machine learning lessons for geophysics

Leonardo Uieda

Dept. of Earth Sciences - UH Manoa

leouieda.com

Oct 12 2018

Outline

A.I. and Machine Learning

Glossary

Unsupervised learning

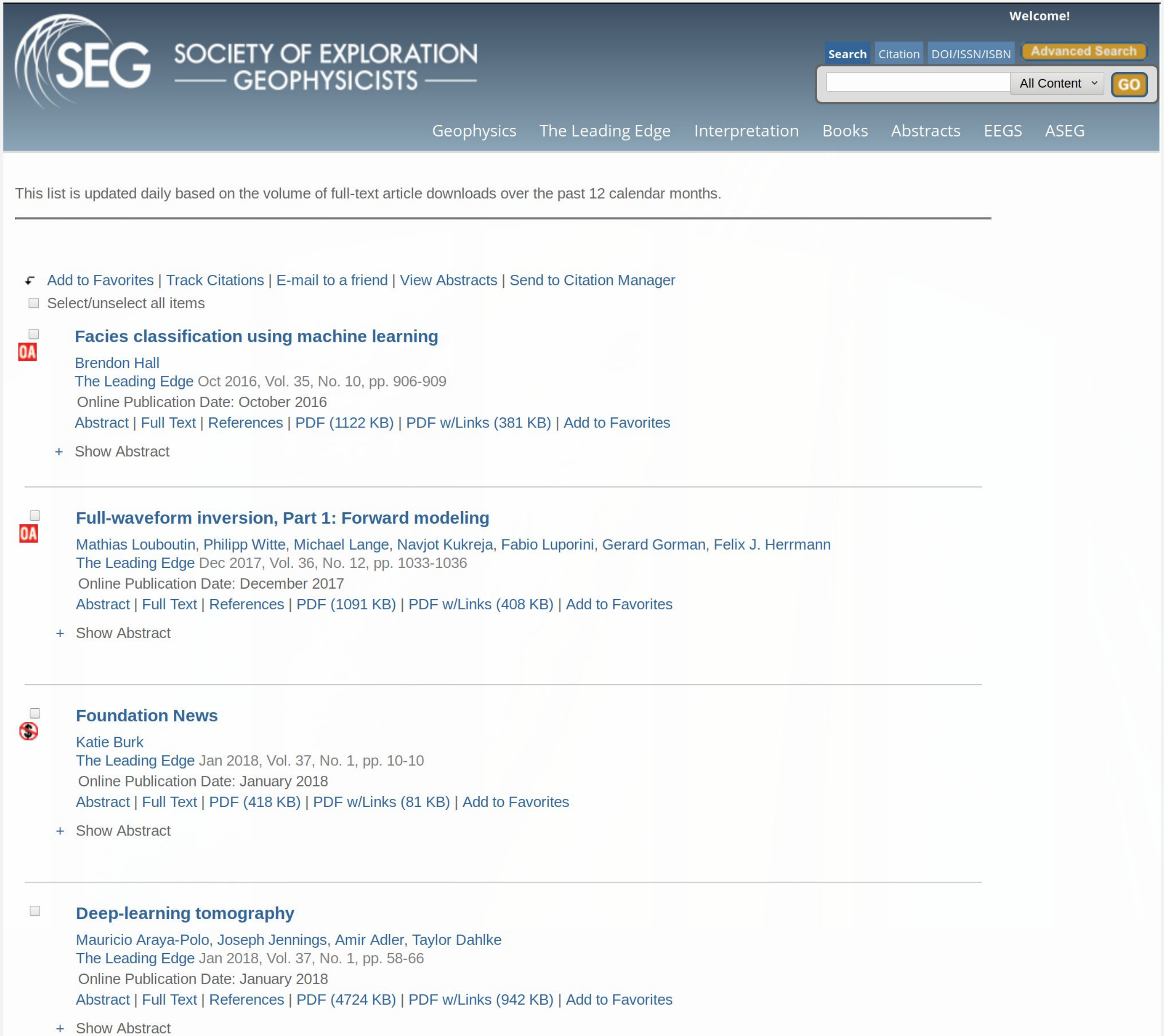
Supervised learning

Model selection

Gridding geophysical data

Equivalent layer

The Leading Edge top downloads (2018)



Welcome!

SEG SOCIETY OF EXPLORATION GEOPHYSICISTS

Search | Citation | DOI/ISSN/ISBN | Advanced Search

All Content | GO

Geophysics | The Leading Edge | Interpretation | Books | Abstracts | EEGS | ASEG

This list is updated daily based on the volume of full-text article downloads over the past 12 calendar months.

Add to Favorites | Track Citations | E-mail to a friend | View Abstracts | Send to Citation Manager

Select/unselect all items

Facies classification using machine learning

Brendon Hall

The Leading Edge Oct 2016, Vol. 35, No. 10, pp. 906-909

Online Publication Date: October 2016

Abstract | Full Text | References | PDF (1122 KB) | PDF w/Links (381 KB) | Add to Favorites

+ Show Abstract

Full-waveform inversion, Part 1: Forward modeling

Mathias Louboutin, Philipp Witte, Michael Lange, Navjot Kukreja, Fabio Luporini, Gerard Gorman, Felix J. Herrmann

The Leading Edge Dec 2017, Vol. 36, No. 12, pp. 1033-1036

Online Publication Date: December 2017

Abstract | Full Text | References | PDF (1091 KB) | PDF w/Links (408 KB) | Add to Favorites

+ Show Abstract

Foundation News

Katie Burk

The Leading Edge Jan 2018, Vol. 37, No. 1, pp. 10-10

Online Publication Date: January 2018

Abstract | Full Text | PDF (418 KB) | PDF w/Links (81 KB) | Add to Favorites

+ Show Abstract

Deep-learning tomography

Mauricio Araya-Polo, Joseph Jennings, Amir Adler, Taylor Dahlke

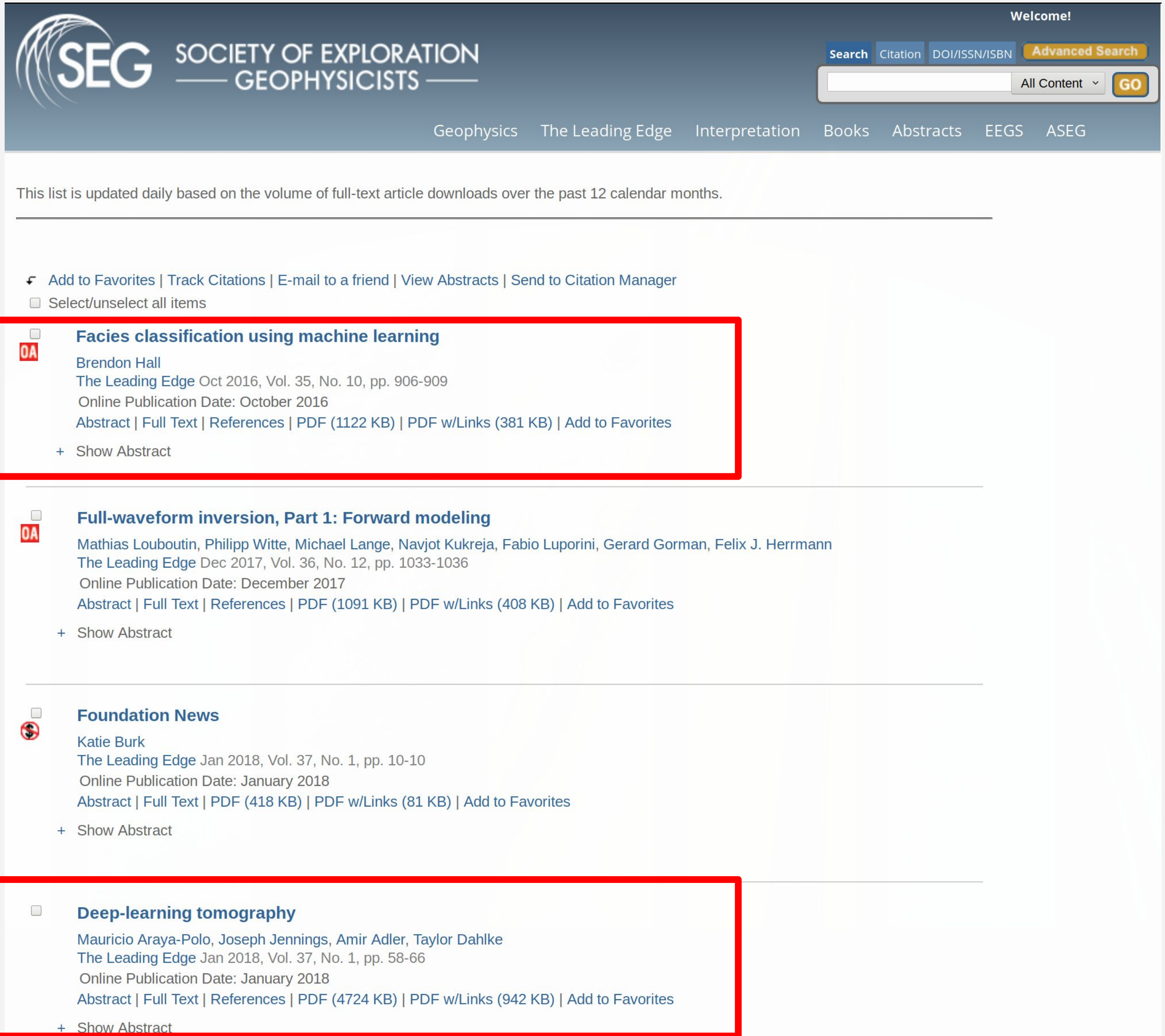
The Leading Edge Jan 2018, Vol. 37, No. 1, pp. 58-66

Online Publication Date: January 2018

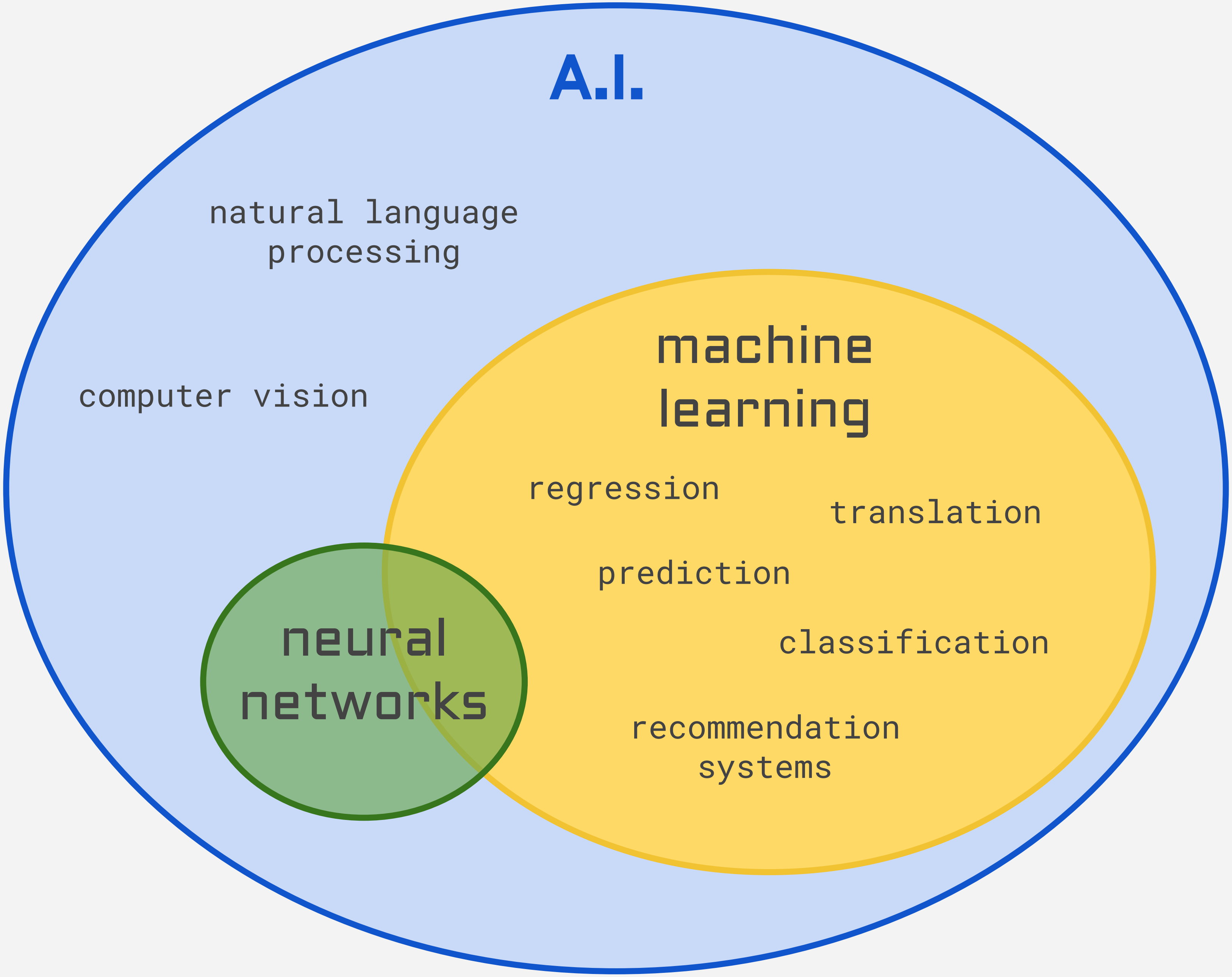
Abstract | Full Text | References | PDF (4724 KB) | PDF w/Links (942 KB) | Add to Favorites

+ Show Abstract

The Leading Edge top downloads (2018)



A.I. and
Machine Learning


A.I.
natural language processing
computer vision
machine learning
regression
translation
prediction
classification
recommendation systems
neural networks

Machine Learning

Practical problems

Learning from data and making predictions

Overlap with statistics and optimization

Computational approach

Summary (over-simplified)

> *Fit a mathematical model to data and use it to make predictions.*

ML Glossary

model

mathematical formula used to approximate the data

parameter

variable in the model that controls its behaviour

labels/classes

quantity/type that we want to predict

features

measurements used as predictors of labels/classes

training

using features and known labels/classes to fit a model

* I’m not an ML expert. Don’t quote me on this.

Different flavors

Supervised Learning

Fit model on data to “train” it for predictions. Apply to new data.

Ex: regression, spam detection

Unsupervised Learning

Extract information and structure from the data without “training”.

Ex: clustering, principal component analysis

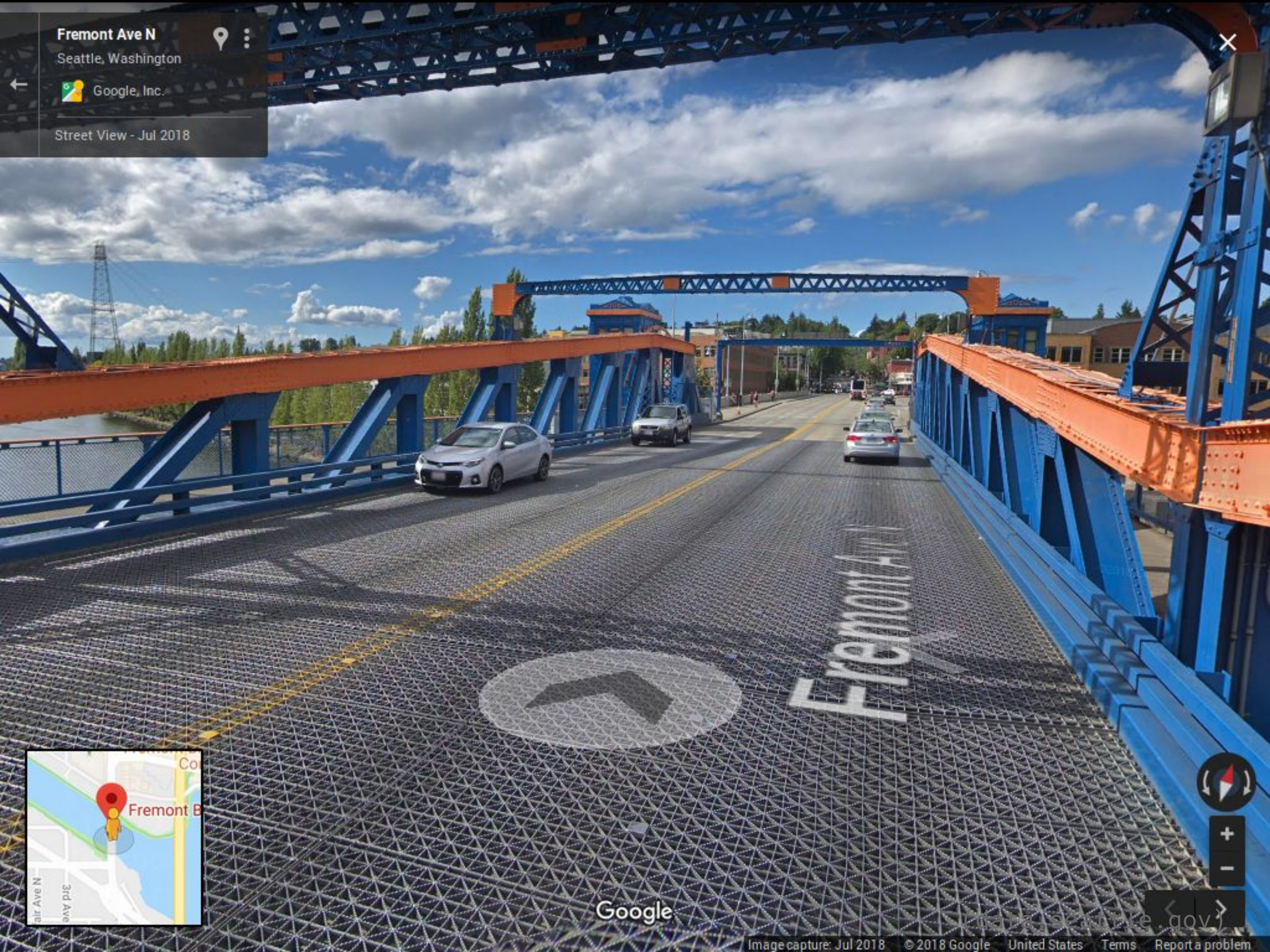
Unsupervised Learning (by example)

Based on “Learning Seattle's Work Habits from Bicycle Counts” by Jake VanderPlas (http://jakevdp.github.io).

Data

Hourly bicycle trips across Seattle's Fremont Bridge:

(data.seattle.gov)



Fremont Ave N
Seattle, Washington
Google, Inc.
Street View - Jul 2018
3rd Ave
Google
Image capture: Jul 2018
© 2018 Google
United States
Terms
Report a problem

Data

Hourly bicycle trips across Seattle's Fremont Bridge:

For each day:

Hourly count (24)

X

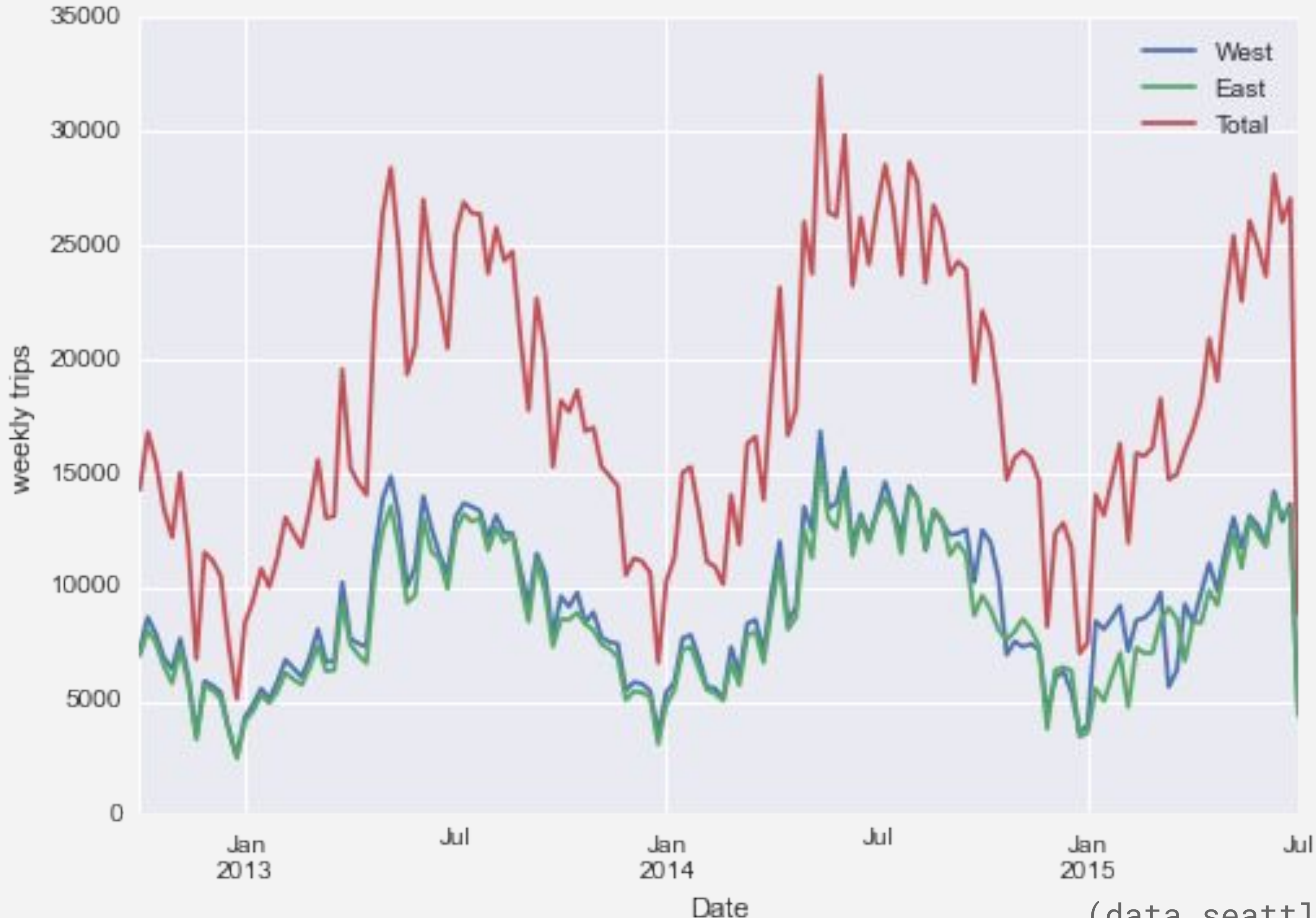
East and West sidewalk sensors (2)

= 48 total observations

How to visualize a 48-dimensional dataset?

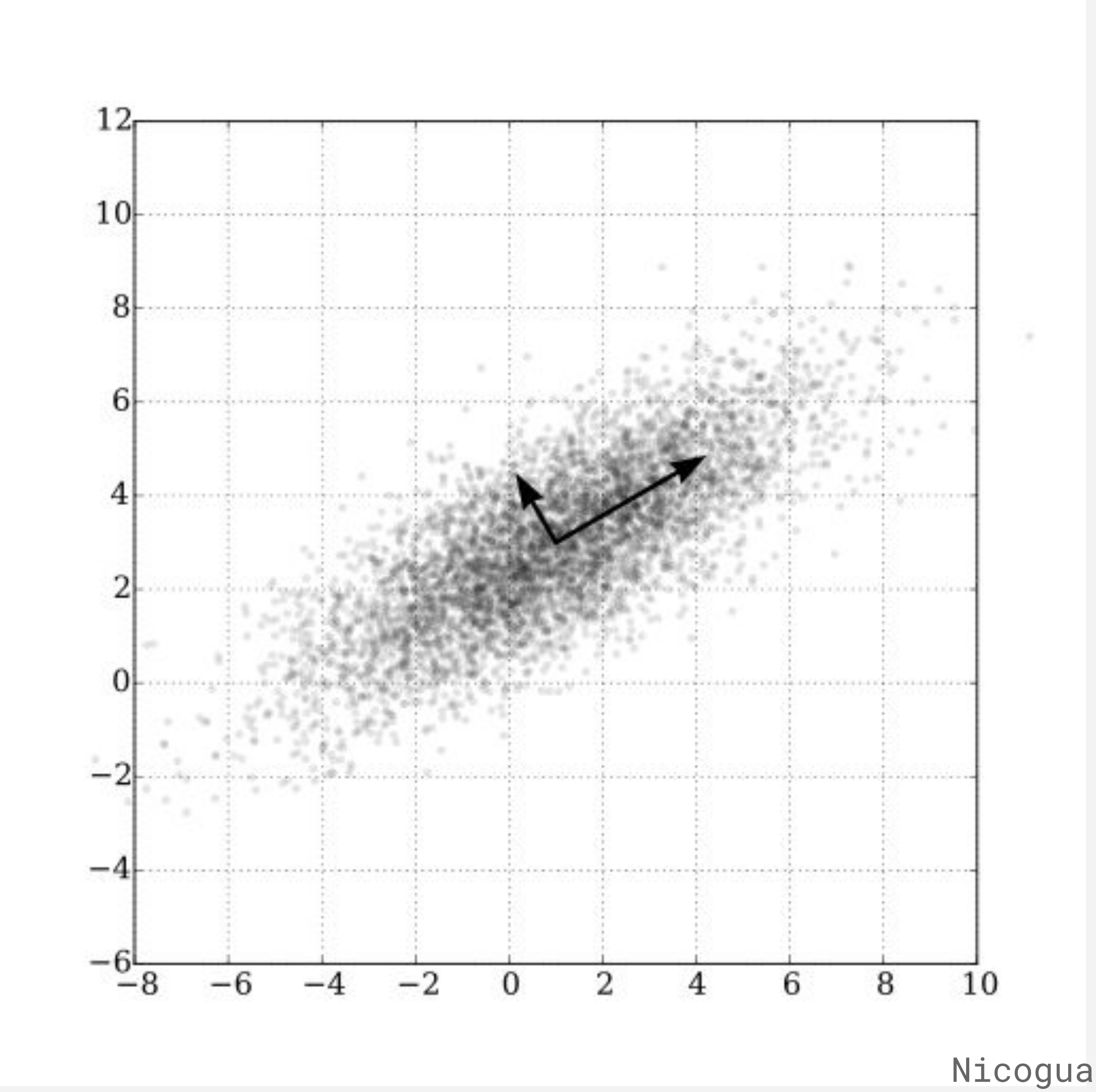
Data

Hourly bicycle trips across Seattle's Fremont Bridge:

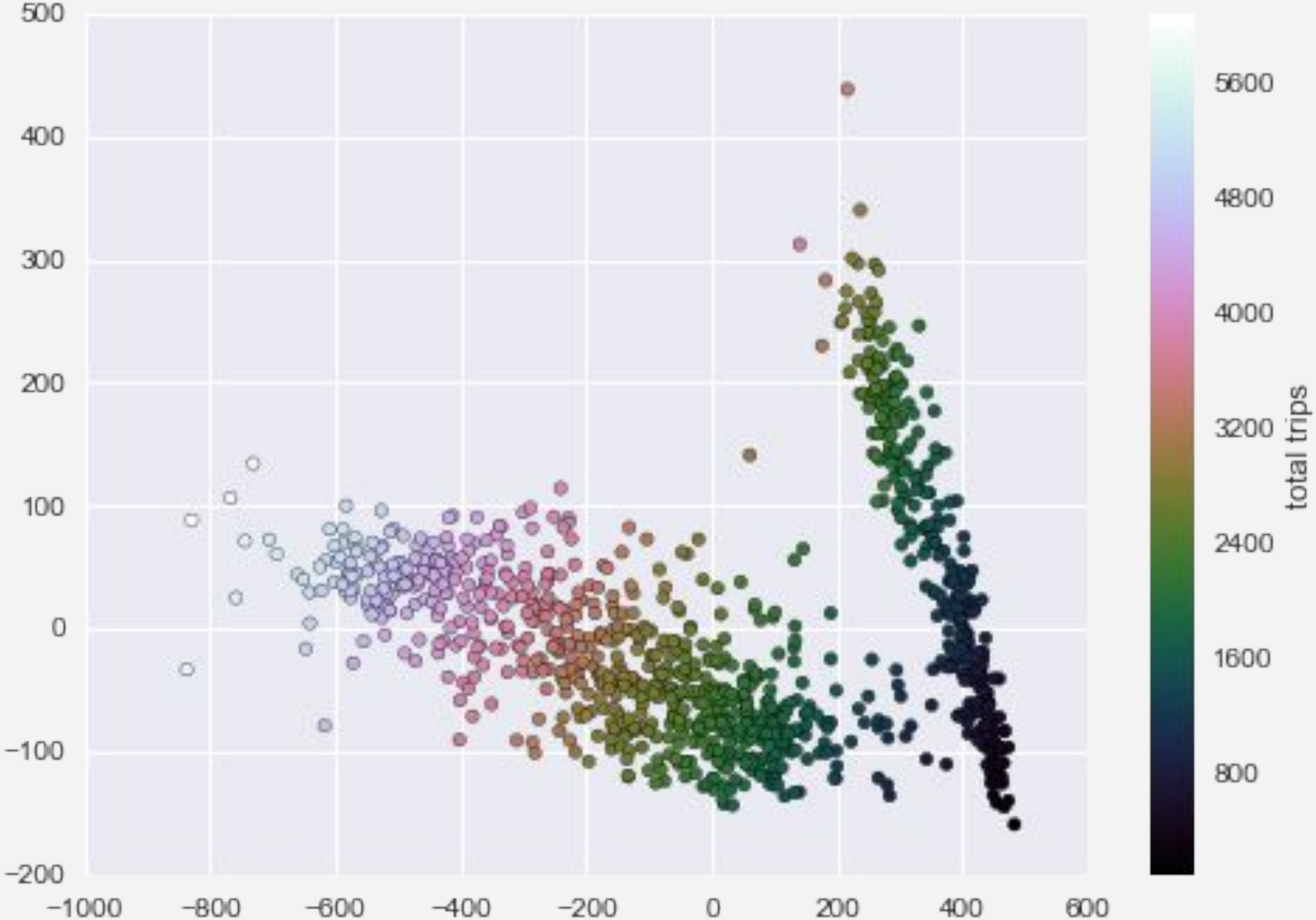


(data.seattle.gov)

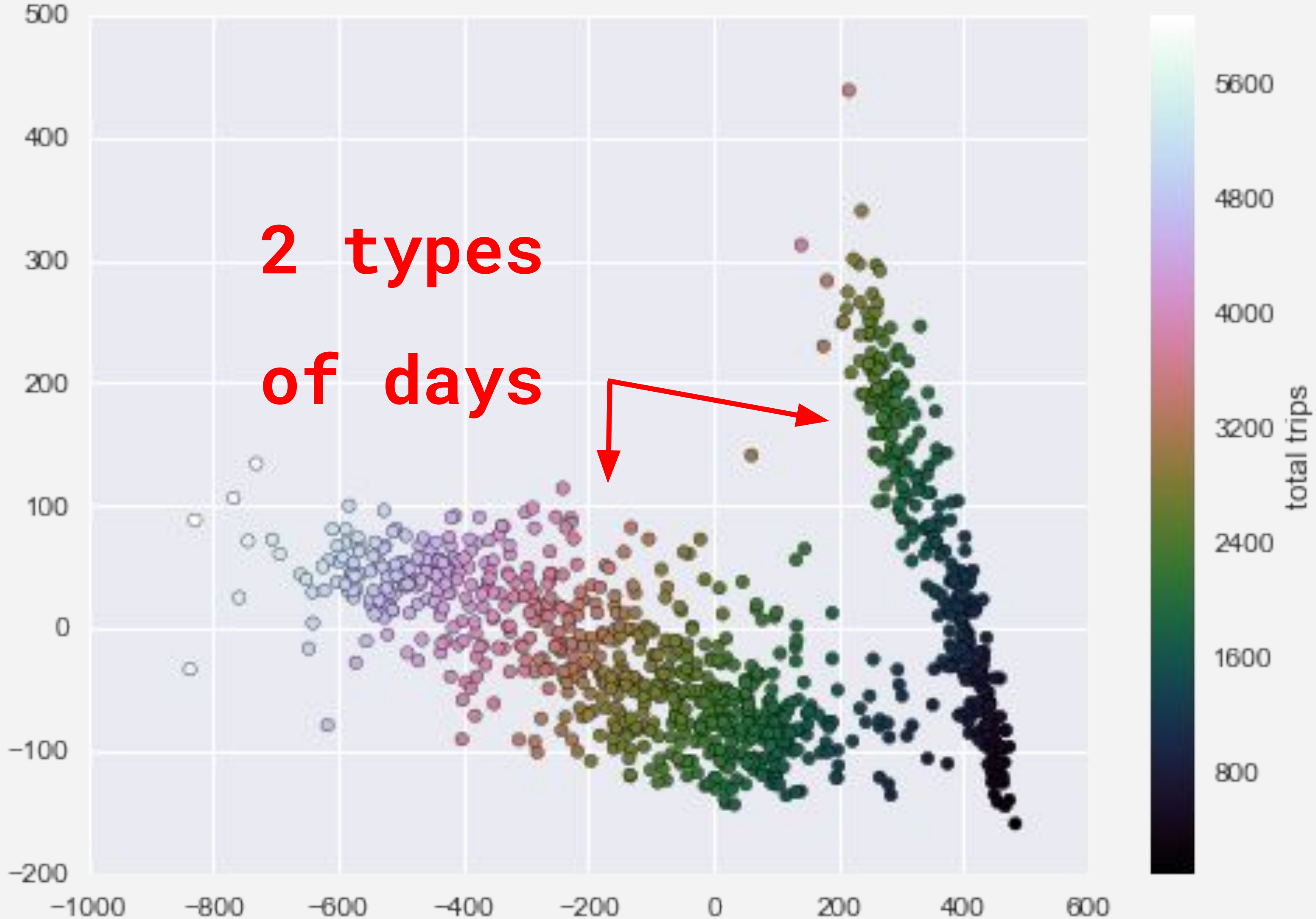
Principal Component Analysis



Principal Component Analysis

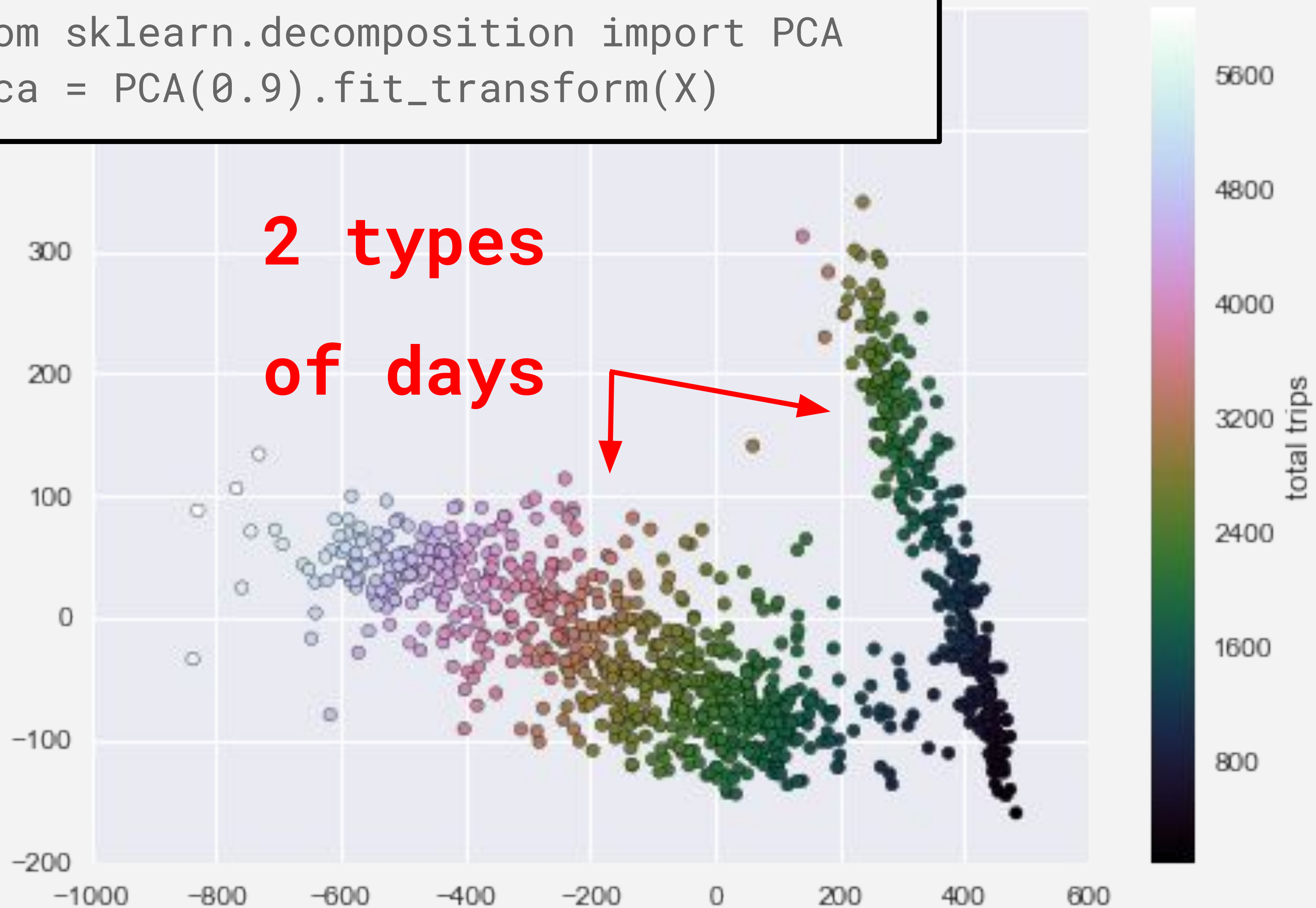


Principal Component Analysis

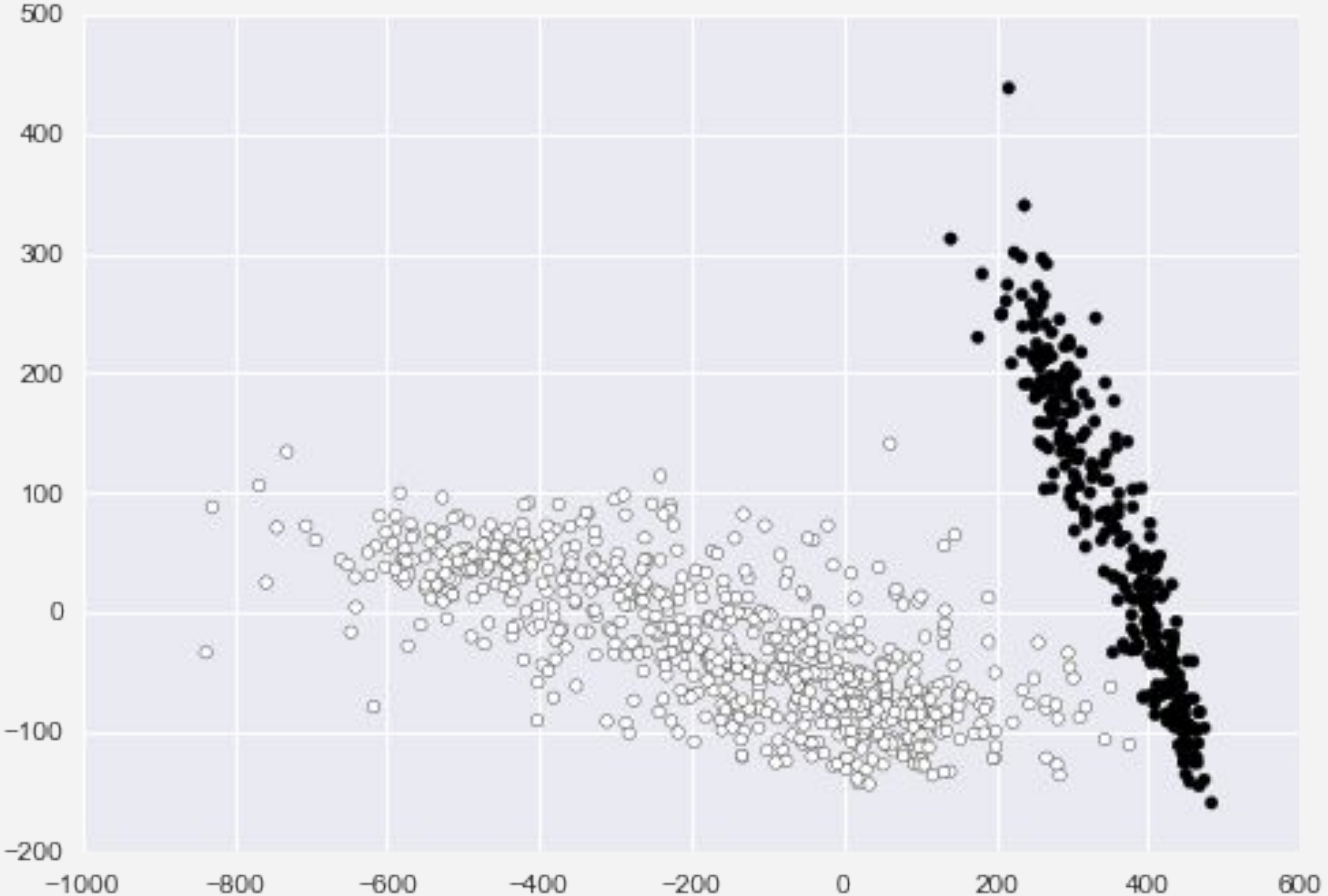


Principal Component Analysis

```
from sklearn.decomposition import PCA
Xpca = PCA(0.9).fit_transform(X)
```

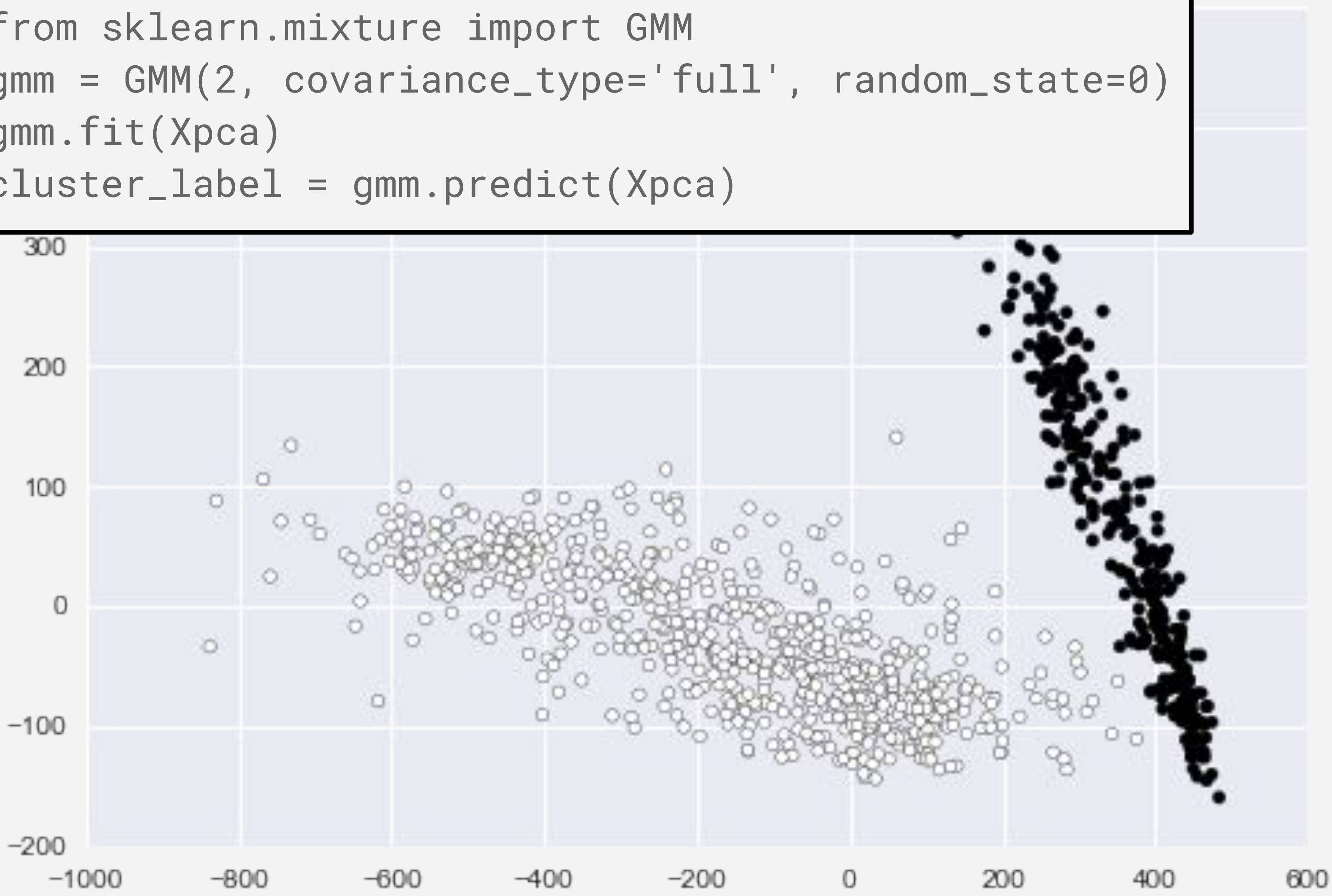


Clustering



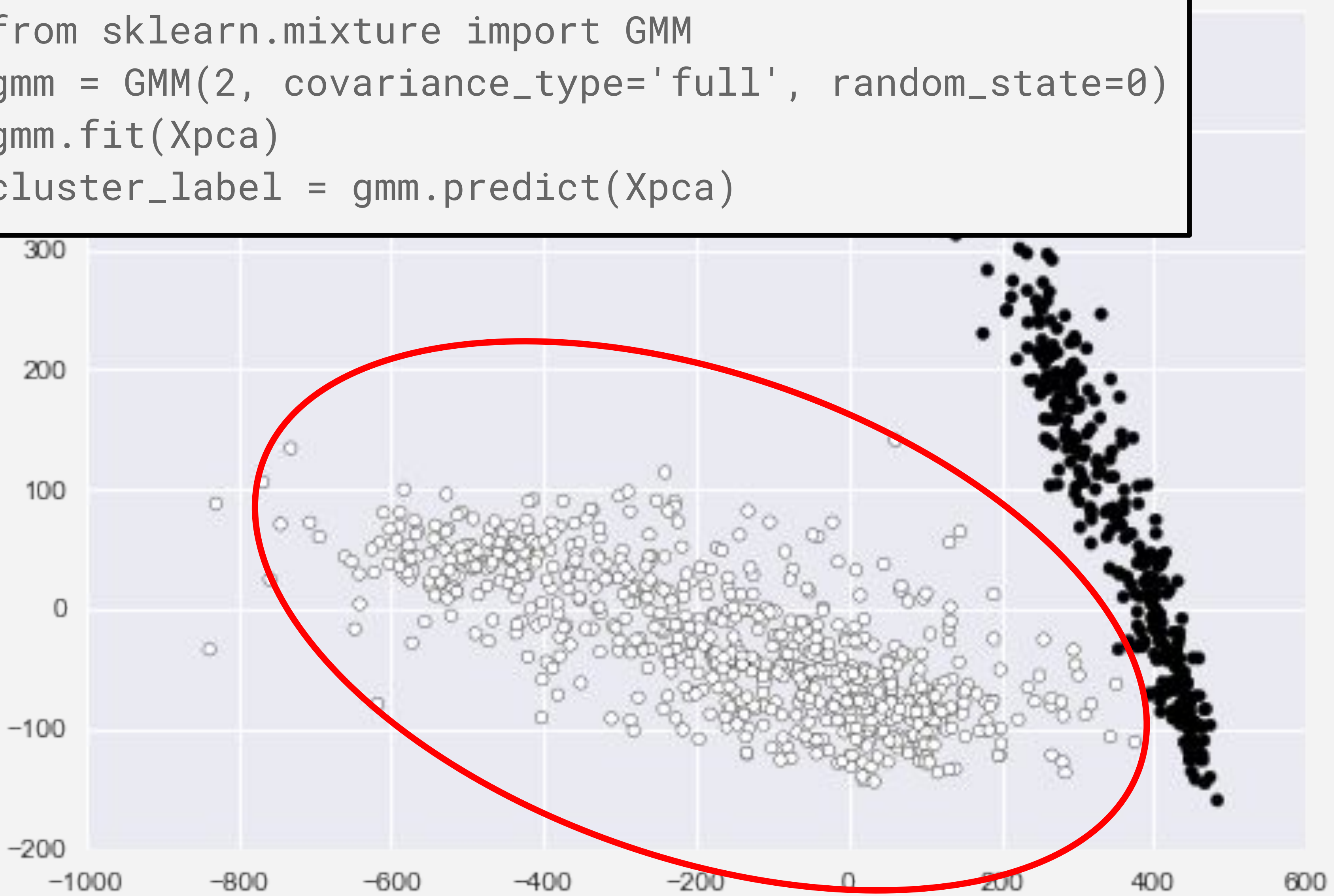
Clustering

```
from sklearn.mixture import GMM
gmm = GMM(2, covariance_type='full', random_state=0)
gmm.fit(Xpca)
cluster_label = gmm.predict(Xpca)
```

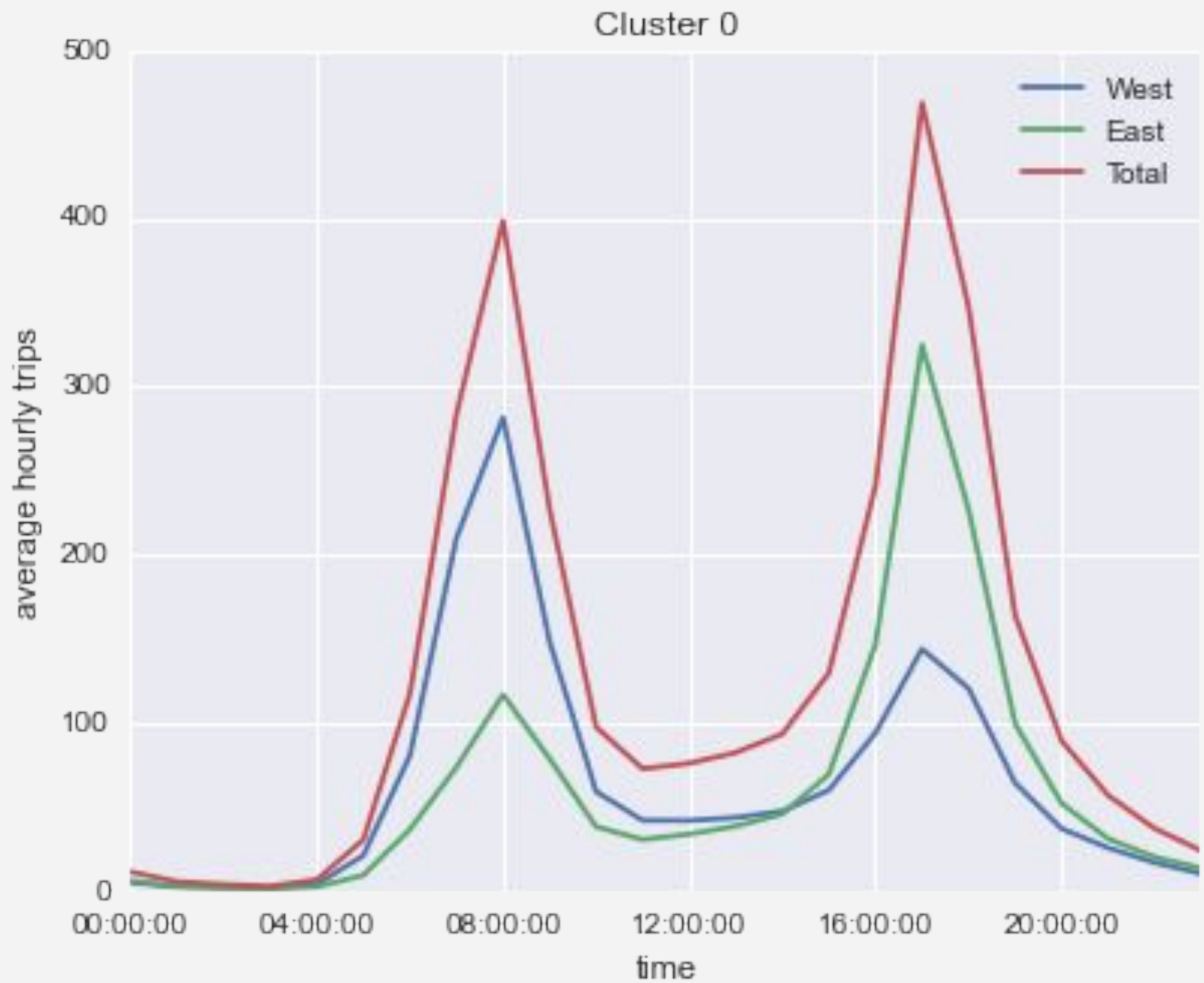


Clustering

```
from sklearn.mixture import GMM
gmm = GMM(2, covariance_type='full', random_state=0)
gmm.fit(Xpca)
cluster_label = gmm.predict(Xpca)
```

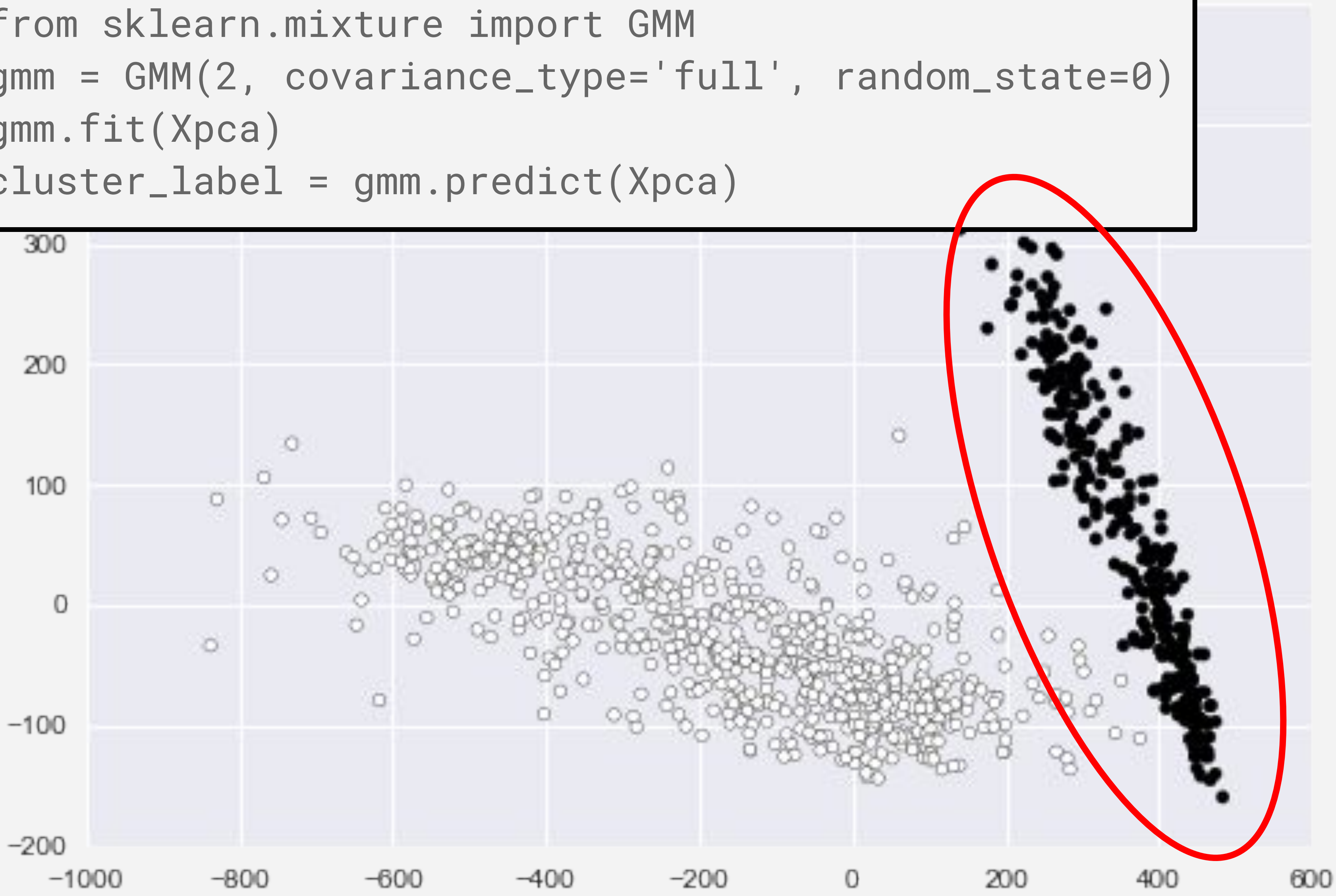


Clustering (back to original data)

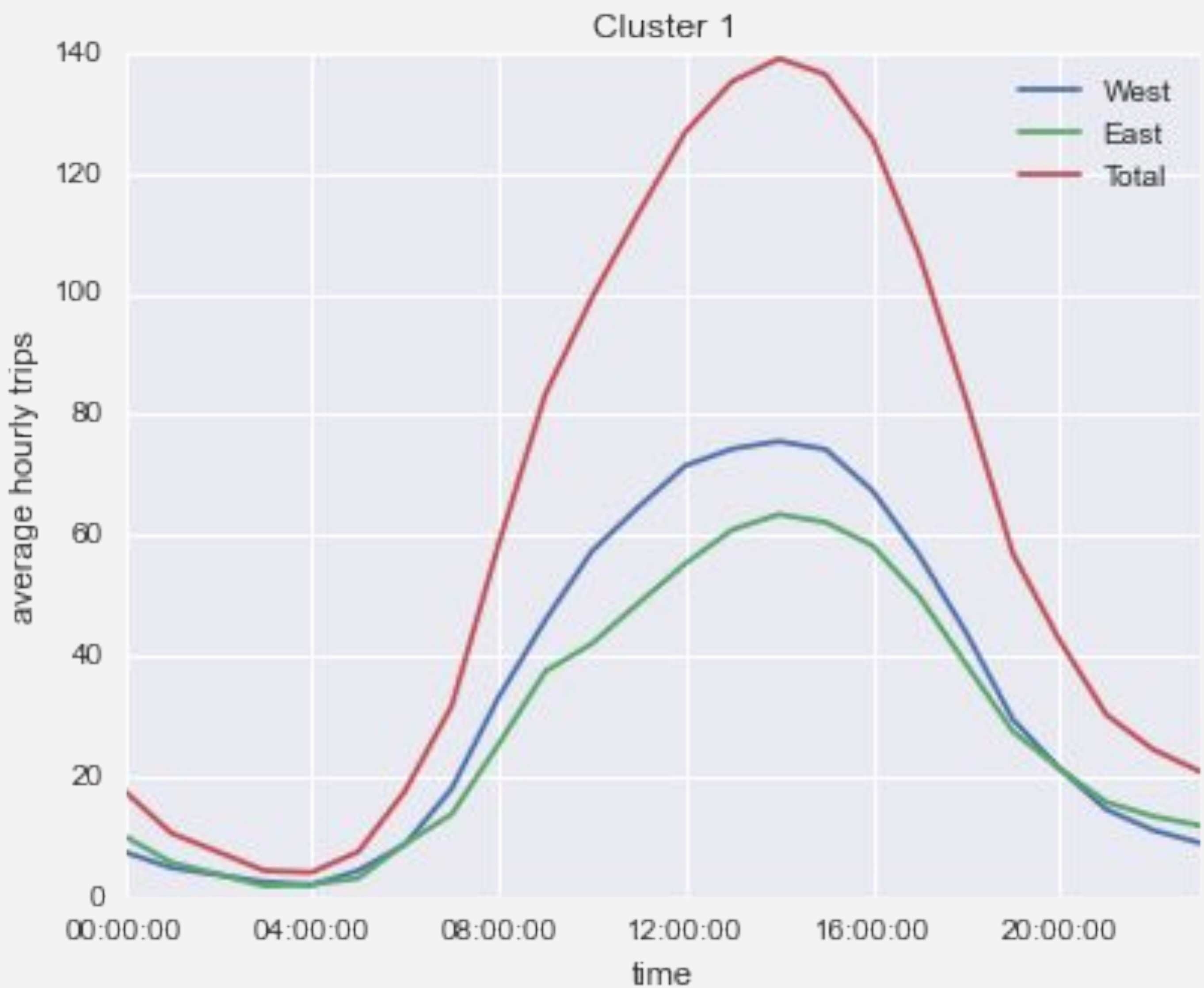


Clustering

```
from sklearn.mixture import GMM
gmm = GMM(2, covariance_type='full', random_state=0)
gmm.fit(Xpca)
cluster_label = gmm.predict(Xpca)
```



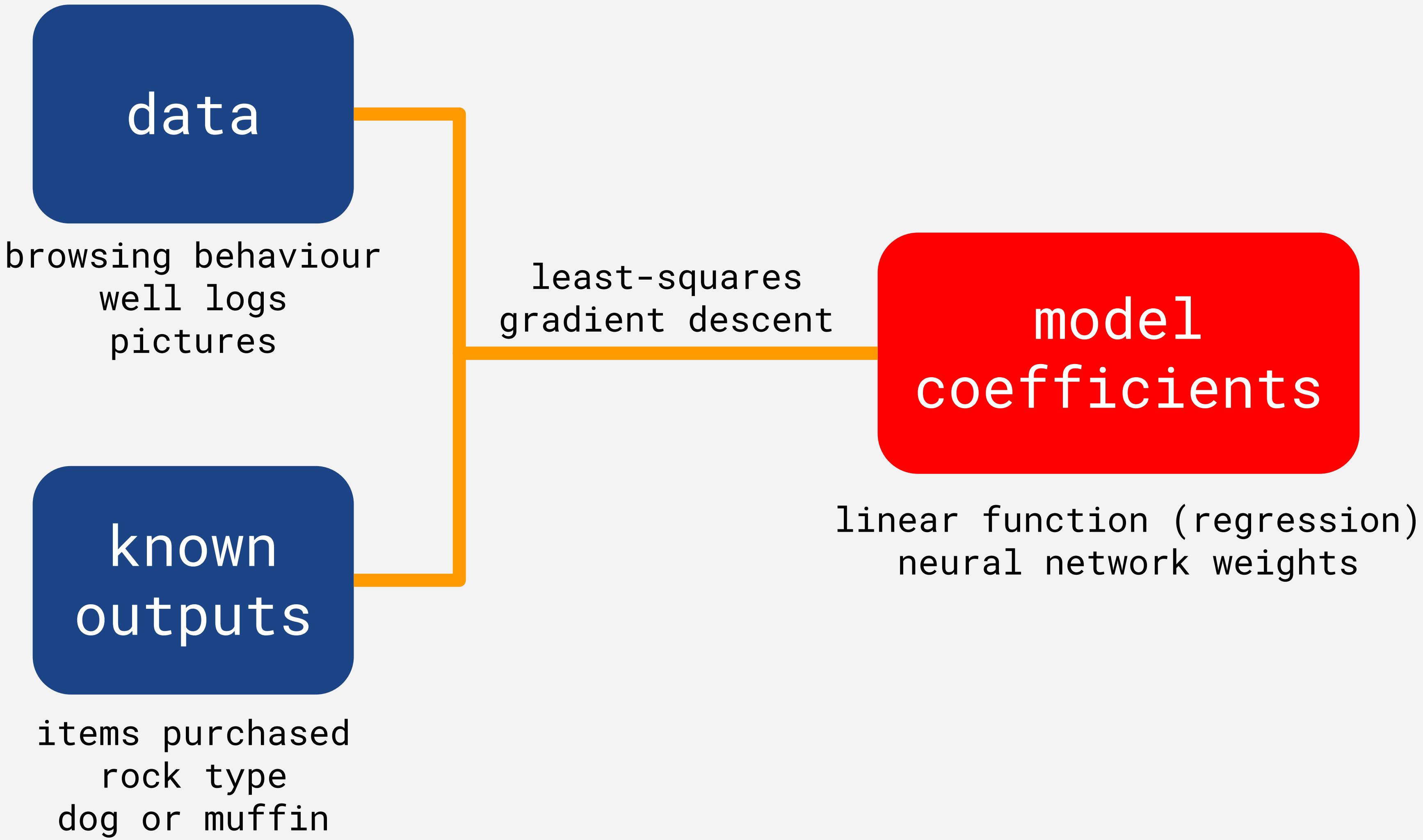
Clustering (back to original data)



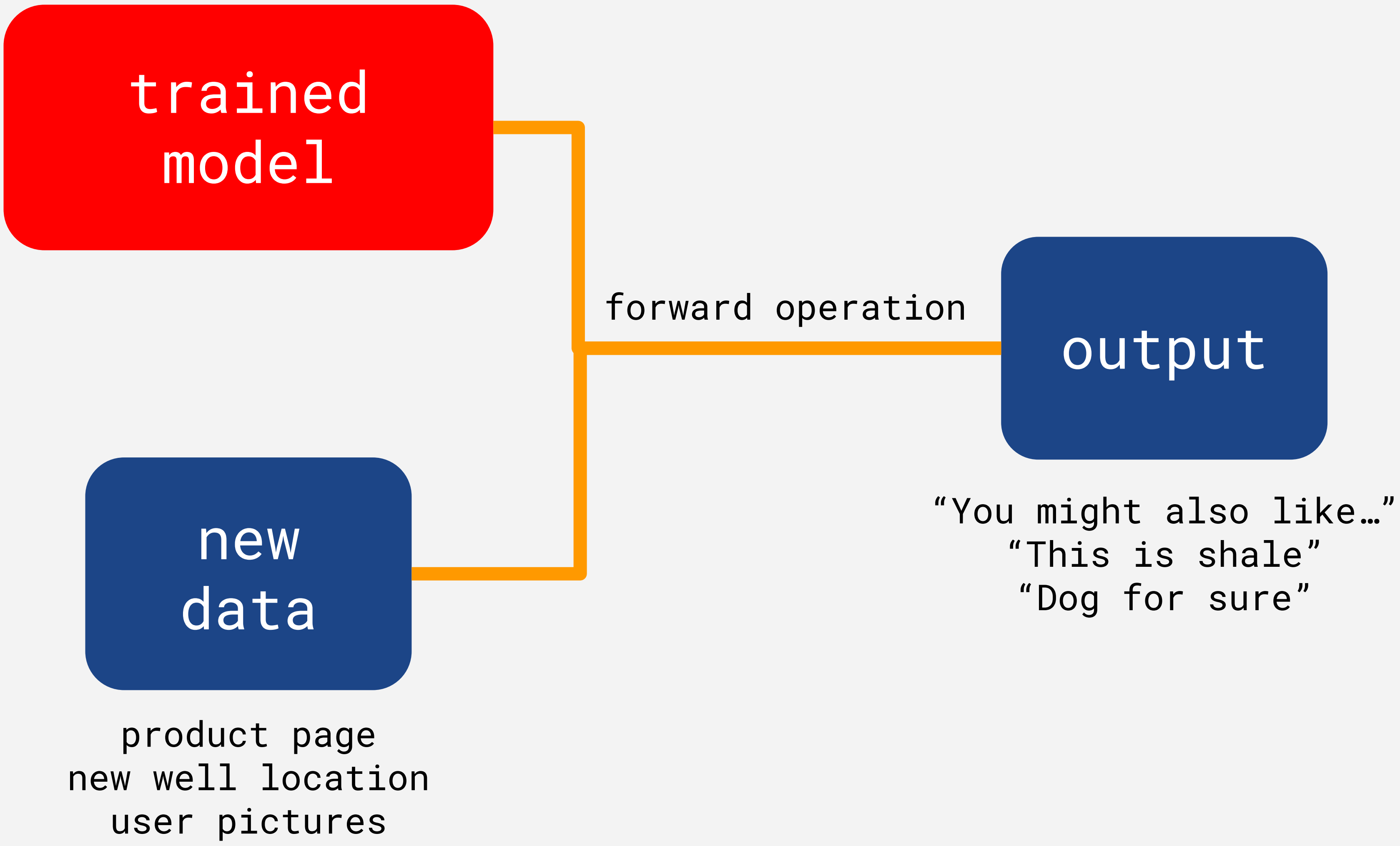
Information comes from the data itself (no models)

Supervised Learning

Train a model on data



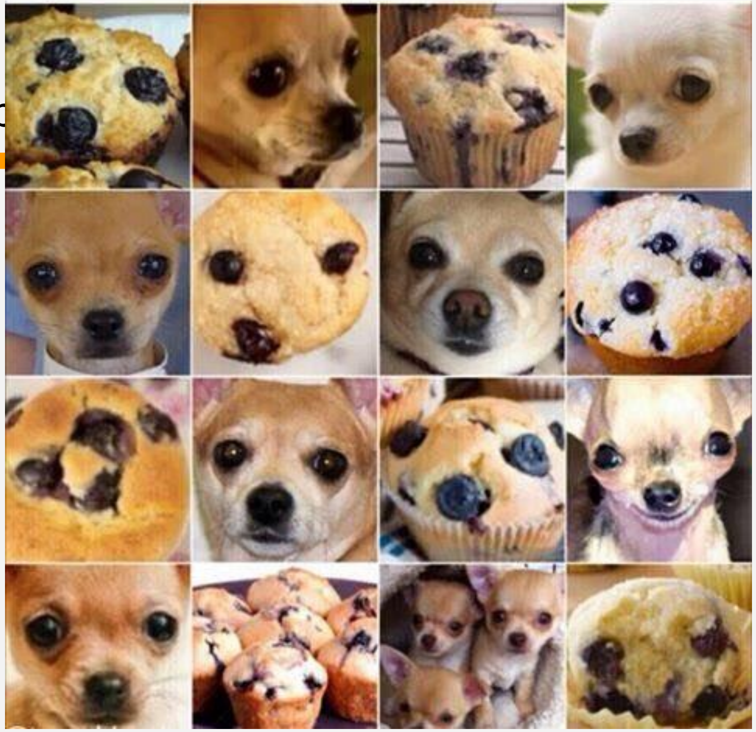
Make predictions with the model



Make predictions with the model

trained model

forward op



new data

product page
new well location
user pictures

<u>Example:</u> facies classification from well logs

Based on Hall (2016) tutorial on The Leading Edge

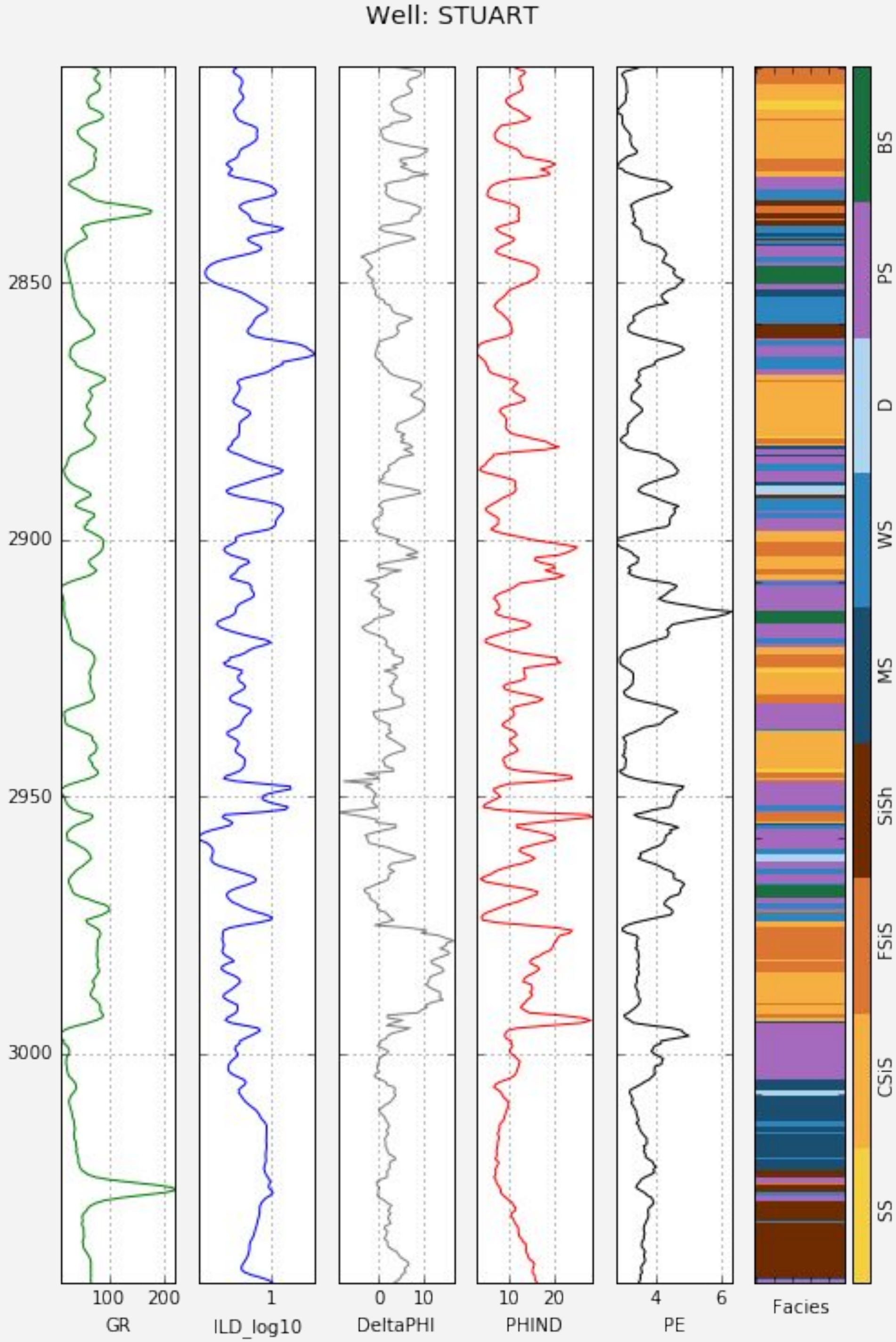
Data

Features (well logs)

- Gamma ray
- Resistivity
- Photoelectric effect
- Neutron-density porosity difference
- Average neutron-density porosity
- Nonmarine/marine indicator
- Relative position

Classes (facies)

- Nonmarine sandstone
- Nonmarine coarse siltstone
- Nonmarine fine siltstone
- Marine siltstone and shale
- Mudstone
- Wackestone
- Dolomite
- Packstone-grainstone
- Phylloid-algal bafflestone

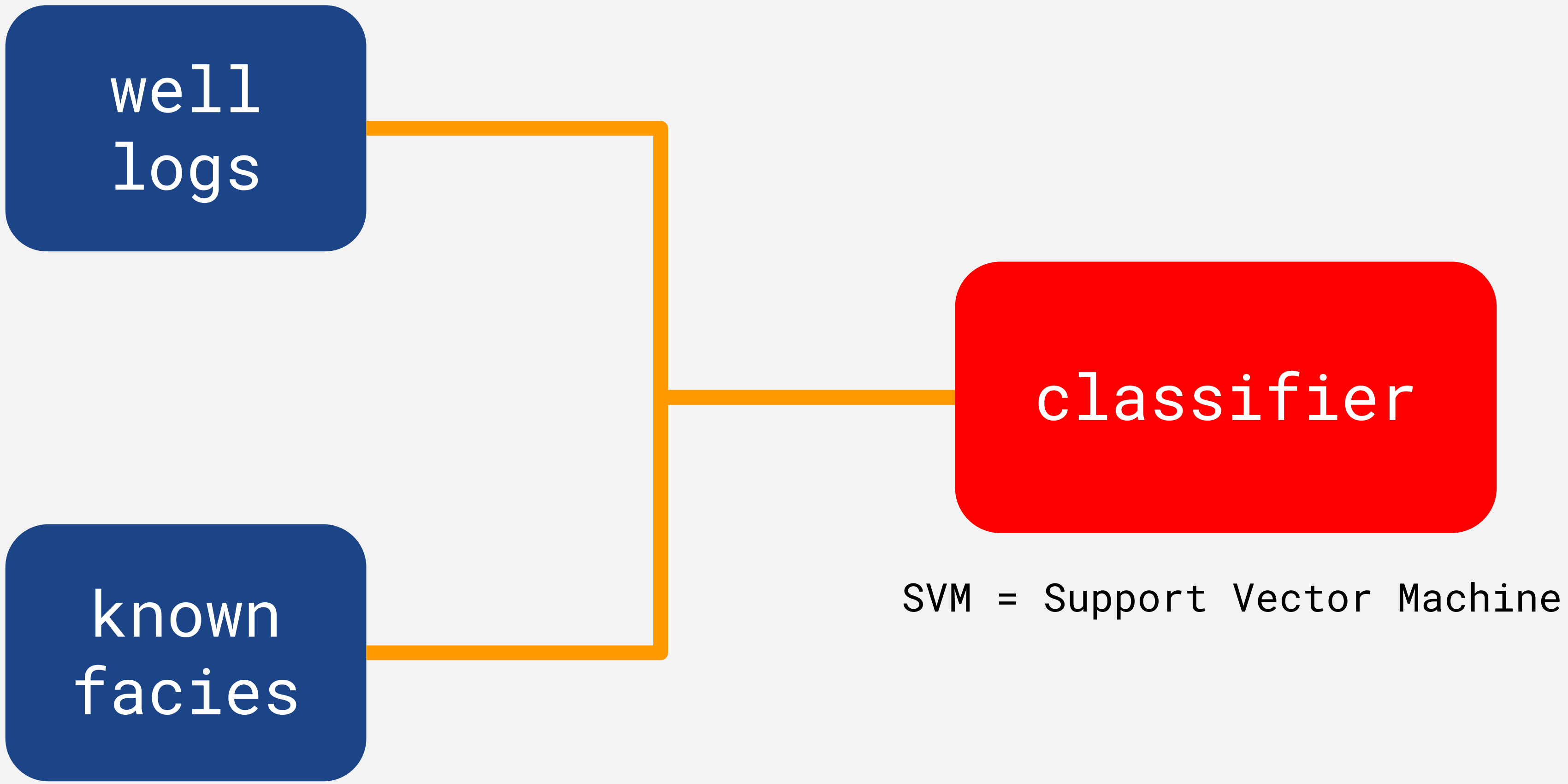

Well: STUART
2850
2900
2950
3000
100
200
GR
1
ILD_log10
0
10
DeltaPHI
10
20
PHIND
4
6
PE
Facies
BS
PS
D
WS
MS
SiSh
FSiS
CSiS
SS

Data

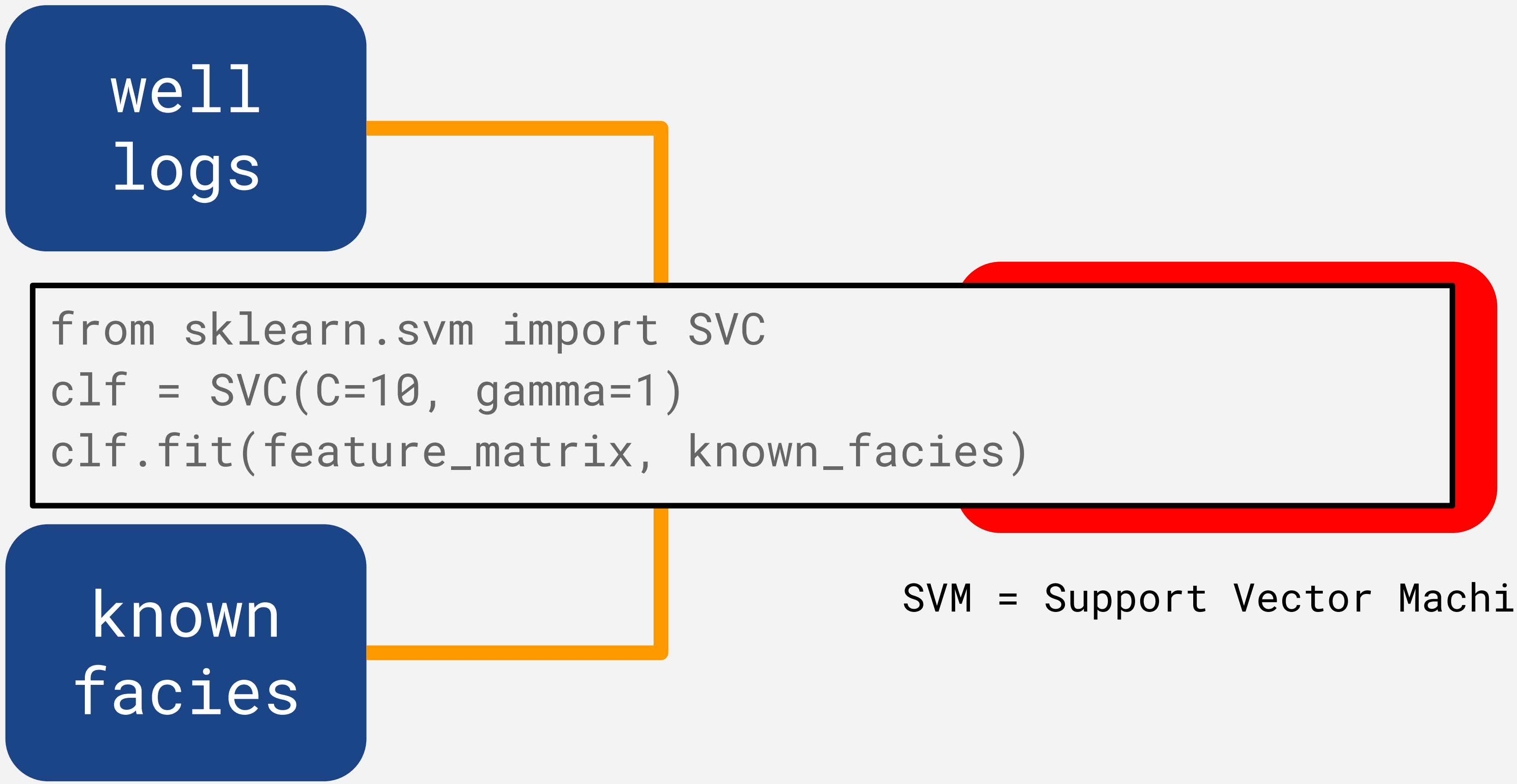
Features (well logs)

	gamma	resistivity	position	…
obs1	…	…	…	
obs2	…	…	…	
…				
obsN	…	…	…	

Training


well logs
known facies
classifier
SVM = Support Vector Machine

Training



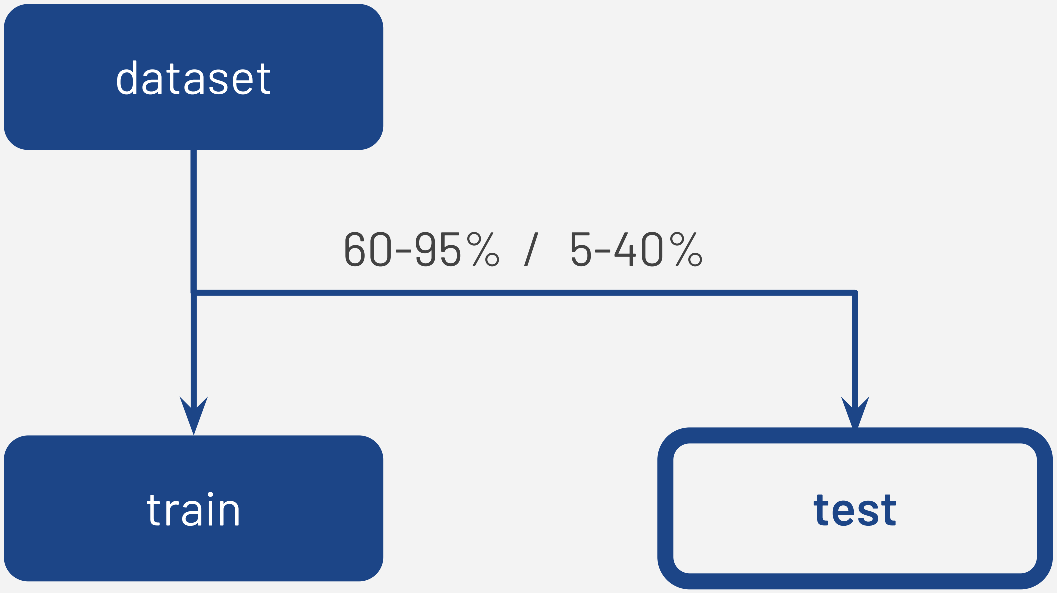

```
from sklearn.svm import SVC
clf = SVC(C=10, gamma=1)
clf.fit(feature_matrix, known_facies)
```

SVM = Support Vector Machine

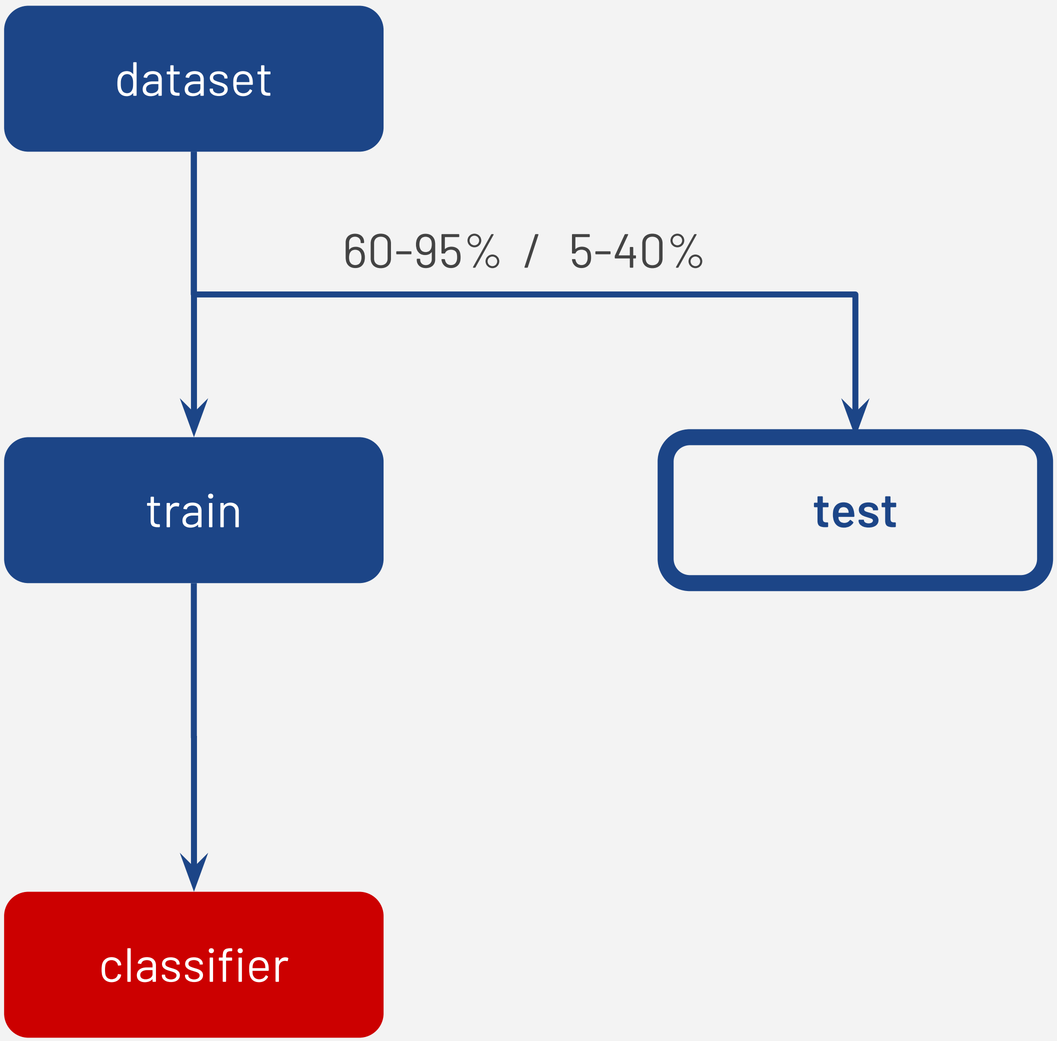
How well does the model perform?

Validation

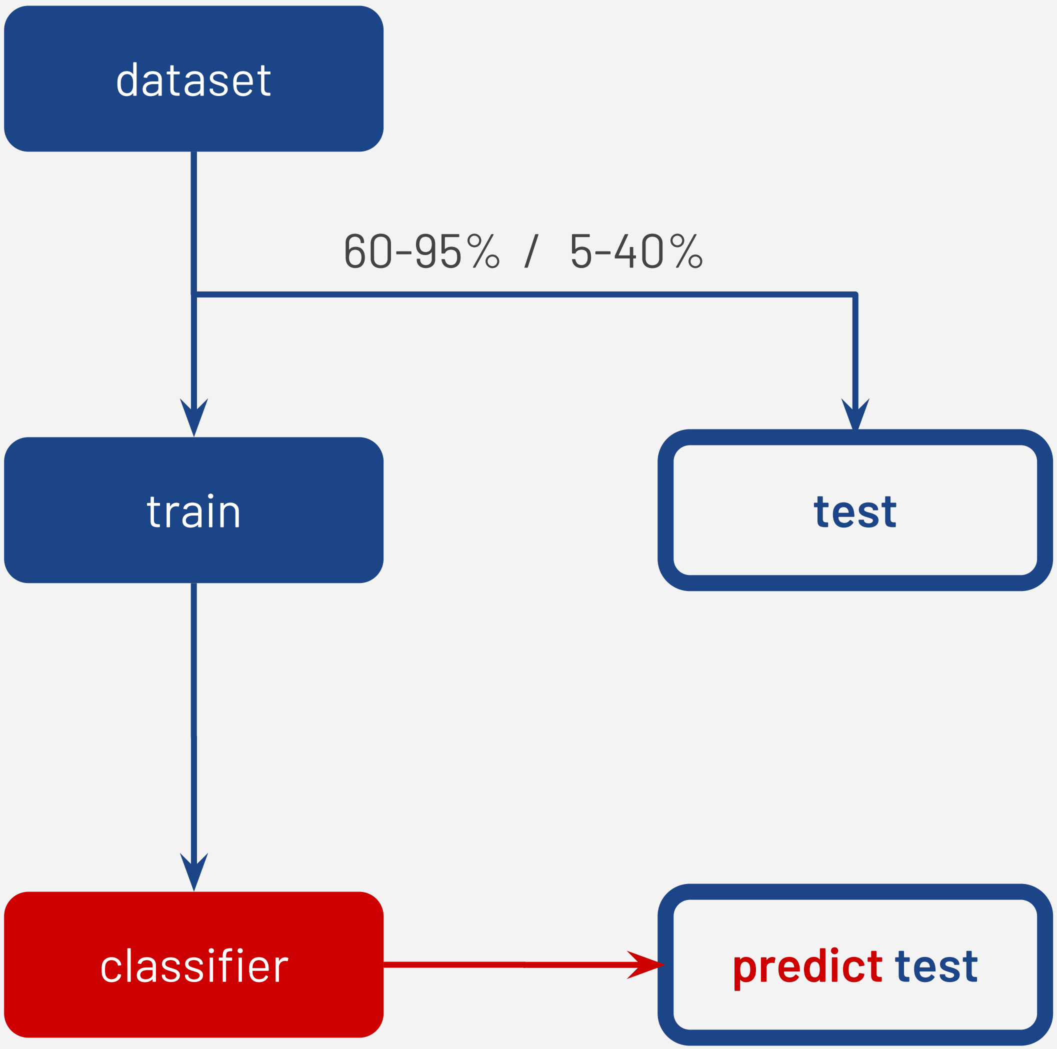
Split the dataset



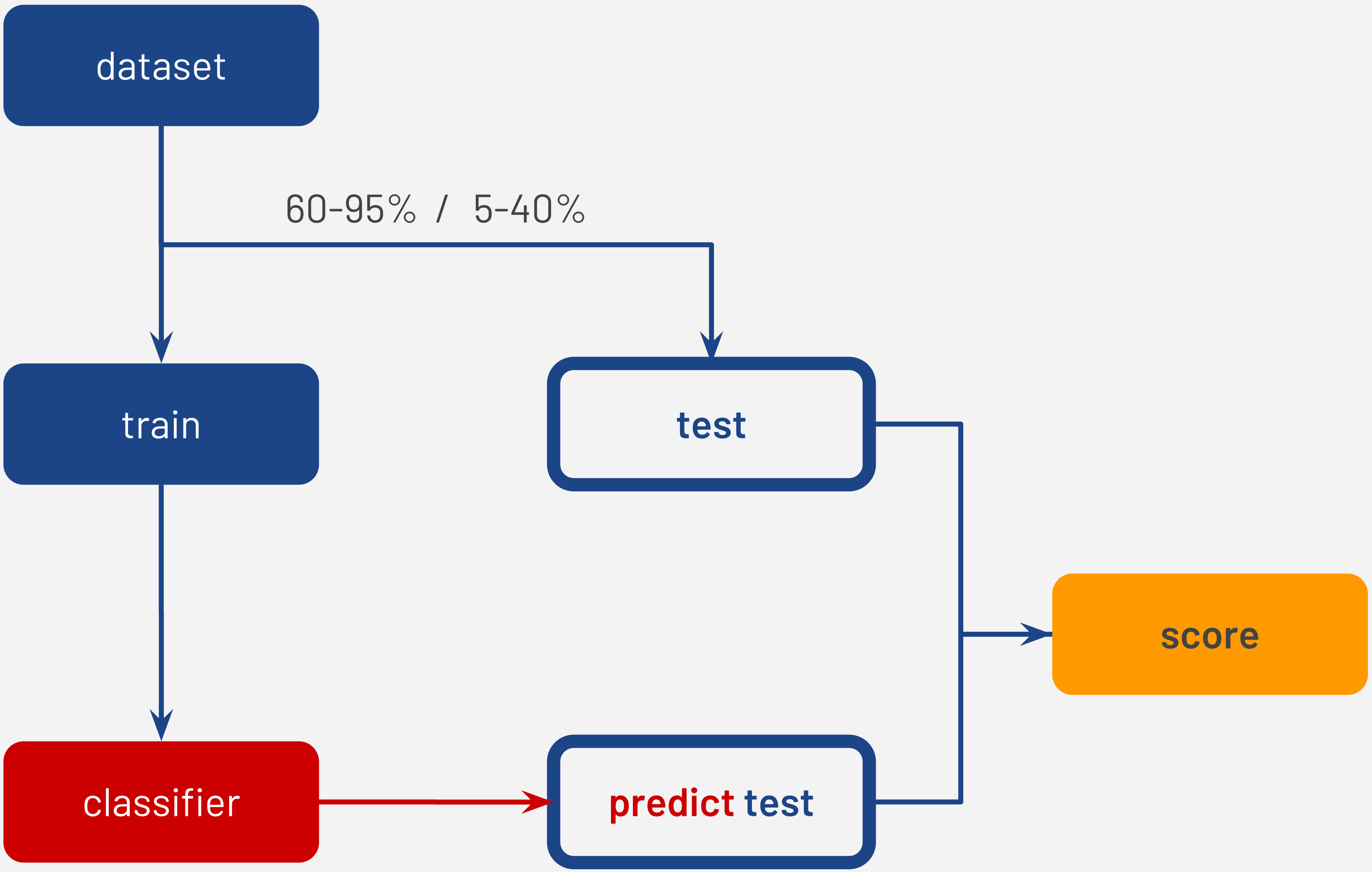
Split the dataset



Split the dataset



Split the dataset



Facies prediction score

95% train - 5% test

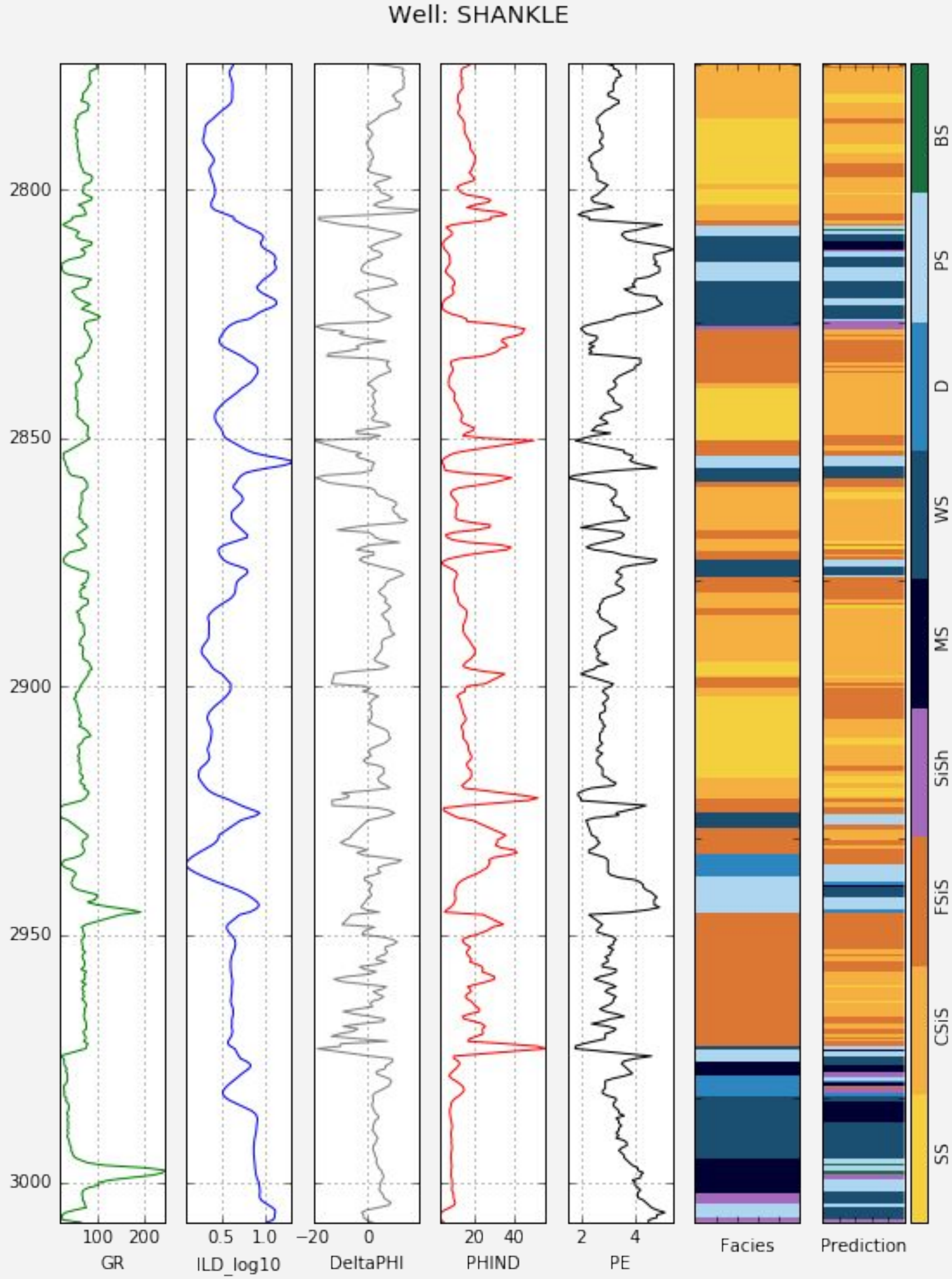
F1 score

Overall measurement of classification accuracy.

1 - perfect prediction

0 - worse possible

0.43



Well: SHANKLE
2800
2850
2900
2950
3000
100 200
GR
0.5 1.0
ILD_log10
−20 0
DeltaPHI
20 40
PHIND
2 4
PE
Facies
Prediction
BS
PS
D
WS
MS
SiSh
FSiS
CSiS
SS

DATA ARE EVERYTHING

DATA ARE EVERYTHING*

Model Selection

Model selection

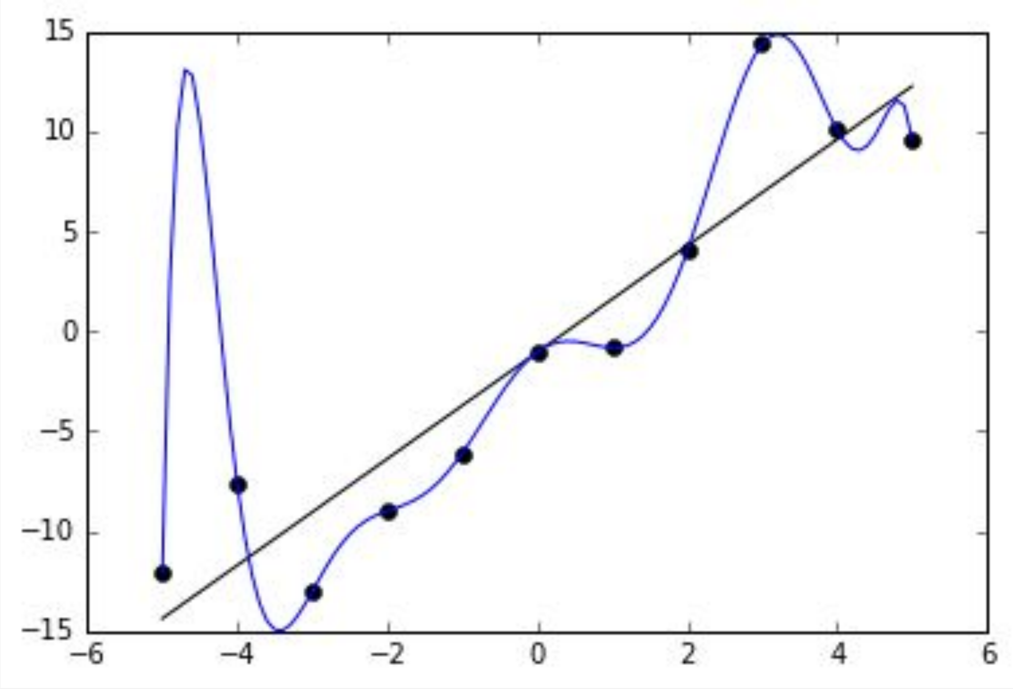
Tune model parameters based on score against test data.

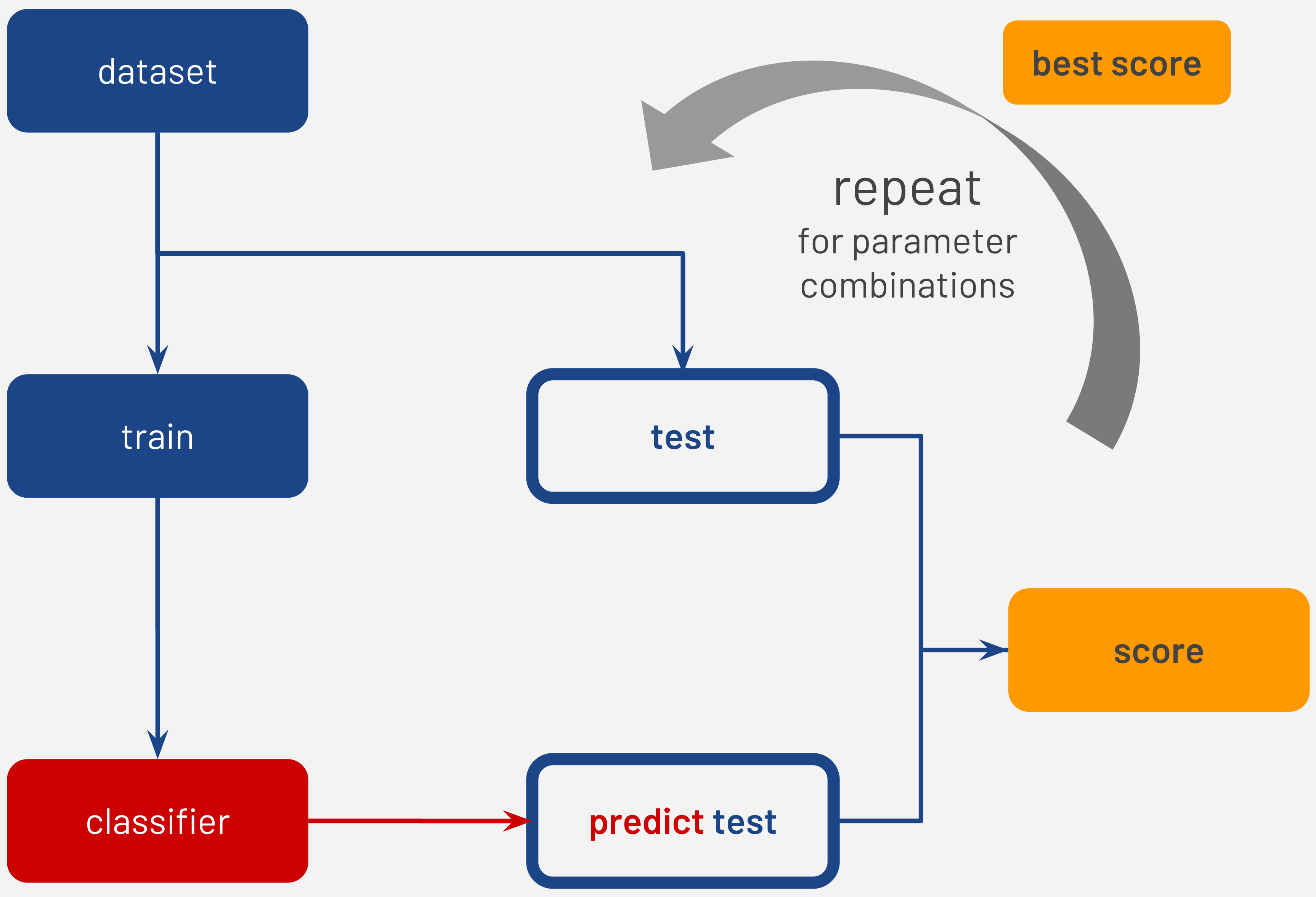
- Automatic
- Prevent overfitting

Model selection

Tune model parameters based on score against test data.

- Automatic
- Prevent overfitting



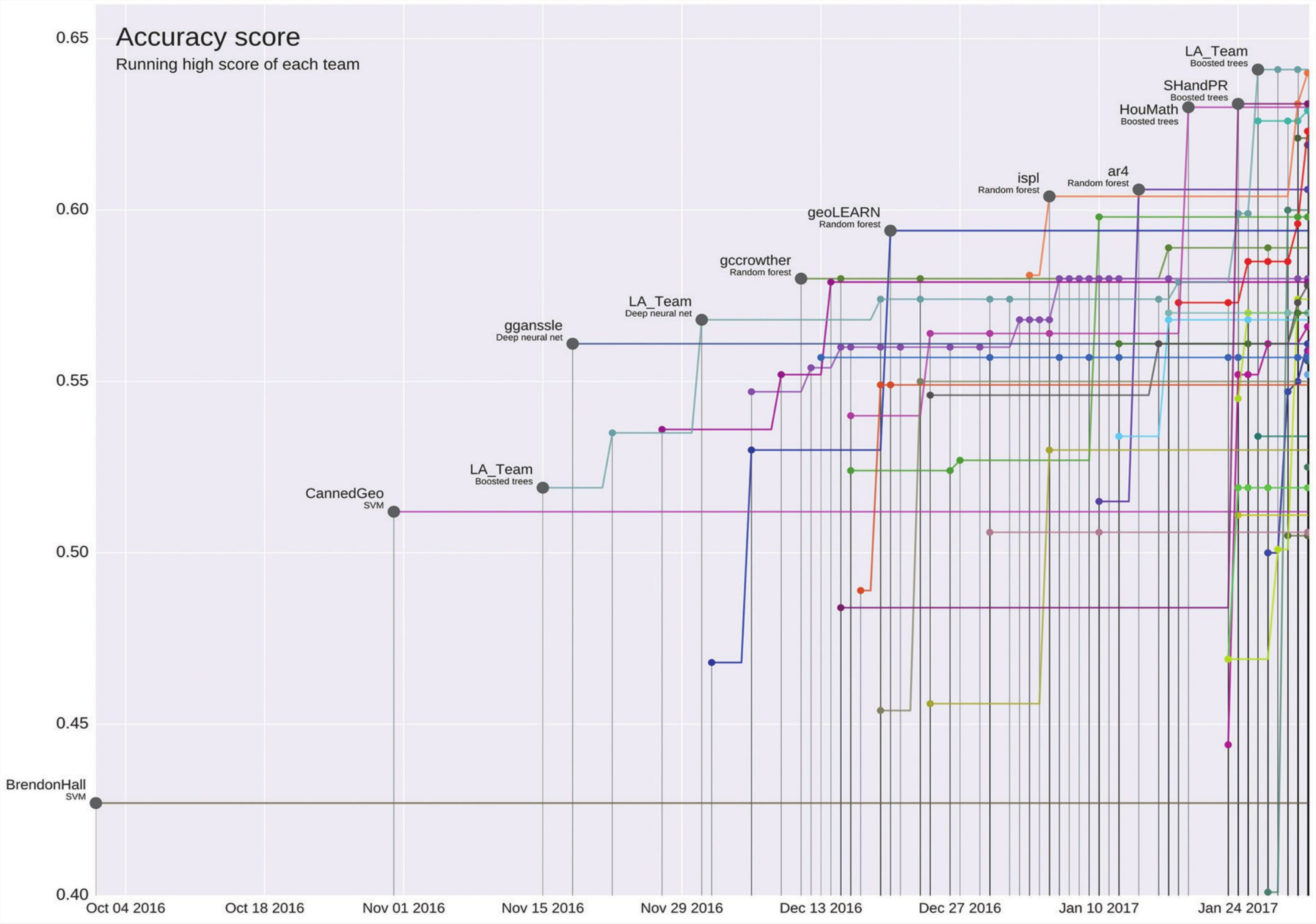

dataset
best score
repeat
for parameter
combinations
train
test
score
classifier
predict test

Model that best predicts data it has never seen

Model that best predicts data it has never seen*

Hand tuning

Hall & Hall (2017) The Leading Edge contest



Gridding

Gridding is prediction

Predict values on points without measurements

Green's functions

- Linear model: data = f(coefficients)
- Estimate coeffs based on observations
- Predict data on grid using coeffs
- AKA radial basis functions

fatiando.org/verde

VERDE

v1.0.1

Search docs

GETTING STARTED

Overview
Installing
Citing Verde
Gallery

USER GUIDE

Sample Data
Trend Estimation
Data Decimation
Geographic Coordinates
Chaining Operations
Model Selection
Using Weights
Vector Data

REFERENCE DOCUMENTATION

API Reference
Changelog
References

GETTING HELP AND CONTRIBUTING

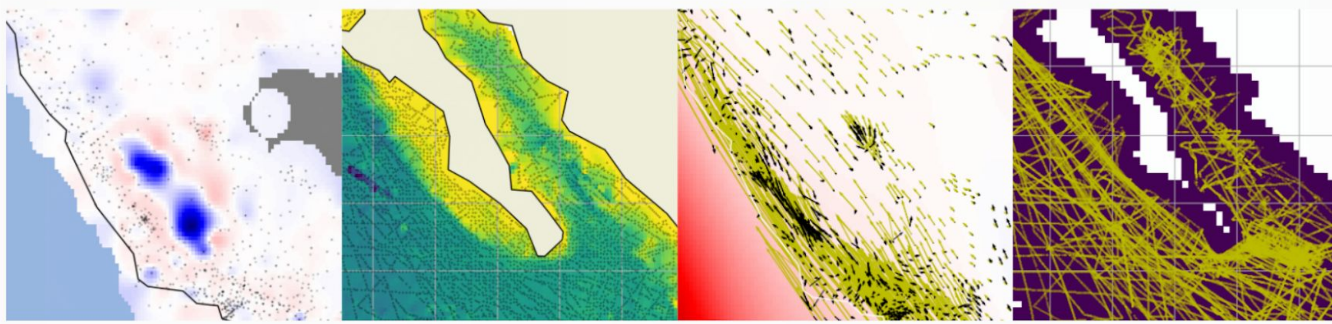
Fatiando a Terra
Contributing

Docs » Home

Improve this page

Processing and gridding spatial data

A part of the Fatiando a Terra project.



About

Verde is a Python library for processing spatial data (bathymetry, geophysics surveys, etc) and interpolating it on regular grids (i.e., *gridding*).

Verde: Processing and gridding spatial data using Green's functions

Article details

- View review »
- Download paper »
- Software repository »
- Software archive »

Submitted: 14 September 2018
Accepted: 11 October 2018

Cite as:

Uieda, (2018). Verde: Processing and gridding spatial data using Green's functions. Journal of Open Source Software, 3(30), 957, https://doi.org/10.21105/joss.00957

Status badge

JOSS 10.21105/joss.00957

License

JOSS
The Journal of Open Source Software

Verde: Processing and gridding spatial data using Green's functions

Leonardo Uieda[1]

1 Department of Earth Sciences, SOEST, University of Hawai'i at Mānoa, Honolulu, Hawaii, USA

DOI: 10.21105/joss.00957

Software

- Review
- Repository
- Archive

Submitted: 14 September 2018
Published: 11 October 2018

Summary

Measurements made on the surface of the Earth are often sparse and unevenly distributed. For example, GPS displacement measurements are limited by the availability of ground stations and airborne geophysical measurements are highly sampled along flight lines but there is often a large gap between lines. Many data processing methods require data distributed on a uniform regular grid, particularly methods involving the Fourier transform or the computation of directional derivatives. Hence, the interpolation of sparse measurements onto a regular grid (known as *gridding*) is a prominent problem in the Earth Sciences.

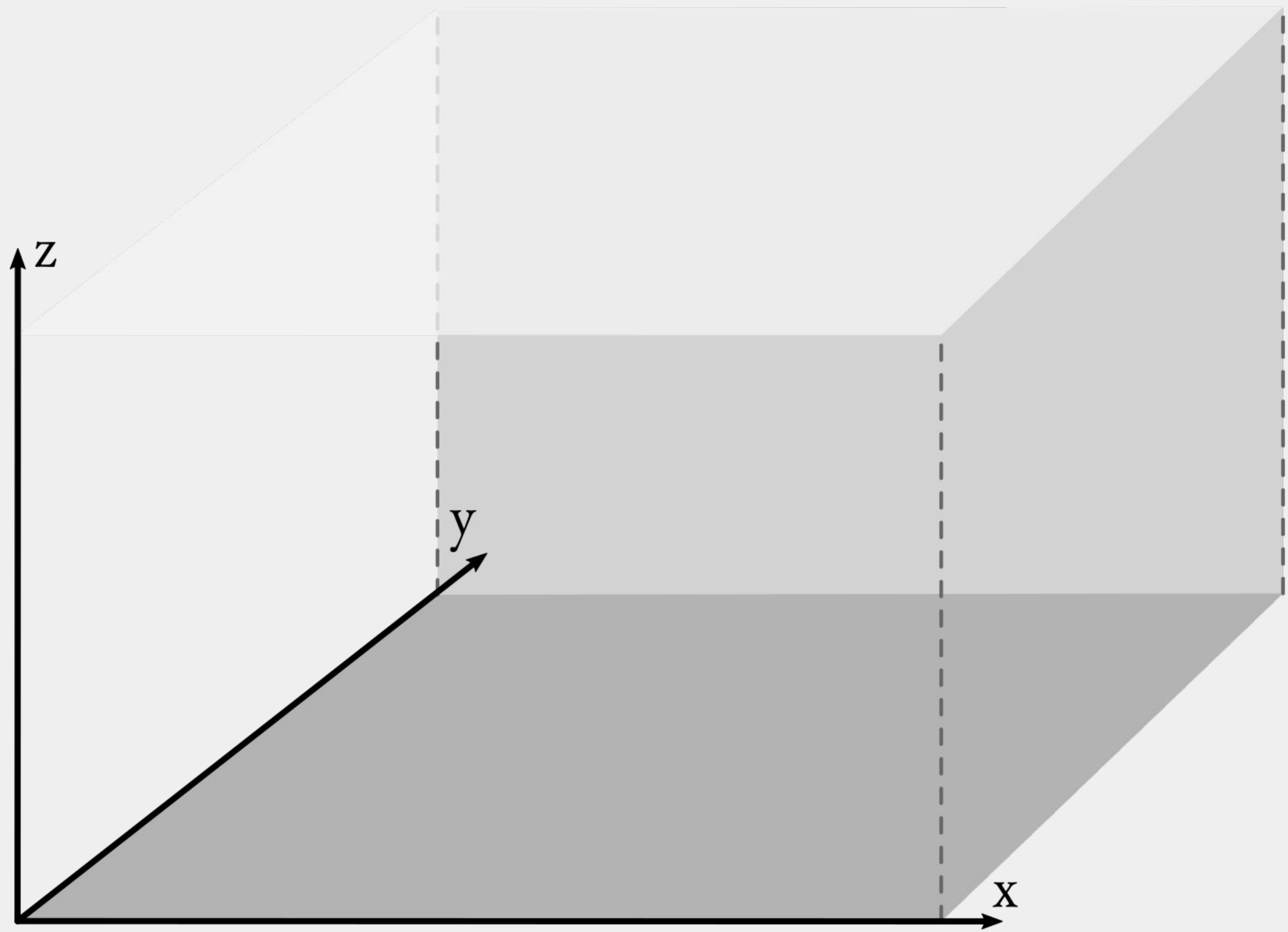
Popular gridding methods include kriging, minimum curvature with tension (W. Smith & Wessel, 1990), and bi-harmonic splines (D. T. Sandwell, 1987). The latter belongs to a group of methods often called *radial basis functions* and is similar to the *thin-plate spline* (Franke, 1982). In these methods, the data are assumed to be represented by a linear combination of Green's functions,

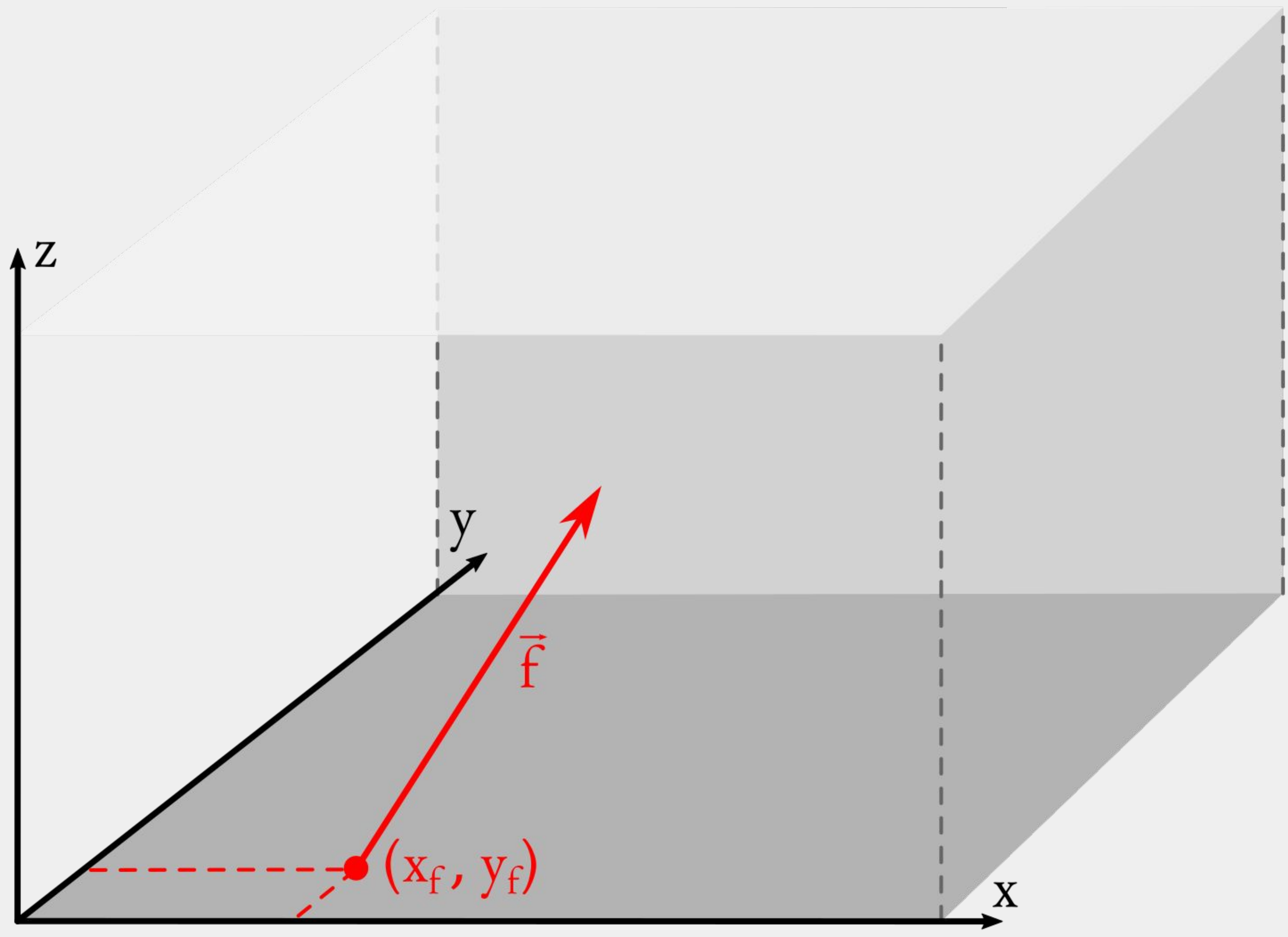
$$d_i = \sum_{j=1}^{M} p_j G(\mathbf{x}_i, \mathbf{x}_j),$$

in which d_i is the ith datum, p_j is a scalar coefficient, G is a Green's function, and $\mathbf{x}_i$ and $\mathbf{x}_j$ are the position vectors for the datum and the point defining the Green's function, respectively. Interpolation is done by estimating the M p_j coefficients through linear least-

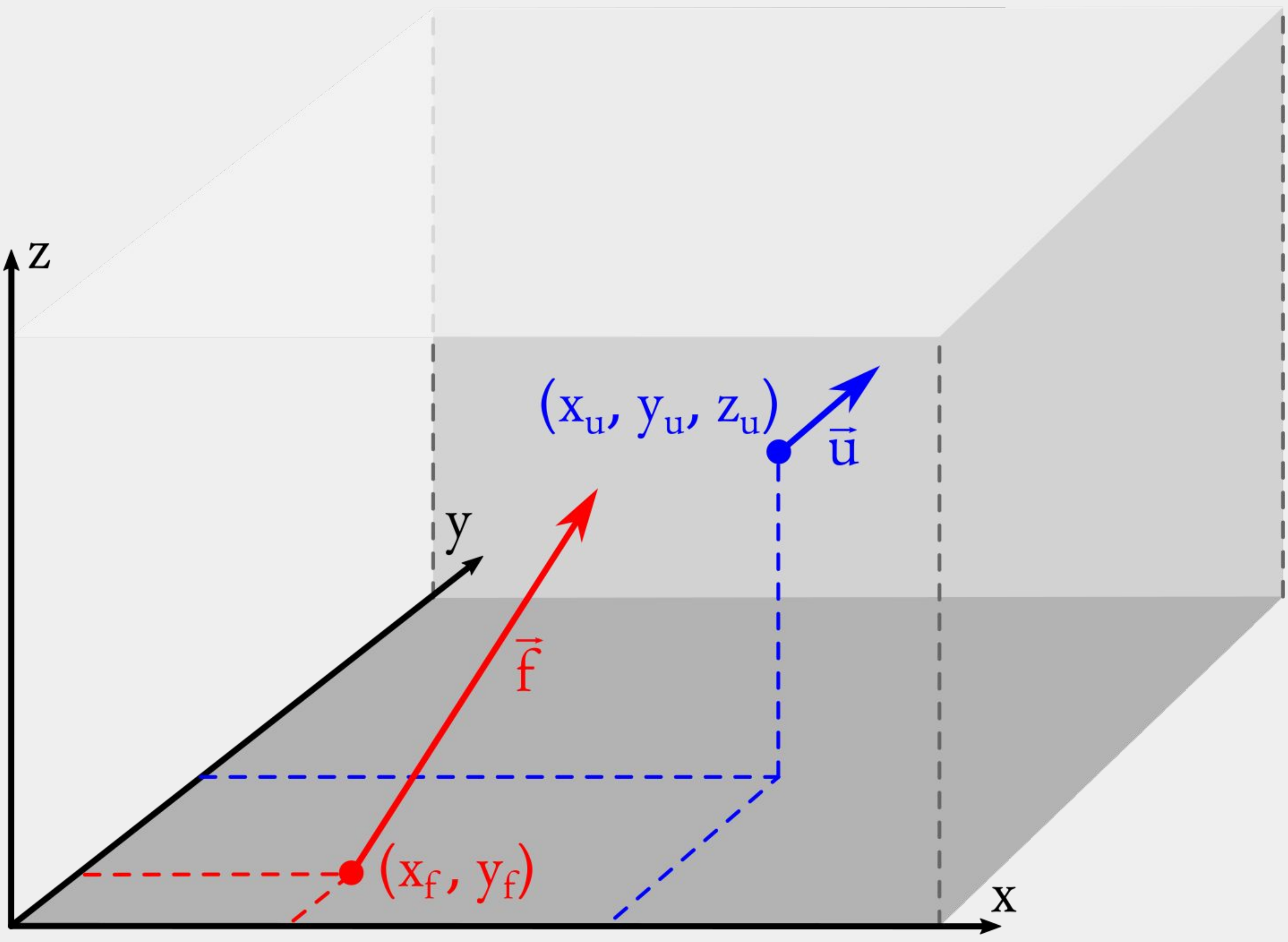
3-component GPS

Extension of Sandwell & Wessel (2016) GPS gridder to 3D


z
y
x



z
y
x
$\vec{f}$
(x_f, y_f)



z
y
x
(x_u, y_u, z_u)
$\vec{u}$
$\vec{f}$
(x_f, y_f)

$$\begin{bmatrix} G_{xx} & G_{xy} & G_{xz} \\ G_{yx} & G_{yy} & G_{yz} \\ G_{zx} & G_{zy} & G_{zz} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

Green's functions

$$\begin{bmatrix} G_{xx} & G_{xy} & G_{xz} \\ G_{yx} & G_{yy} & G_{yz} \\ G_{zx} & G_{zy} & G_{zz} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

(Okumura, 1995)

Green's functions force displacement

$$\begin{bmatrix} G_{xx} & G_{xy} & G_{xz} \\ G_{yx} & G_{yy} & G_{yz} \\ G_{zx} & G_{zy} & G_{zz} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

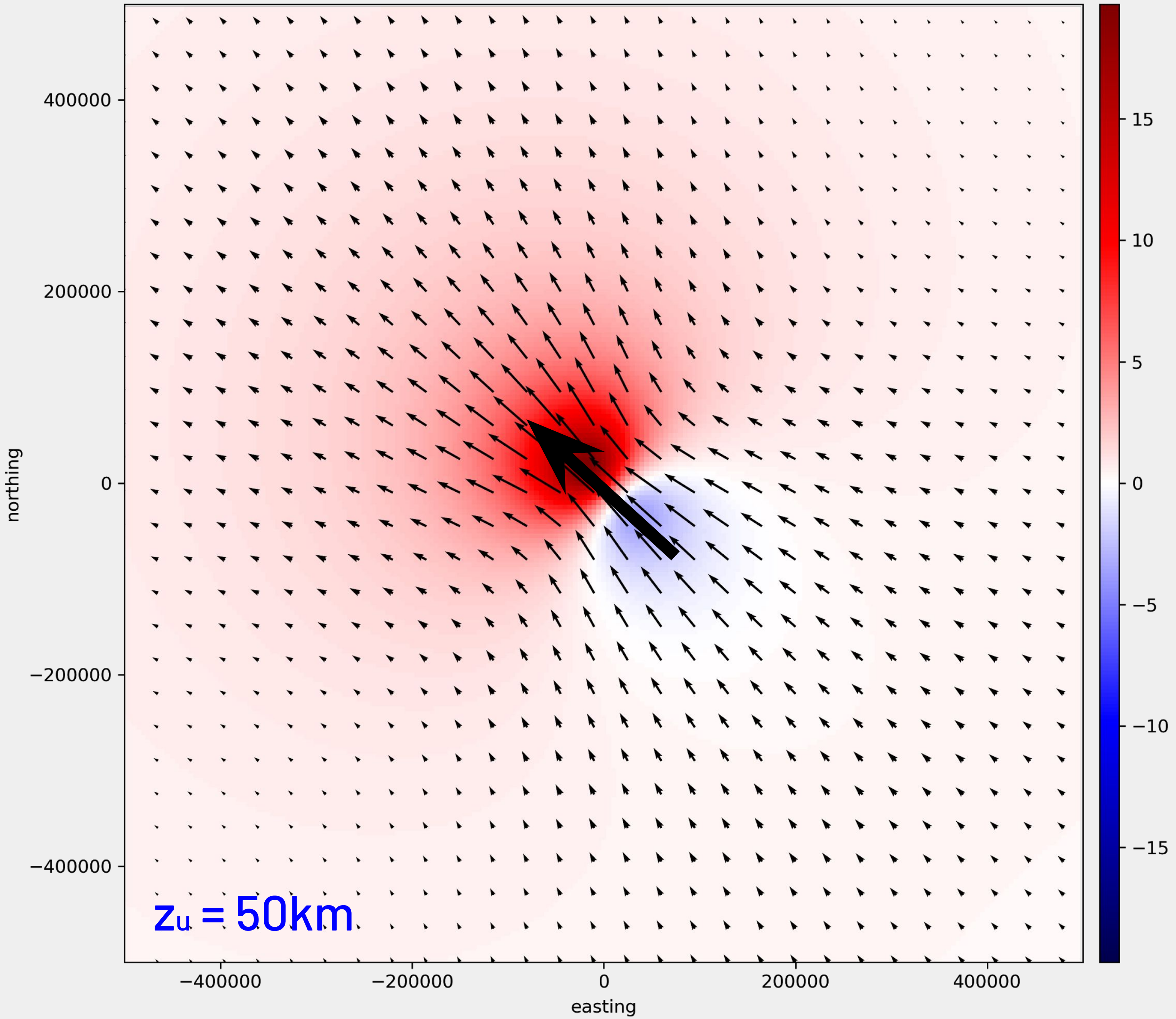
(Okumura, 1995)

Green's functions   force   displacement

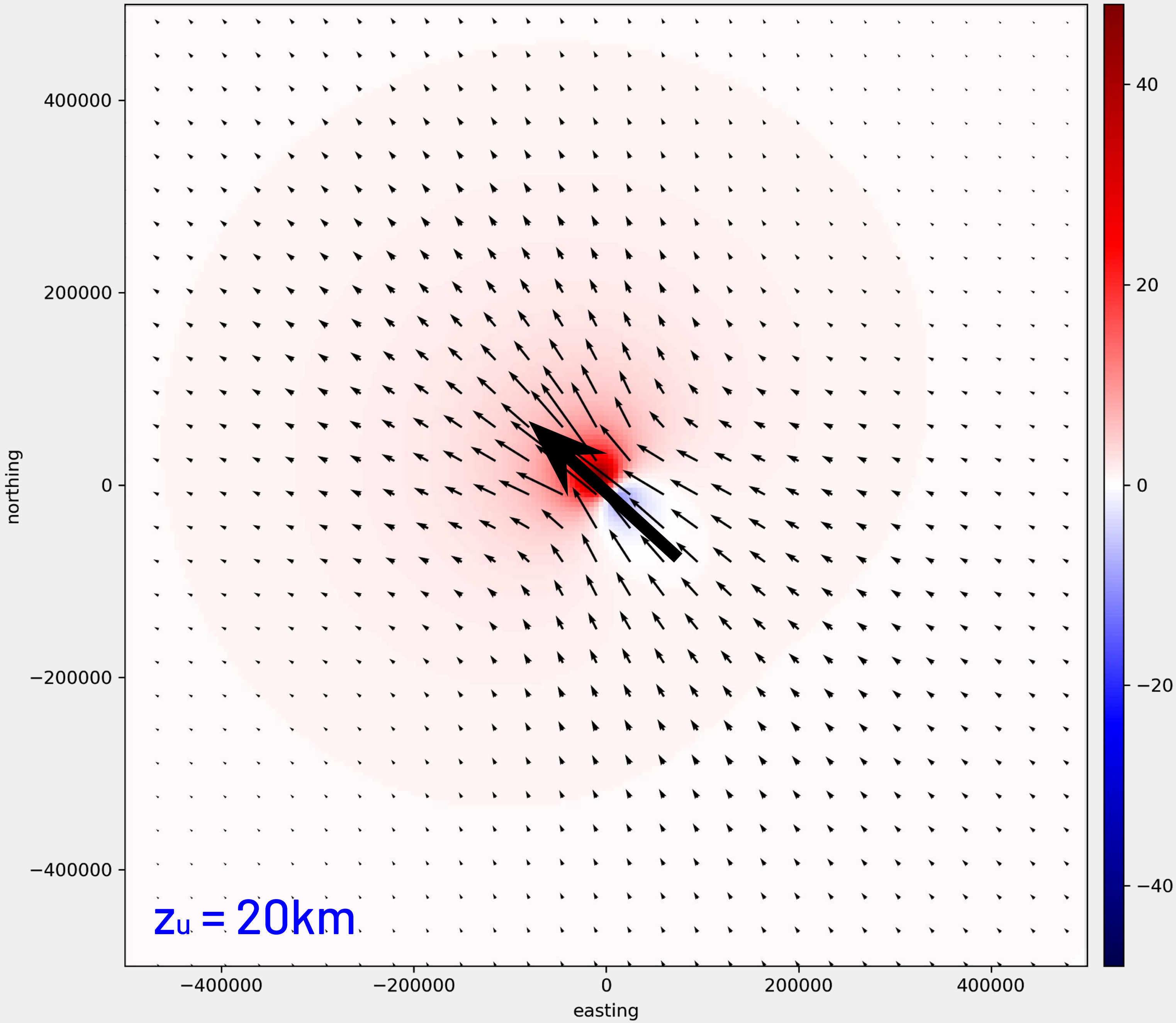
$$\begin{bmatrix} G_{xx} & G_{xy} & G_{xz} \\ G_{yx} & G_{yy} & G_{yz} \\ G_{zx} & G_{zy} & G_{zz} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

ML speak:

feature matrix   coefficients   labels



northing
400000
200000
0
−200000
−400000
−400000
−200000
0
200000
400000
easting
15
10
5
0
−5
−10
−15
zu = 50km

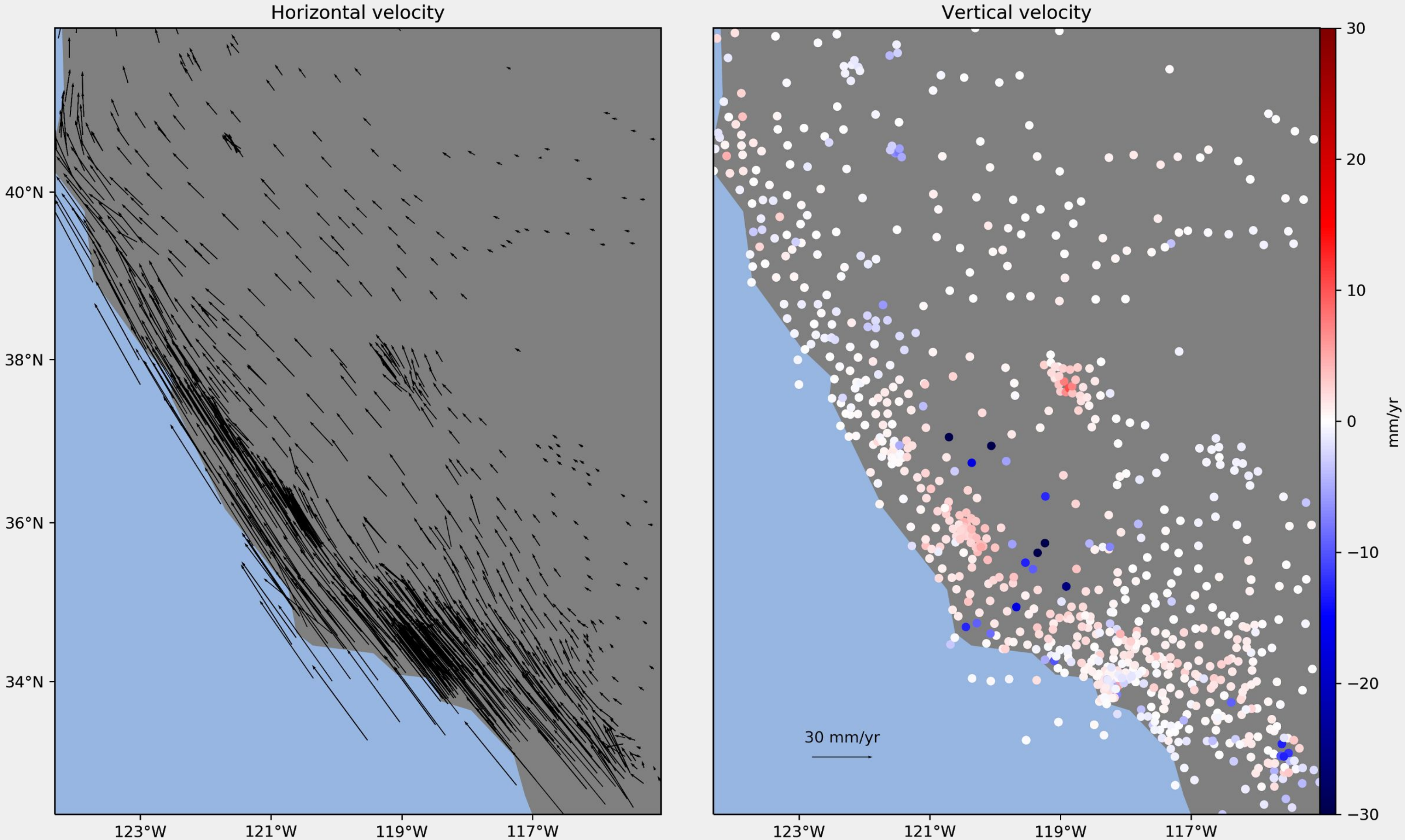


400000
200000
0
−200000
−400000
northing
−400000
−200000
0
200000
400000
easting
40
20
0
−20
−40
zu = 20km

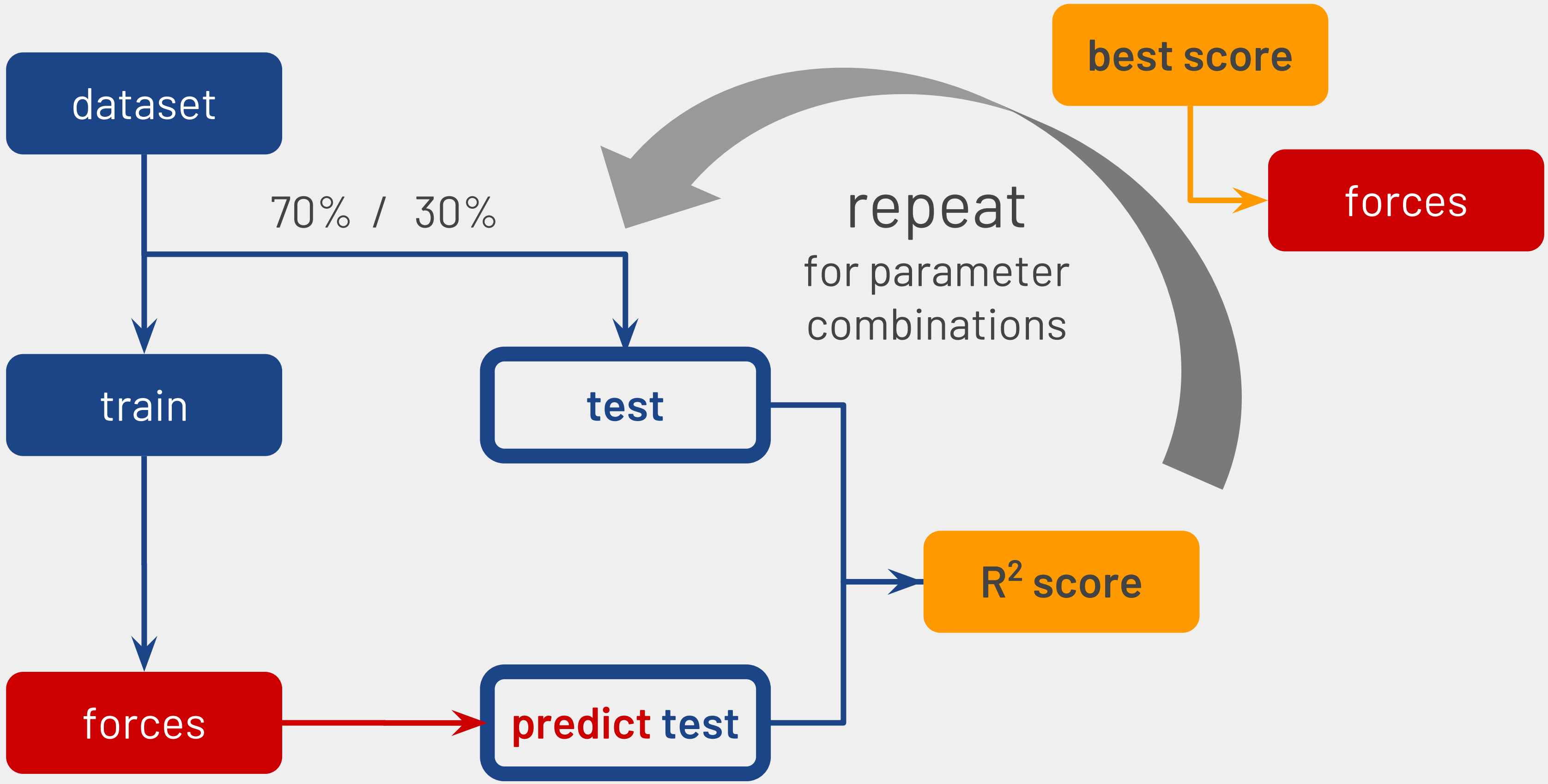
Controlling parameters:

regularization parameter	μ
height of displacements	z_u
Poisson's ratio	ν
force locations	(x_f, y_f)

Plate Boundary Observatory 2017 data

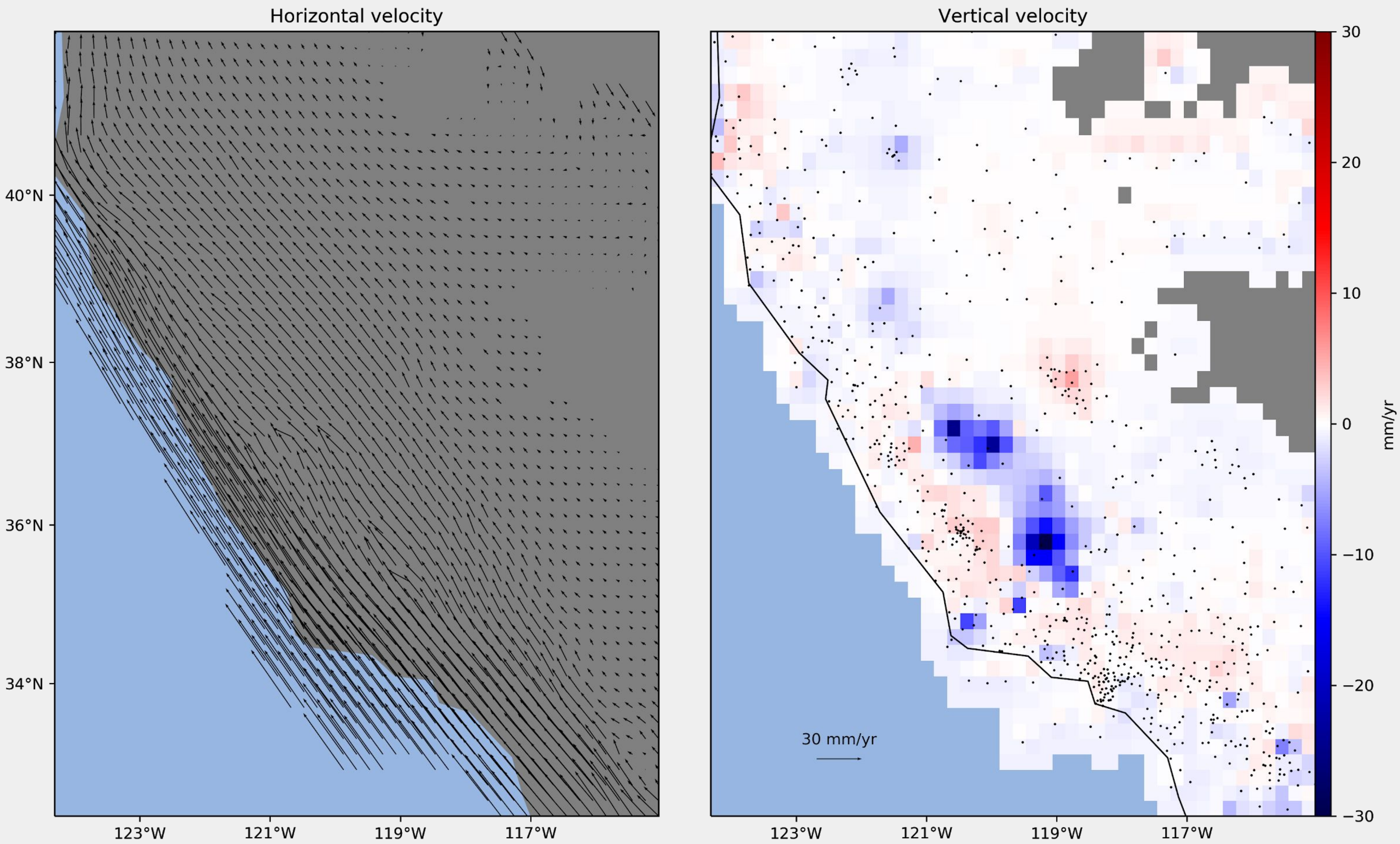


Automatic tuning:



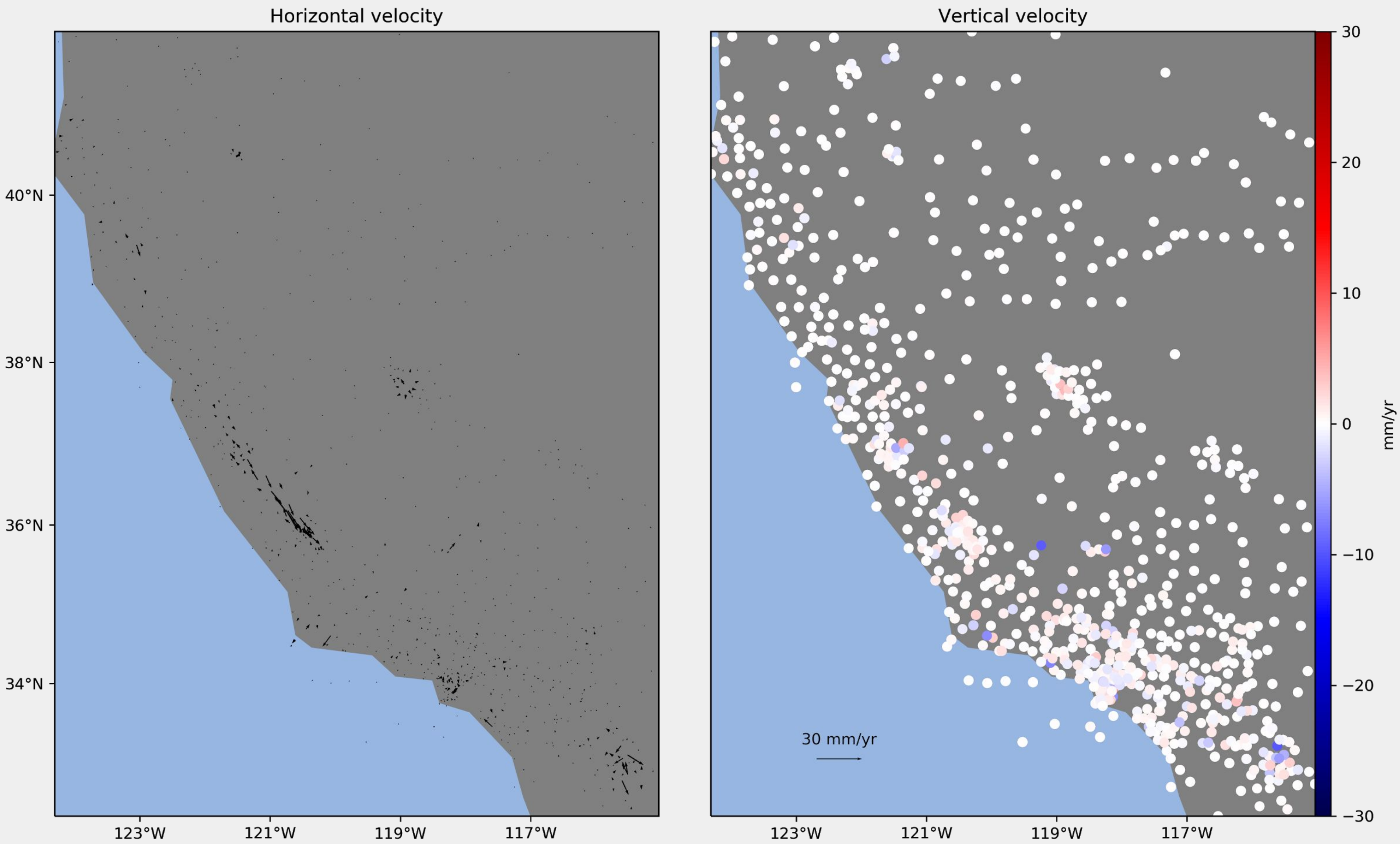
Automatic tuning:

best score (R^2)	0.91
configurations tested	120
regularization parameter	50
height of displacements	10 km
Poisson's ratio	0.5
force locations (fixed)	same as data



Horizontal velocity
Vertical velocity
40°N
38°N
36°N
34°N
123°W
121°W
119°W
117°W
30 mm/yr
30
20
10
0
−10
−20
−30
mm/yr

residuals



Horizontal velocity
Vertical velocity
40°N
38°N
36°N
34°N
123°W
121°W
119°W
117°W
30 mm/yr
mm/yr
30
20
10
0
−10
−20
−30

Main points:

Coupled 3-component gridding works

Even if physics is not exact

Use weights to account for uncertainty

Automatic tuning == easy to use

Large memory footprint

Main points:

Coupled 3-component gridding works

Even if physics is not exact

Use weights to account for uncertainty

Automatic tuning == easy to use

Large memory footprint

Future work

Tune location of forces

Larger datasets

Comparisons with other methods

Conclusions

ML == Automation

Data selection and sorting

Identification of anomalies/faults/features

ML == Automation

- *Data selection and sorting*
- *Identification of anomalies/faults/features*

Open-source is the future (mostly Python)

- *scikit-learn is most popular*
- *TensorFlow (Google)*
- *PyTorch (Facebook)*

ML == Automation

Data selection and sorting

Identification of anomalies/faults/features

Open-source is the future (mostly Python)

scikit-learn is most popular

TensorFlow (Google)

PyTorch (Facebook)

Borrow techniques for geophysical inversion

Model selection and validation

Equivalent layer

BEWARE OF OVERFITTING

ALWAYS keep some data for validation

If automatically tuning, split 3 ways

BEWARE OF OVERFITTING

ALWAYS keep some data for validation

If automatically tuning, split 3 ways

Models are only as good as training data

Neural networks need a lot of data

Data is the new gold

Where human bias creeps in

BEWARE OF OVERFITTING

ALWAYS keep some data for validation

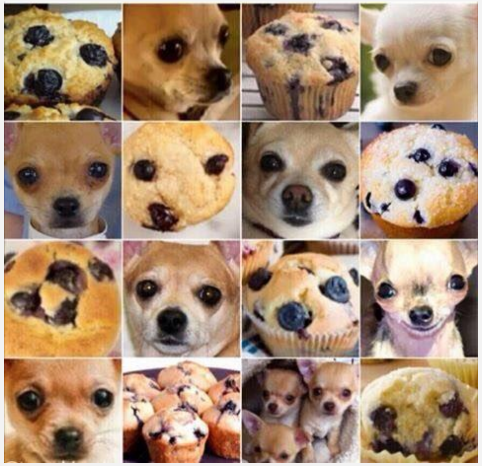
If automatically tuning, split 3 ways

Models are only as good as training data

Neural networks need a lot of data

Data is the new gold

Where human bias creeps in



Acknowledgments

GPS gridding:

Collaborators: Paul Wessel, Xiaohua (Eric) Xu, David Sandwell

NSF-EAR grant #1829371 (Wessel, Smith-Konter, Uieda)

The Leading Edge tutorials (started by Matt Hall)

Jake VanderPlas' excelent blog (jakevdp.github.io)

Slides at **leouieda.com**

Oct 12 2018