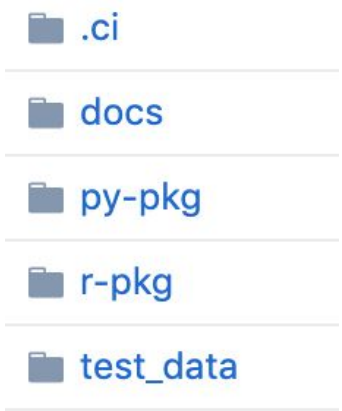
PEOPLE SHAPE SOFTWARE:

WHAT I LEARNED COMPARING

PYTHON AND R APIS

/jameslamb @_jameslamb

Common problem: “Packages in Python and R (at least) that wrap the same underlying thing”



Machine Learning Algorithms

CatBoost

LightGBM

RGF

xlearn

xgboost

Data Processing

pyspark / sparkR

arrow / pyarrow

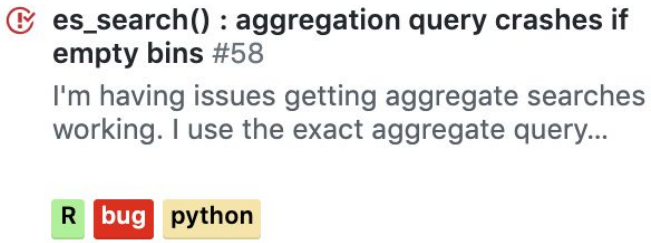
ETL

argparse

feather / pyarrow

uptasticsearch

Having identical public interfaces is desirable!



Scala Java Python R

```python
from pyspark.ml.classification import LogisticRegression

# Load training data
training = spark \
    .read \
    .format("libsvm") \
    .load("data/mllib/sample_multiclass_classification_data.txt")

lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)
```

for developers

bug reporting improves both packages

shared test data

docs only need to be written once

hard to make complicated API changes

for users

no need to re-learn API when switching languages

better docs

less complicated API

But almost no packages actually do this 0_o

argparse

- 0 overlap
- Python is 11 classes, R is one big function

LightGBM

- different default training params
- *Dataset()* object has different param order, different params
- Python feature importance plots have way more features

feather

- R → *read_feather(), write_feather()*
- Python → *read_dataframe(), write_dataframe()*
- no *feather_examples()* in Python

XGBoost

- 0 overlap
- different default training params
- Python is OO, R is functional

There are defensible *technical* reasons for this!

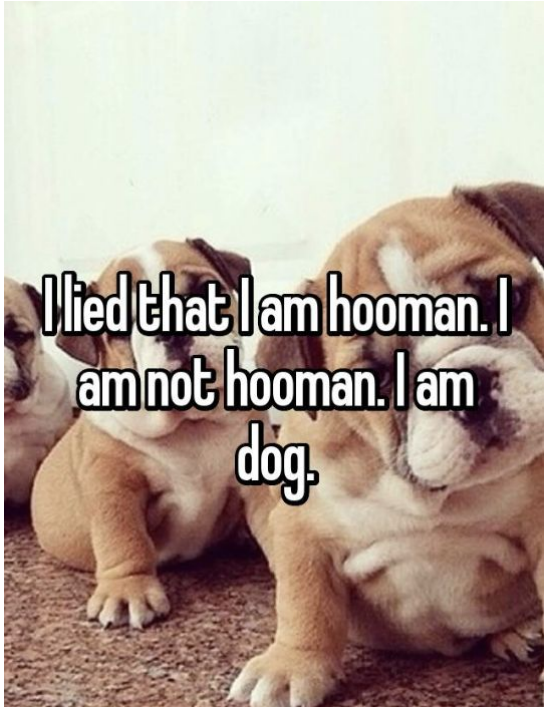
Why Python might differ from R

- scikit-learn interface compatibility
- use of decorators
- OO is easier to reason about

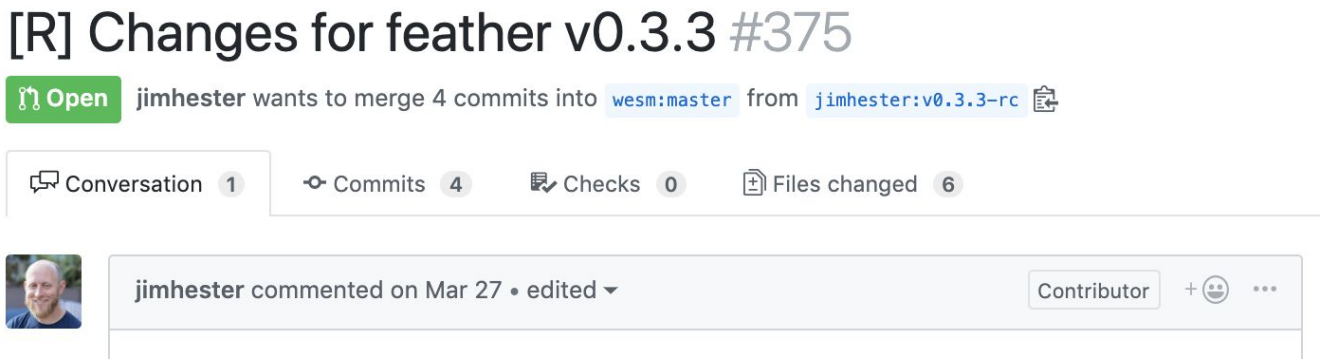
Why R might differ from Python

- **caret** expects S3 method dispatching, e.g. *predict()*
- %>% convention for compatibility with other libraries

A conjecture: most differences are from normal social forces acting on the very human developers doing the things

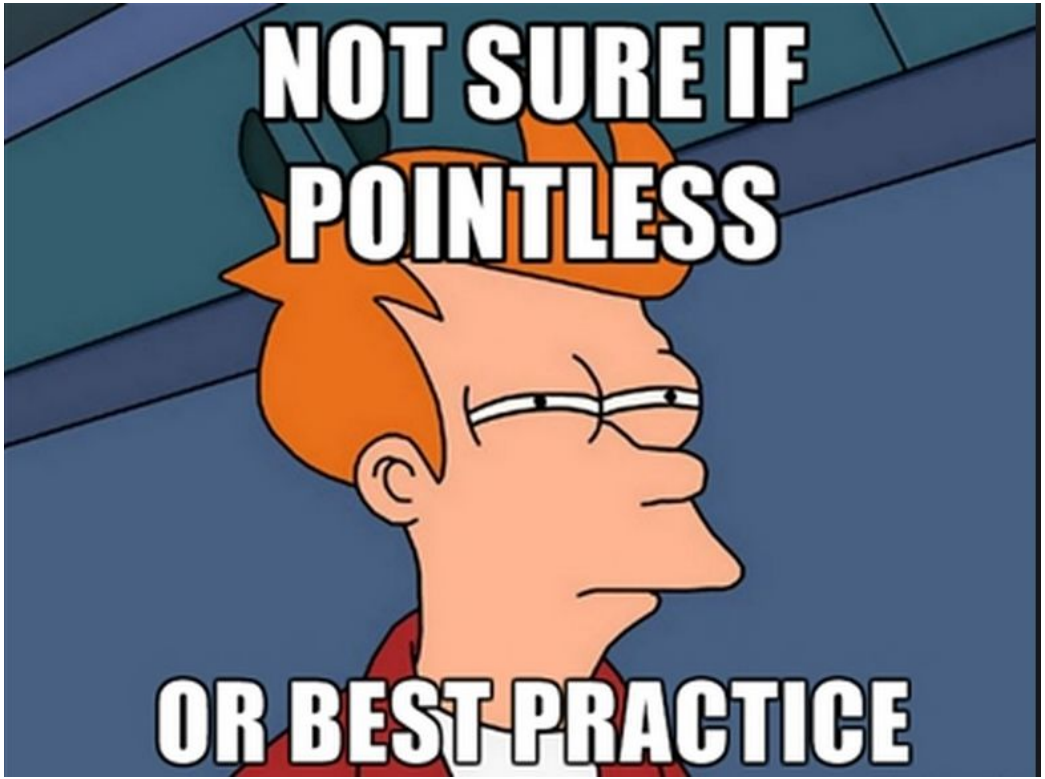


Labeling Theory: “I am an R person, I do the R stuff”

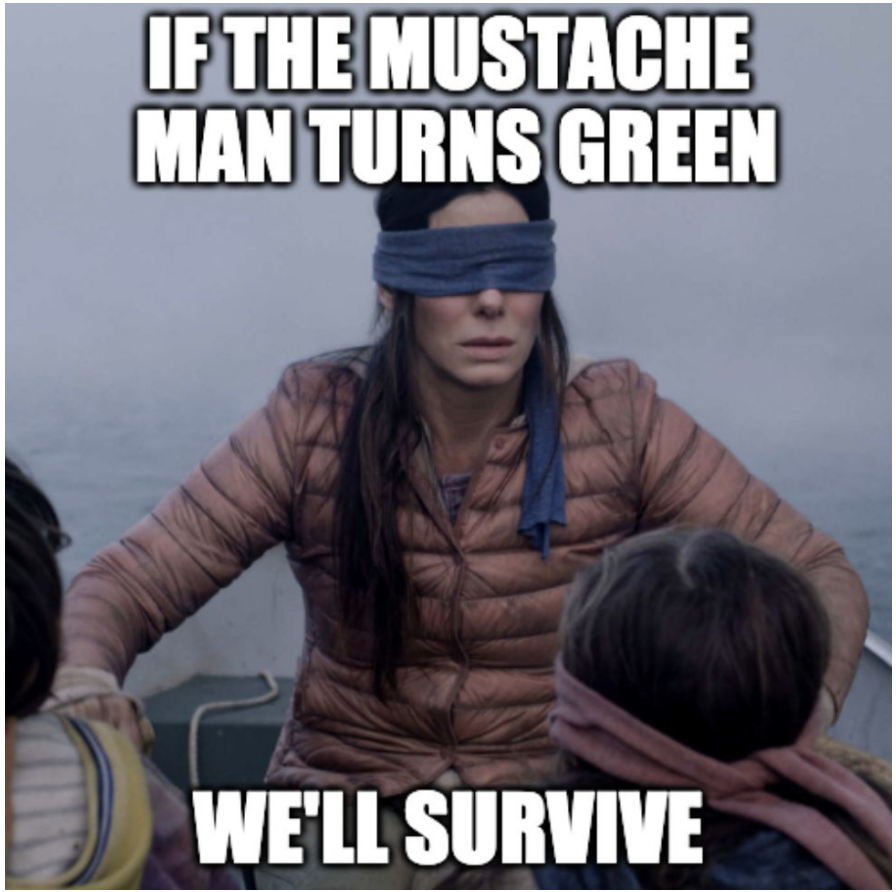


- @guolinke **Guolin Ke** (C++ code / R-package / Python-package)
- @chivee **Qiwei Ye** (C++ code / Python-package)
- @Laurae2 **Damien Soukhavong** (R-package)
- @jameslamb **James Lamb** (R-package)
- @wxchan **Wenxuan Chen** (Python-package)
- @henry0312 **Tsukasa Omoto** (Python-package)
- @StrikerRUS **Nikita Titov** (Python-package)
- @huanzhang12 **Huan Zhang** (GPU support)

Cargo cult programming: “Hadley did it, seems cool”



WYSIATI: “I added stuff to the Python package and the build passed”



build passing

dmlc/xgboost -

One solution: I wrote a piece of software to measure these differences and automatically prevent them in CI

```
Test Failures (12)
====================
1. Function 'ngettext()' is not exported by all packages

2. Function 'ArgumentParser()' is not exported by all packages

3. Packages have different counts of exported classes!
   argparse [python] (9), argparse [r] (0)

4. Class 'HelpFormatter()' is not exported by all packages
```

https://github.com/jameslamb/doppel-cli

```
Function Count
==============
+--------------------+----------------+
|   argparse [python] |   argparse [r] |
+====================+================+
|                  0 |              1 |
+--------------------+----------------+


Function Names
==============
+-----------------+--------------------+----------------+
| function_name   | argparse [python]  | argparse [r]   |
+=================+====================+================+
| ArgumentParser  | no                 | yes            |
+-----------------+--------------------+----------------+


Function Argument Names
=======================
No shared functions.

Class Count
===========
+--------------------+----------------+
|   argparse [python] |   argparse [r] |
+====================+================+
|                  9 |              0 |
+--------------------+----------------+
```

Thanks for your time!