Demonstration of Ballet: A Framework for Open-Source Collaborative Feature Engineering

Micah J. Smith, Kelvin Lu, Kalyan Veeramachaneni
Massachusetts Institute of Technology

Introduction and Motivation

Large scale collaborations have been successful in open source software engineering but nothing similar exists in open source data science.

Software engineering		Data science	
Linux kernel	20,000+	drug-spending	20
Ruby on Rails	3,900+	police-eis	19
kubernetes	2,400+	crash-model	18
tensorflow	2,400+	food-inspections	8

Number of unique contributors to largest projects.

Idea: make data science development more like open-source software development!

We draw inspiration from concepts in software engineering to propose a workflow for collaborative data science.

concept	software engineering	feature engineering
patch	• bugfix • software feature	• logical feature
acceptance procedure	• unit test • integration test	• feature test • streaming logical feature selection
product	• application • library	• feature engineering pipeline

Ballet is a lightweight software framework for collaborative, open-source data science through a focus on feature engineering.

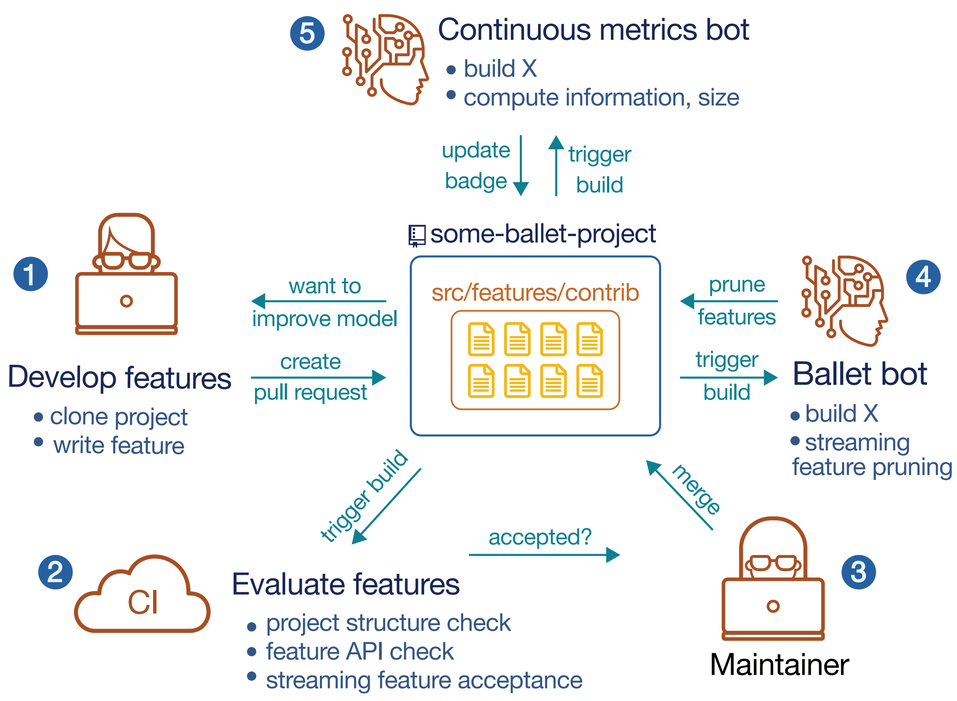
⇒ https://github.com/HDI-Project/ballet

Ballet development workflow

We invite you to join a live, real-time feature engineering collaboration!

⇒ http://bit.ly/ballet-demo

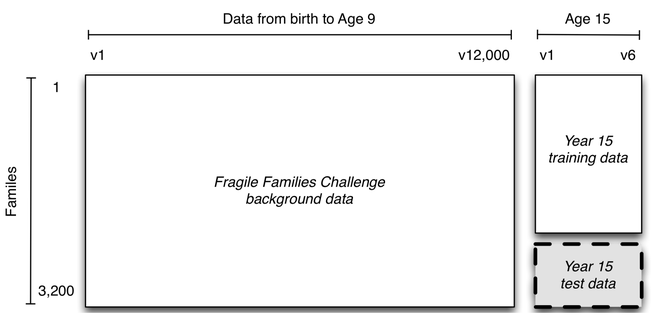
1. **Launch.** Launch repo in interactive Binder environment
2. **Develop.** Develop features in familiar notebook environment
3. **Submit.** Submit features from within notebook



- Submission formulated under the hood as pull request to upstream repo
- Feature extensively validated by continuous integration (CI) service for API requirements (feature tests) and ML performance (streaming logical feature selection)
- `ballet-bot` automatically merges/closes features in response to validation
- Features automatically pruned by CI/ballet-bot (streaming logical feature selection)
- Resulting feature engineering pipeline used as a dependency of a downstream ML model.
- Same development workflow applies to data programming and prediction engineering in addition to feature engineering!

Feature engineering in Ballet

In many real-world problems, feature engineering is a challenging process that requires significant human intuition and expertise. *Fragile Families data is 4,242 rows (one per child) and 12,943 columns.*



The Fragile Families Challenge dataset

Users are responsible for defining an *input* and a *transformer*.

They can use our library of useful feature engineering primitives, or other common libraries like `sklearn.preprocessing`.

```python
from ballet import Feature
from ballet.eng import ConditionalTransformer
import numpy as np
from sklearn.impute import SimpleImputer

input = 'Lot Area'
transformer = [
    ConditionalTransformer(
        lambda ser: ser.skew() > 0.75,
        lambda ser: np.log1p(ser)),
    SimpleImputer(strategy='mean'),
]
name = 'Lot area unskewed'
feature = Feature(input=input, transformer=transformer,
    ↪  name=name)
```

A user-submitted logical feature in Ballet that conditionally unskews the "lot area" variable by applying a log transformation only if skew is present in the training data and then mean-imputing missing values.

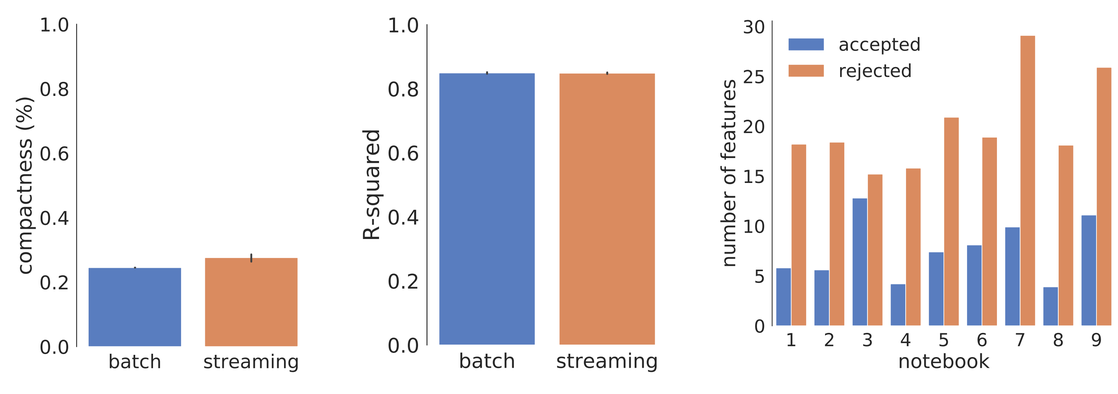
```python
from ballet import Feature
from ballet.eng import NullFiller
import numpy as np


input = ["Garage Area", "Garage Cars"]
transformer = [
    lambda df: df["Garage Area"] / df["Garage Cars"]
    NullFiller(isnull=np.isinf),
    NullFiller(),
]
name = "Garage area per car"
feature = Feature(input=input, transformer=transformer,
    ↪  name=name)
```

A user-submitted logical feature that calculates the garage area per car and cleans infinite and missing values.

Evaluation

Case study: Ames housing price prediction.

- Extract 249 logical features from 9 public notebooks on Kaggle.
- Simulate a scenario in which Kagglers submitted their features to a Ballet project instead.
- Iteratively select random notebook, simulate its submission, and validate using SLFS algorithm.



72.4% of all features are rejected by the feature validation and SLFS algorithm, suggesting substantial work was redundant across notebooks. Every notebook had both accepted and rejected features, suggesting both that everyone had something to contribute to final pipeline but that everyone did redundant work.

References

[1] Fragile families challenge, 2017.

[2] M. J. Smith, K. Lu, and K. Veeramachaneni.
Ballet: A lightweight framework for open-source, collaborative feature engineering.
In *Workshop on Systems for ML at NeuRIPS 2018*, 2018.

[3] M. J. Smith, K. Lu, and K. Veeramachaneni.
Enabling open-source collaborative data science development with the ballet framework.
Preprint, 2020.

{micahs,kelvinlu,kalyanv}@mit.edu