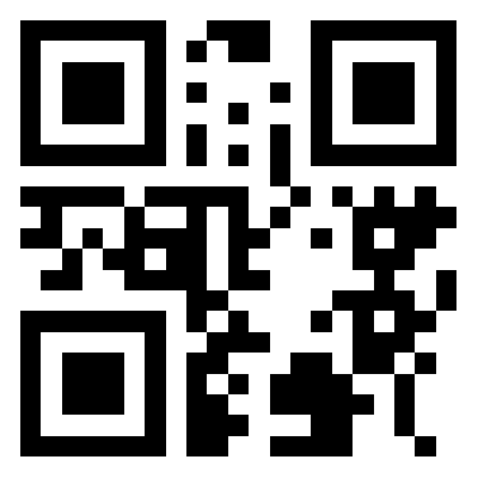
```
def test_incorrect_column_names(self):
    # given:
    df = pd.DataFrame({
        "row": np.arange(41697600),
        "meter_reading": np.zeros(41697600),
    })

    # when:
    actual = get_submission_error(df)

    # then:
    expected = "Submission has incorrect columns: ['row', 'meter_reading'],\
    expected: ['row_id', 'meter_reading']"
    self.assertEqual(expected, actual)
```

End2End-Tests

```
def test(self):
    # given:
    data_dir = "test-data"
    interim_dir = self.test_dir + "/interim"
    processed_dir = self.test_dir + "/processed"

    # when:
    make_dataset(data_dir, interim_dir)

    # then:
    self.assertTrue(os.path.exists(interim_dir + "/test_data.pkl"))
    self.assertTrue(os.path.exists(interim_dir + "/test_data.pkl"))

    # when:
    build_features(data_dir, processed_dir)

    # then:
    self.assertTrue(os.path.exists(processed_dir + "/test_data.pkl"))
    self.assertTrue(os.path.exists(processed_dir + "/test_data.pkl"))
```

App Demo

Check out the EnergApp!



http://138.246.234.9/

Retrospective

Collaboration

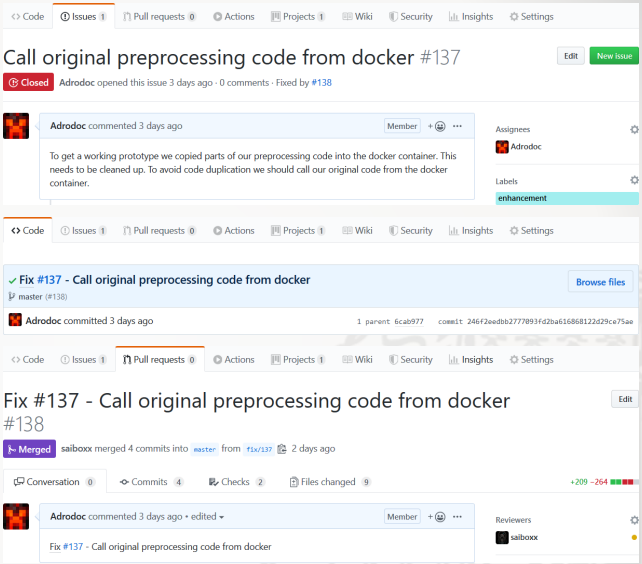
https://github.com/energeeks/ashrae-energy-prediction

Development Workflow & Naming Conventions

Issue #137

Commit on branch fix/137 with message ”Fix #137 - ...”

Pull Request with review



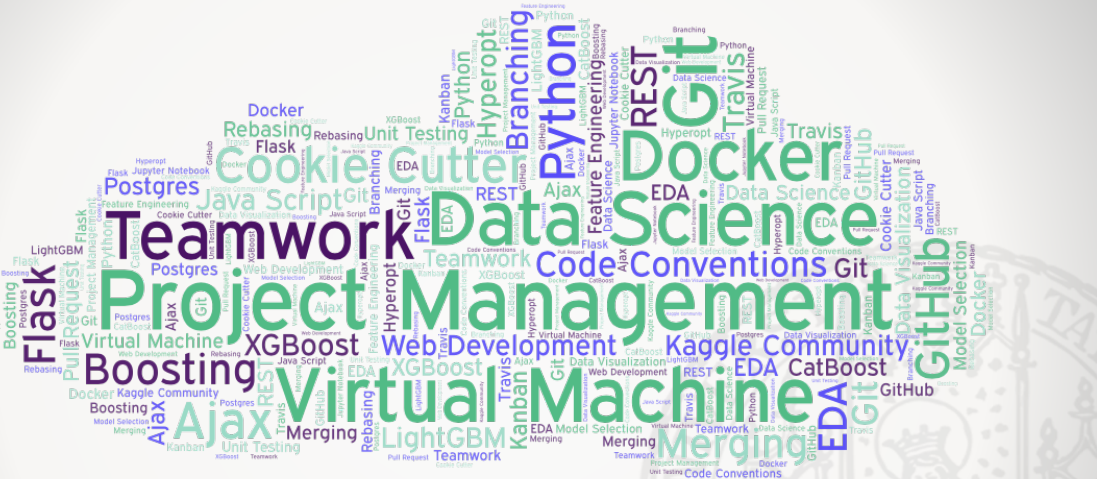
Problems

- Underestimation of tickets
- Sprints were a bit difficult to plan (constantly had new ideas)
- Virtual Machine management
  - Connection problems
  - Only 4 SSH Keys on Website
  - Sometimes needs Proxy Jump via CIP-Pool
- Merging teams too early $\rightarrow$ only 2 instead of 8 daily Kaggle submissions

Success stories

- Ticket system (Kanban Board)
- Team members have different backgrounds and strengths
- Jupyter notebooks
- Makefiles & Configs
- Data Pipeline Architecture

Takeaways


Project Management
Data Science
Teamwork
Virtual Machine
Cookie Cutter
Docker
Git
Python
Flask
Boosting
Ajax
GitHub
Code Conventions
Web Development
Kaggle Community
Java Script
Postgres
Rebasing
Unit Testing
Jupyter Notebook
Feature Engineering
Cookie Cutter
Data Visualization
Model Selection
Pull Request
Merging
Branching
Hyperopt
Kanban
Travis
REST
XGBoost
LightGBM
CatBoost
EDA

Contributors

Team EnerGeek

Adrian
- Proj. Setup & Workflow
- Data Pipeline
- Automated Tests
- Timestamp Alignment
- Cyclic Encoding
- Cross. Valid. Variants
- Model-App Integration

Dario
- Data Cleansing
- Feature Engineering
- Timestamp Alignment
- Weather Api
- Model Evaluation

Erjona
- Expl. Data Analysis
- Preprocessing
- Imputation
- Documentation
- App Front-End

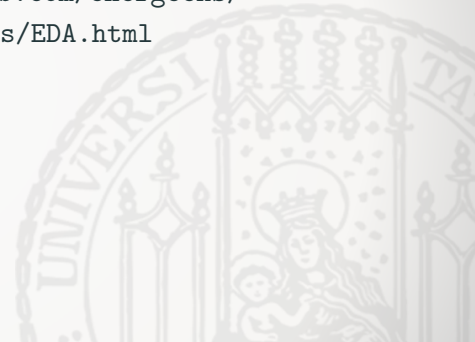
Tobias
- Project Mgmt.
- Data Pipeline
- Model Training
- Model Selection
- App Architecture
- App Front-End
- App Back-End

Thank you for your attention! :)

Appendix

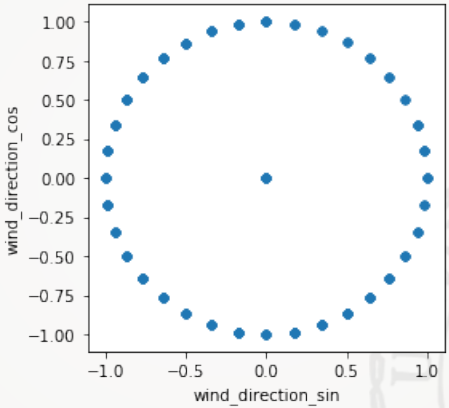
EDA

http://htmlpreview.github.io/?https://github.com/energeeks/ashrae-energy-prediction/blob/master/reports/EDA.html



Cyclic-Encoding

```
df["wind_direction_sin"] = np.sin(2 * np.pi * df["wind_direction"] / 360)
df["wind_direction_cos"] = np.cos(2 * np.pi * df["wind_direction"] / 360)
df.loc[df["wind_direction"].isna(), ["wind_direction_sin", "wind_direction_cos"]] = 0
df.loc[df["wind_speed"] == 0, ["wind_direction_sin", "wind_direction_cos"]] = 0
```



Model Parameters

Parameters	Score: 1.07	Score: 1.06
Boosting Type	GBDT	DART
Early Stopping	YES	NO
Number of Leaves	3480	3630
Learning Rate	0.05	0.05
Training Time	Low	High

SHAP Values for the App Model