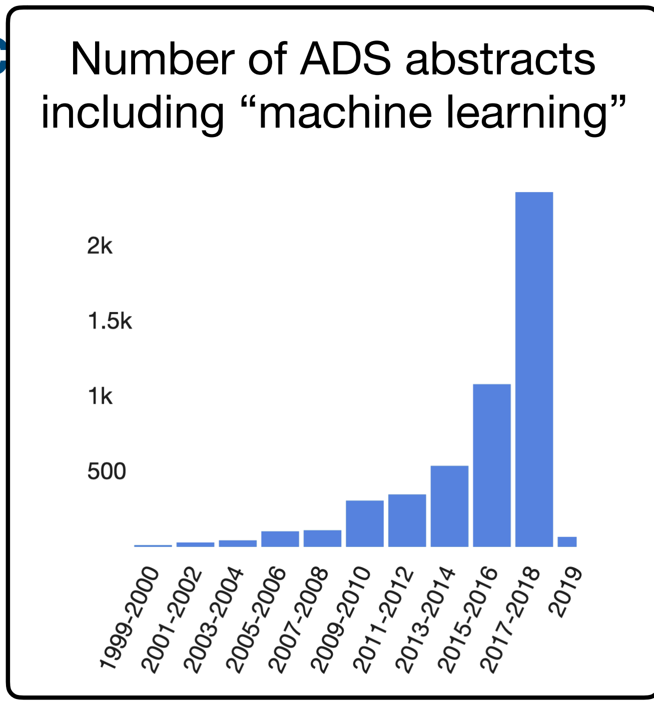
A Typical User's Ground-Level Perspective on Machine Learning in Astronomy

James R. A. Davenport

DiRAC
University of Washington

A Typical User's Ground-Level Perspective on Mac ronomy



Machine Learning is Boring*

Machine Learning is Boring*

*normal, lots of people doing it.
And this is a good thing! (i.e. don't be scared!)

Most of our work isn't "big data"

(i.e. can run most algorithms on your laptop)

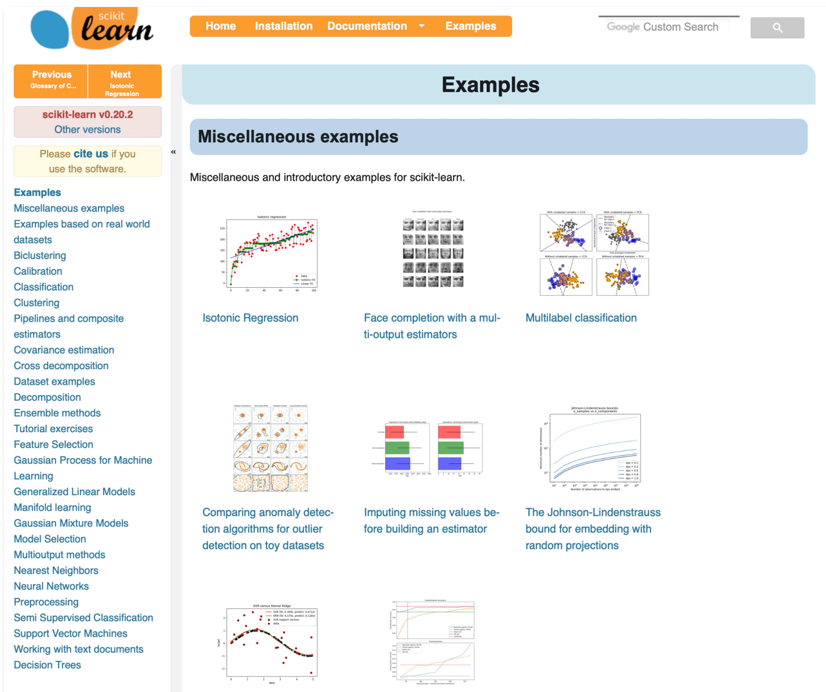
Yet sample size is big enough that
interesting/rare things can be found

Our data is becoming better suited for ML

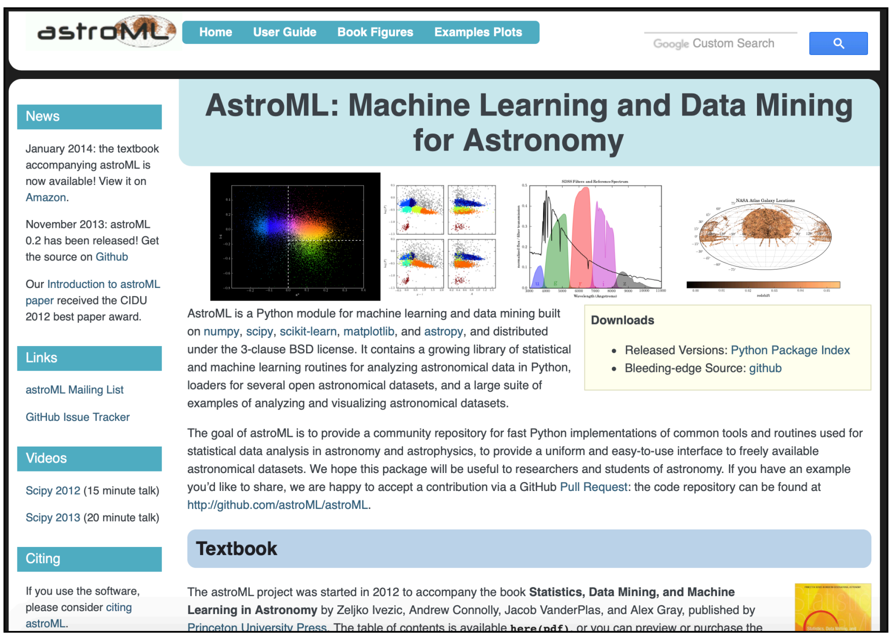
- big datasets (Mario’s talk), especially Gaia!
- easier than ever to get data (Vizier/Xmatch, ADS, journals, Github, Zenodo…)
- value-added datasets for surveys (e.g. stellar parameters from SDSS)

ML is easier than ever to use

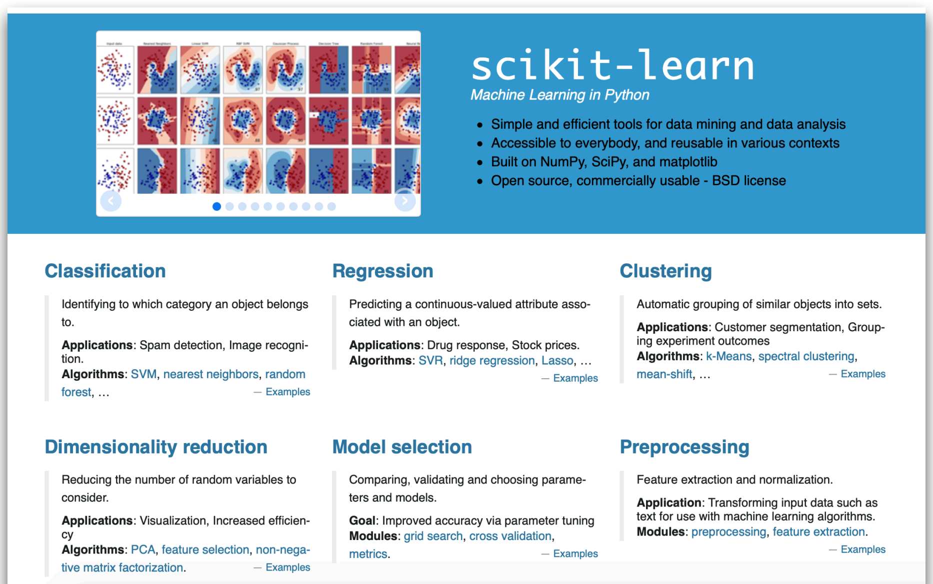
- robust, open source libraries available
- many programming languages
- many domain (astro) experts & workshops available


scikit learn
Home
Installation
Documentation
Examples
Google Custom Search
Previous
Next
scikit-learn v0.20.2
Other versions
Please cite us if you use the software.
Examples
Miscellaneous examples
Examples based on real world datasets
Biclustering
Calibration
Classification
Clustering
Pipelines and composite estimators
Covariance estimation
Cross decomposition
Dataset examples
Decomposition
Ensemble methods
Tutorial exercises
Feature Selection
Gaussian Process for Machine Learning
Generalized Linear Models
Manifold learning
Gaussian Mixture Models
Model Selection
Multioutput methods
Nearest Neighbors
Neural Networks
Preprocessing
Semi Supervised Classification
Support Vector Machines
Working with text documents
Decision Trees
Examples
Miscellaneous examples
Miscellaneous and introductory examples for scikit-learn.
Isotonic Regression
Face completion with a multi-output estimators
Multilabel classification
Comparing anomaly detection algorithms for outlier detection on toy datasets
Imputing missing values before building an estimator
The Johnson-Lindenstrauss bound for embedding with random projections

also astroML!
(see Brigitta's talk later)

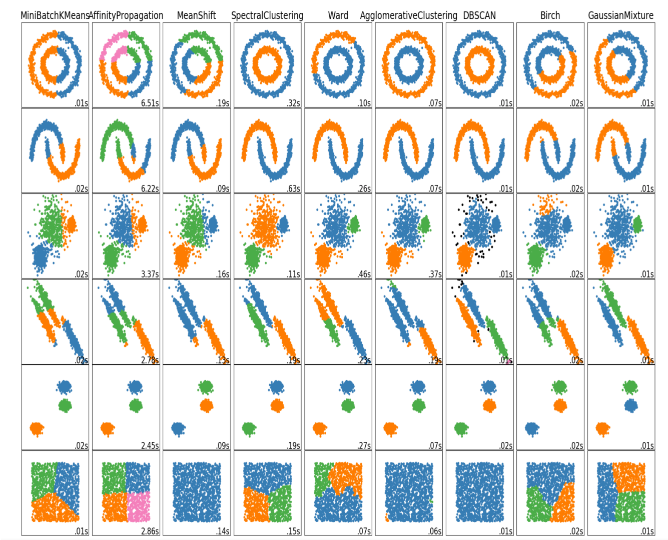


Problems that ML is good for:

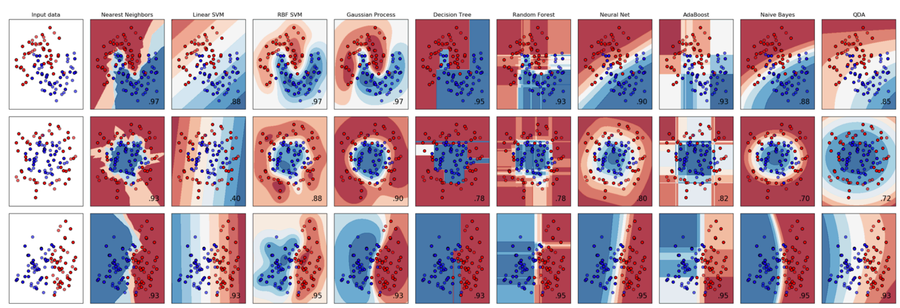


Each algorithm has specific use cases
(sometimes: just try them all!)

Clustering



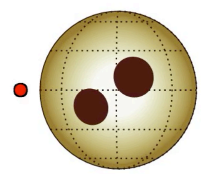
Classification

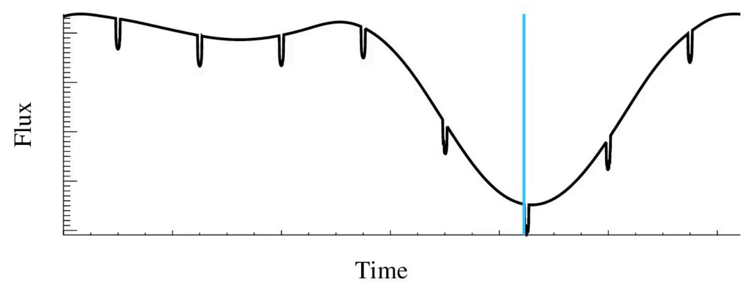


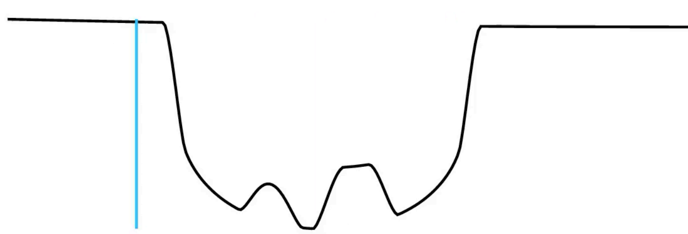
Goal: demonstrate 3 real problems that ML can be used for

Example 1: Clustering starspot evolution tracks

Kepler 17



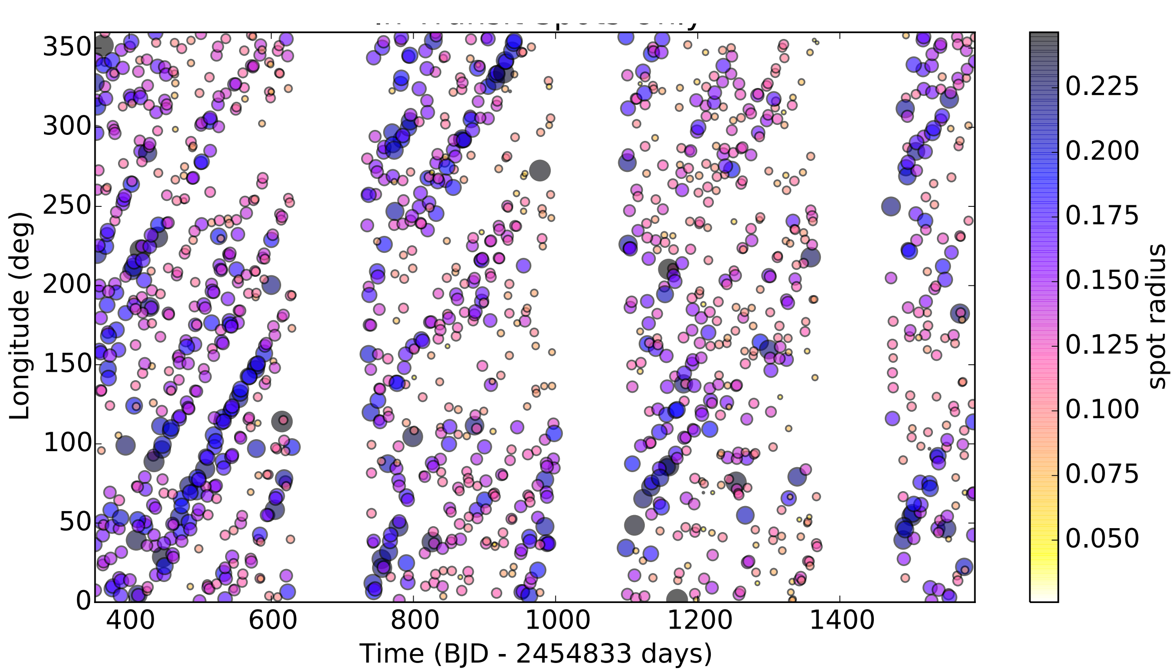




Davenport, Hebb, & Hawley (2015)

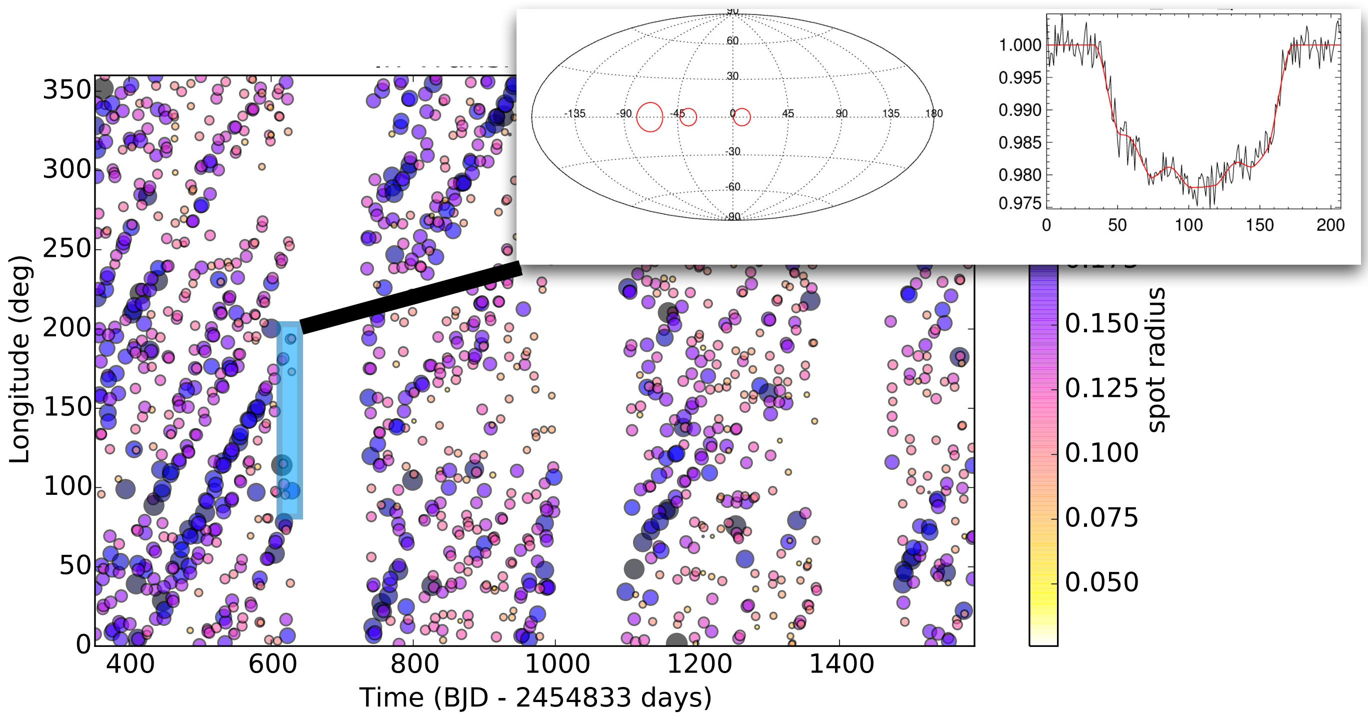
Example 1: Clustering starspot evolution tracks

Kepler 17



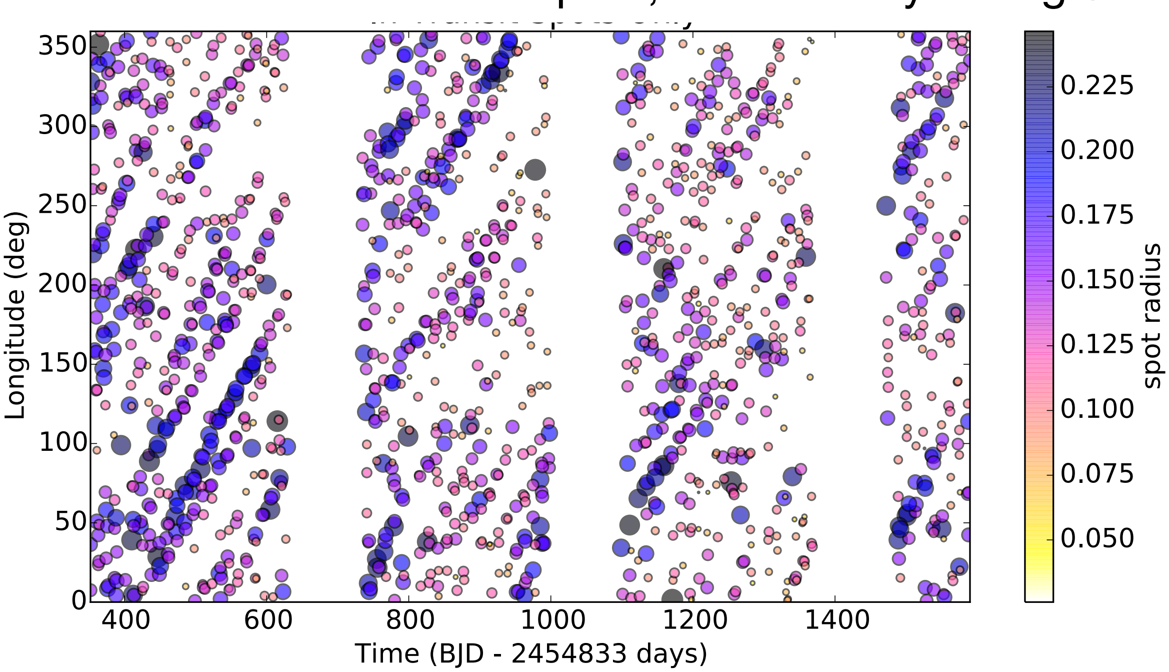
Example 1: Clustering starspot evolution tracks

Kepler 17



Example 1: Clustering starspot evolution tracks Kepler 17

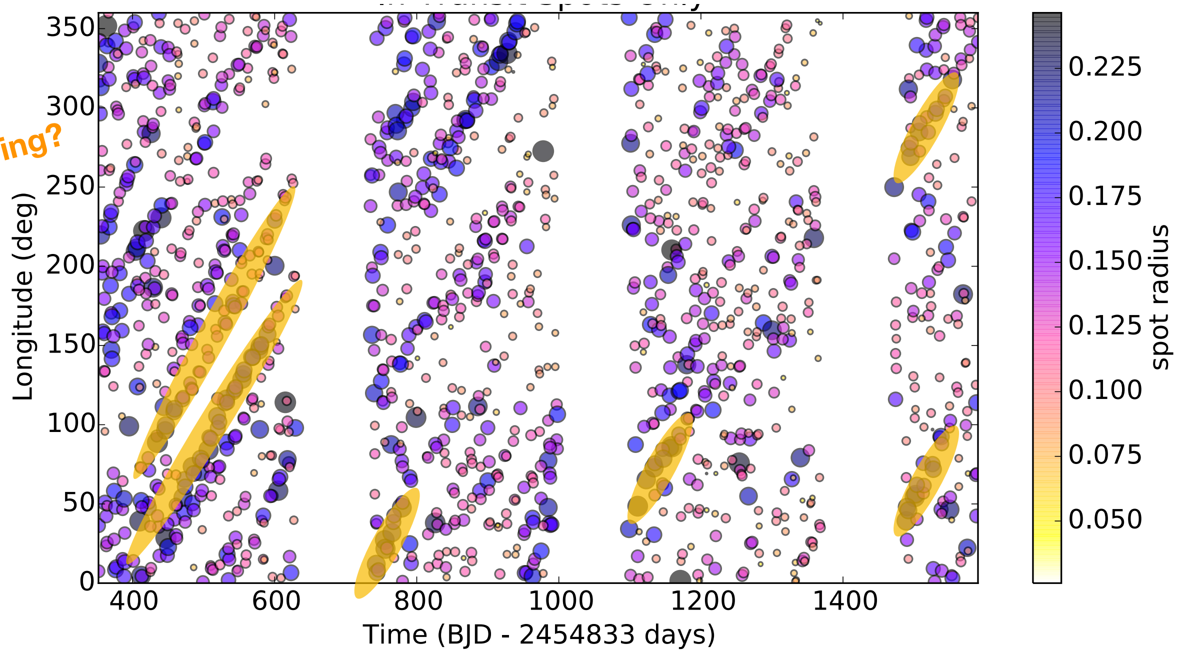
Question: Which tracks are starspots, how do they emerge/decay?



Example 1: Clustering starspot evolution tracks Kepler 17

Question: Which tracks are starspots, how do they emerge/decay?

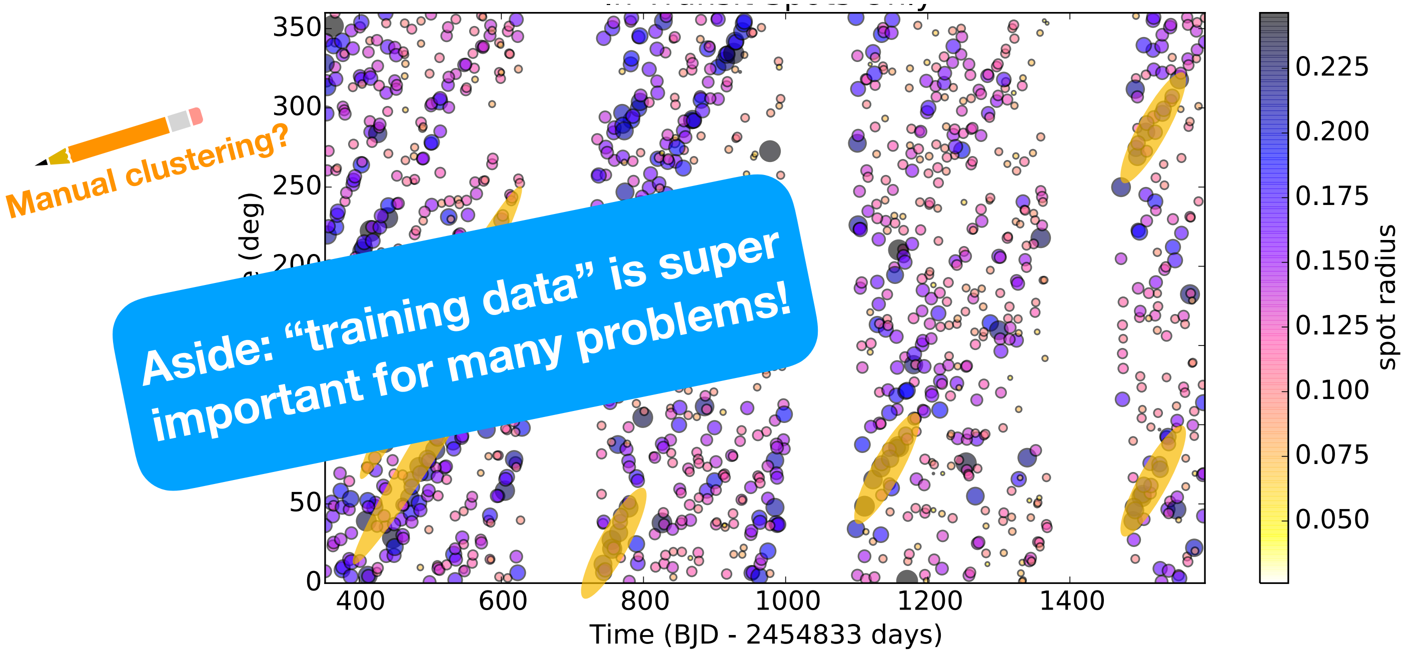
Manual clustering?



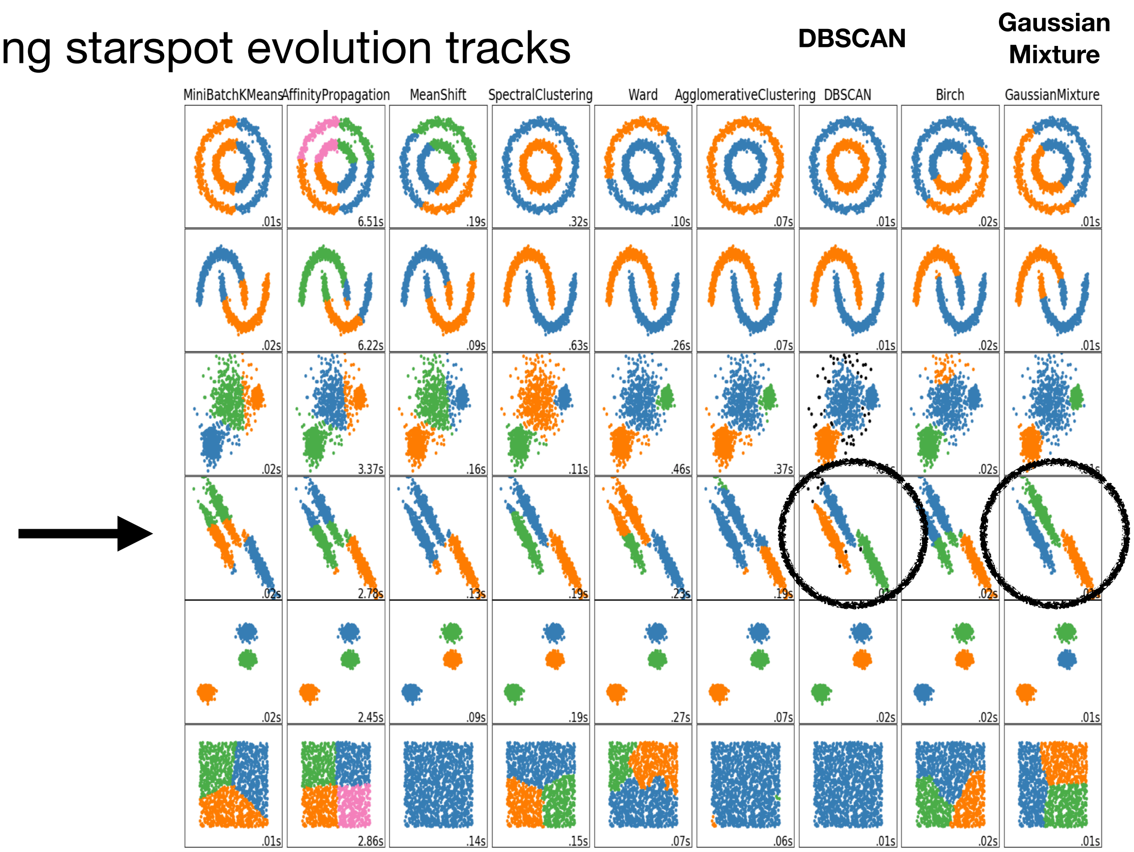
Example 1: Clustering starspot evolution tracks

Kepler 17

Question: Which tracks are starspots, how do they emerge/decay?

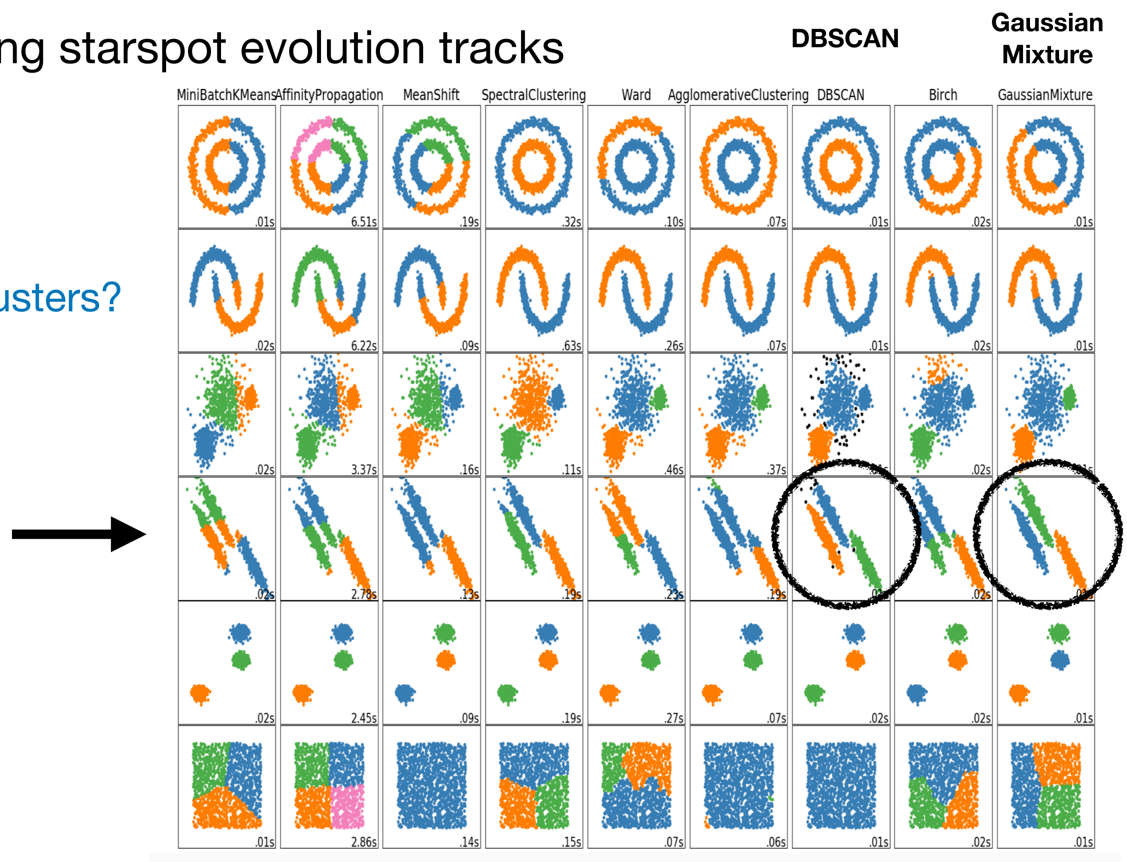


Example 1: Clustering starspot evolution tracks



Example 1: Clustering starspot evolution tracks

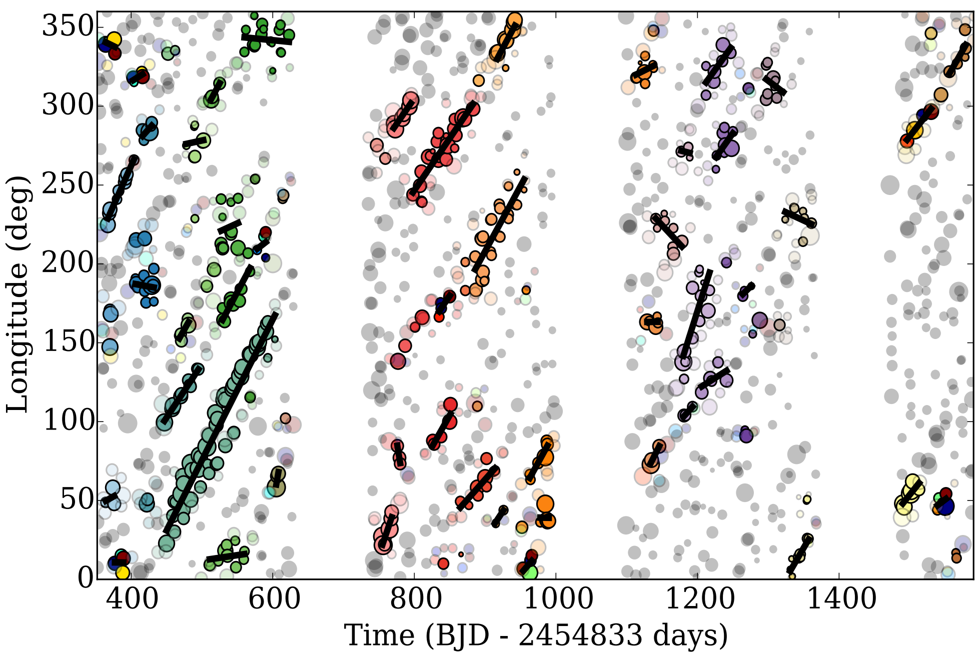
need to predefine Nclusters?



Example 1: Clustering starspot evolution tracks

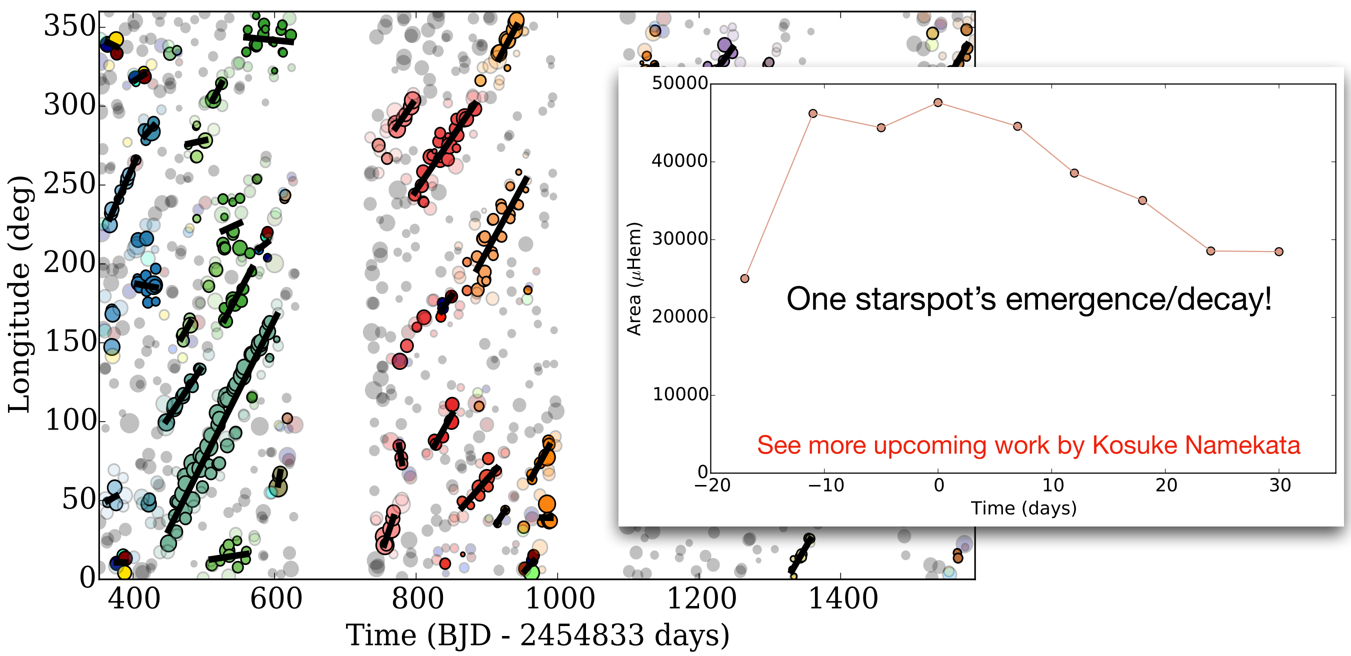
Kepler 17

DBSCAN: Density-based spatial clustering of applications with noise



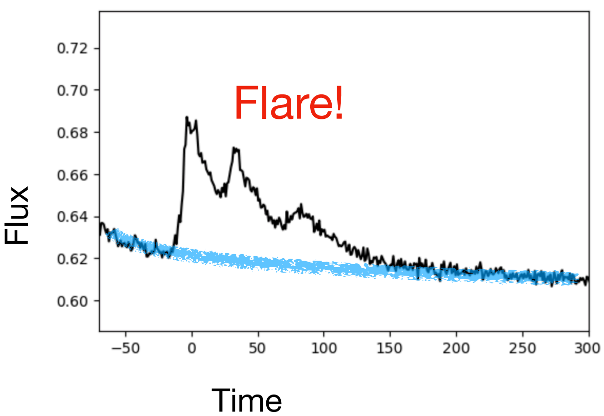
Example 1: Clustering starspot evolution tracks — Kepler 17

DBSCAN: Density-based spatial clustering of applications with noise



Example 2: Modeling a complex stellar flare

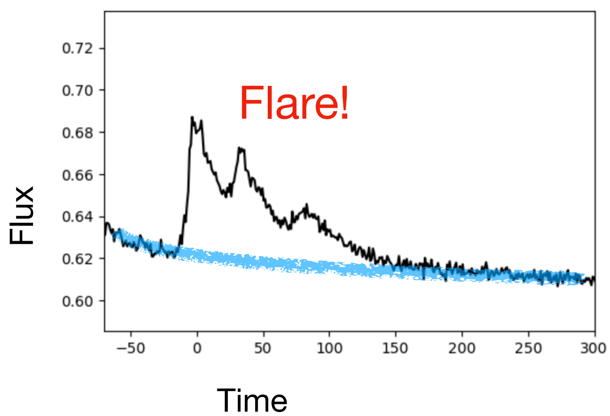
Question: Is there (quasi-) sinusoidal behavior in the flare decay?

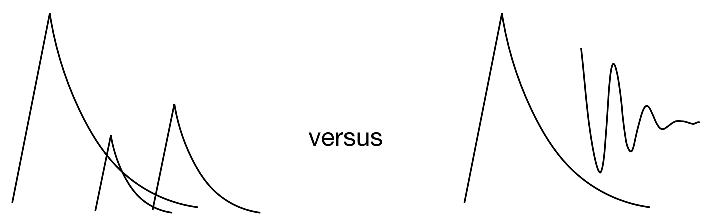


https://github.com/RileyWClarke/QPP-GP

Example 2: Modeling a complex stellar flare

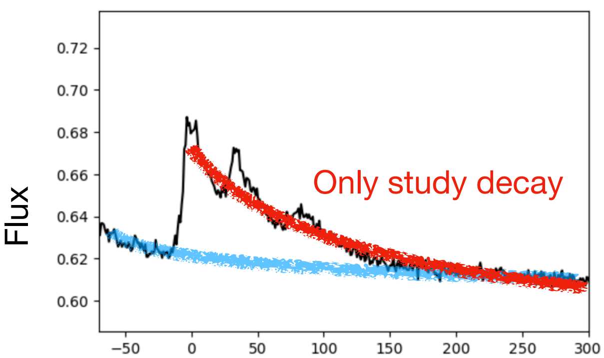
Question: Is there (quasi-) sinusoidal behavior in the flare decay?



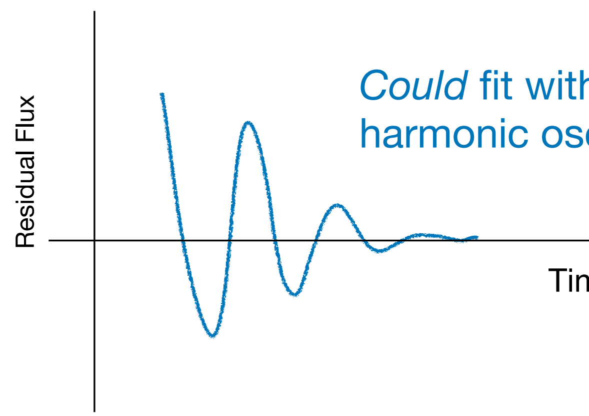


https://github.com/RileyWClarke/QPP-GP

Example 2: Modeling a complex stellar flare



Question: Is there (quasi-) sinusoidal behavior in the flare decay?



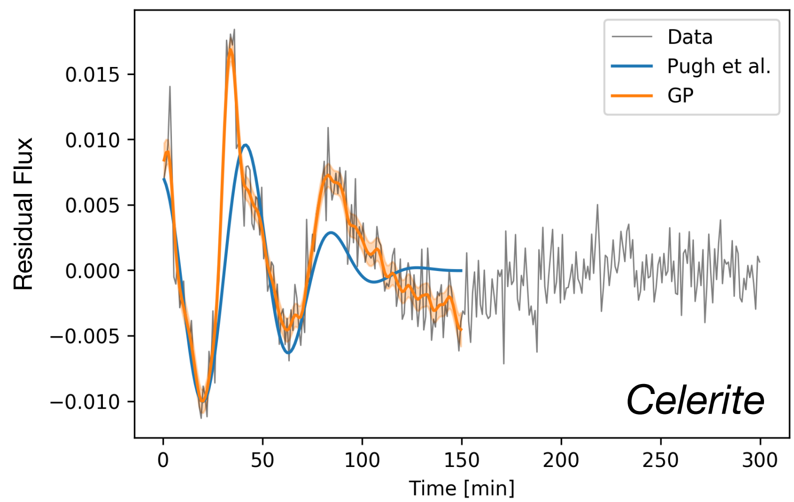
Could fit with a damped harmonic oscillator

Difficult to classify sinusoidal vs. stochastic, & strict vs quasi sinusoid

https://github.com/RileyWClarke/QPP-GP

Example 2: Modeling a complex stellar flare

Gaussian Process

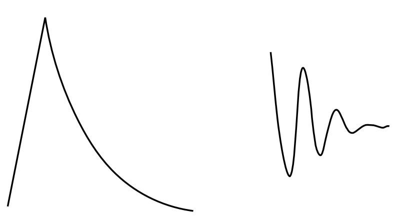


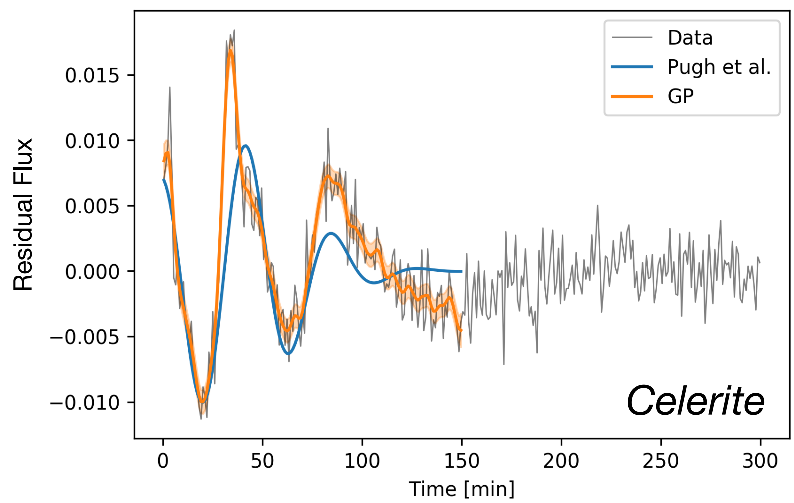
https://github.com/RileyWClarke/QPP-GP

Example 2: Modeling a complex stellar flare

Gaussian Process

Use an exponential +
simple-harmonic-oscillator *kernel*



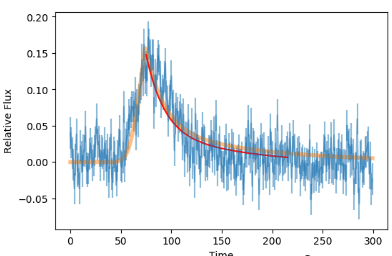


https://github.com/RileyWClarke/QPP-GP

Example 2: Modeling a complex stellar flare

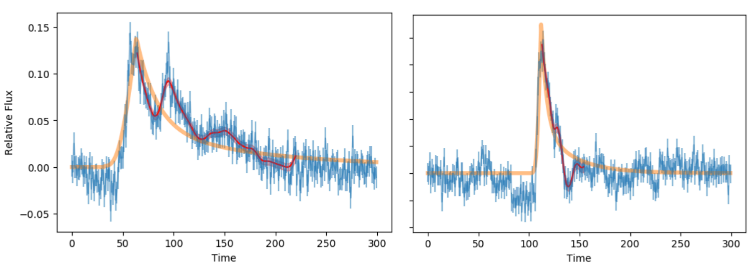
Gaussian Process

No Period



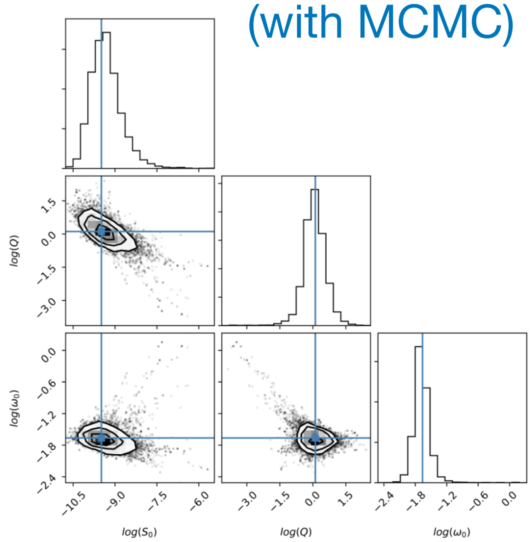
Objective search

Candidate Periods



https://github.com/RileyWClarke/QPP-GP

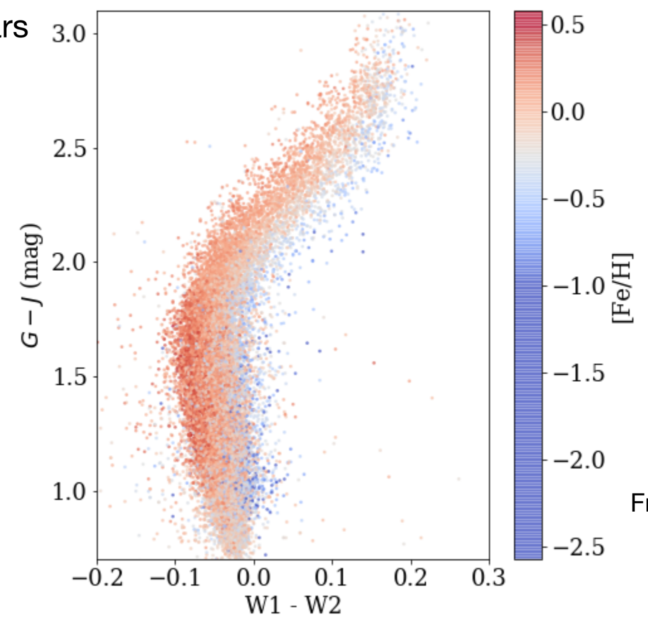
Robust uncertainties!
(with MCMC)



Example 3: Modeling photometric metallicities

Observation: [Fe/H] gradient in stars

Gaia DR2 + (WISE + 2MASS) + APOGEE



[3.4] - [4.6]

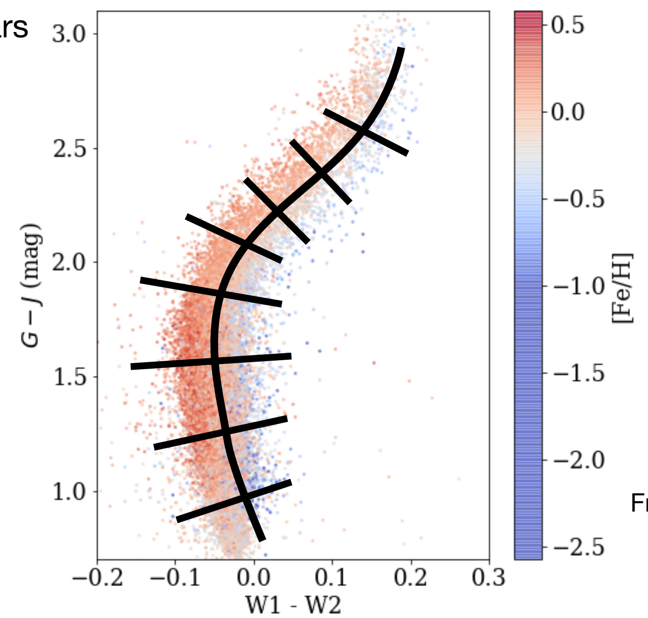
From APOGEE

https://github.com/jradavenport/ingot/

Example 3: Modeling photometric metallicities

Observation: [Fe/H] gradient in stars

We could build a complex polynomial or spline model



[3.4] - [4.6]

From APOGEE

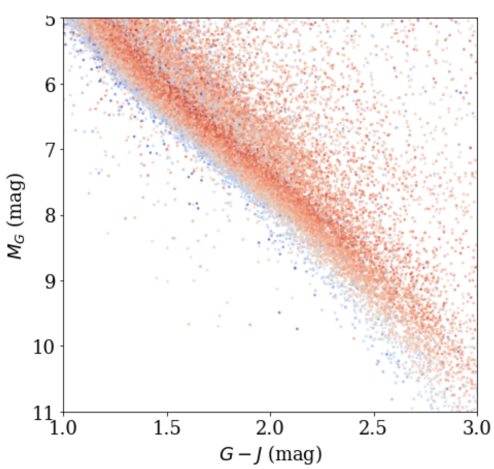
https://github.com/jradavenport/ingot/

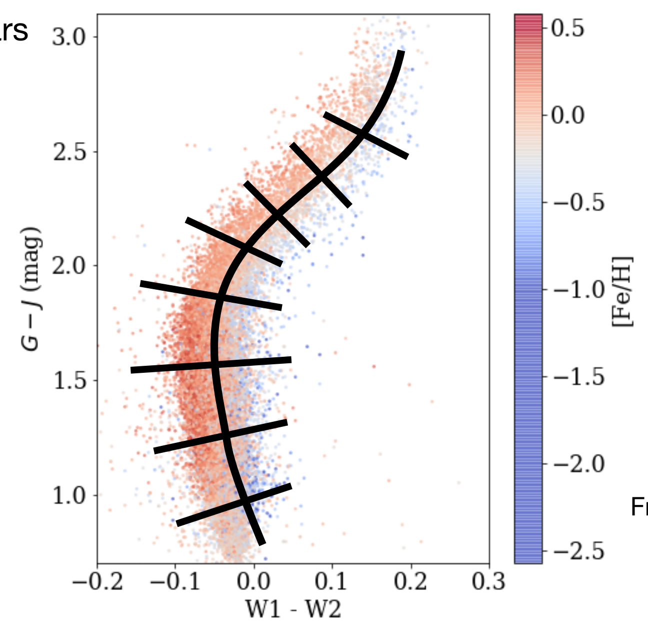
Example 3: Modeling photometric metallicities

Observation: [Fe/H] gradient in stars

We could build a complex polynomial or spline model

Tedious, and difficult to add additional dimensions!





[3.4] - [4.6]

From APOGEE

https://github.com/jradavenport/ingot/

jradavenport

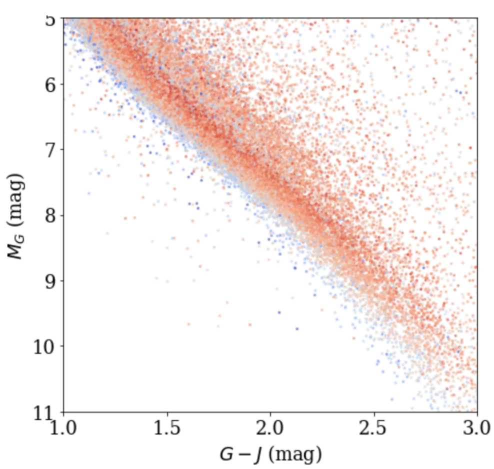
Example 3: Modeling photometric metallicities

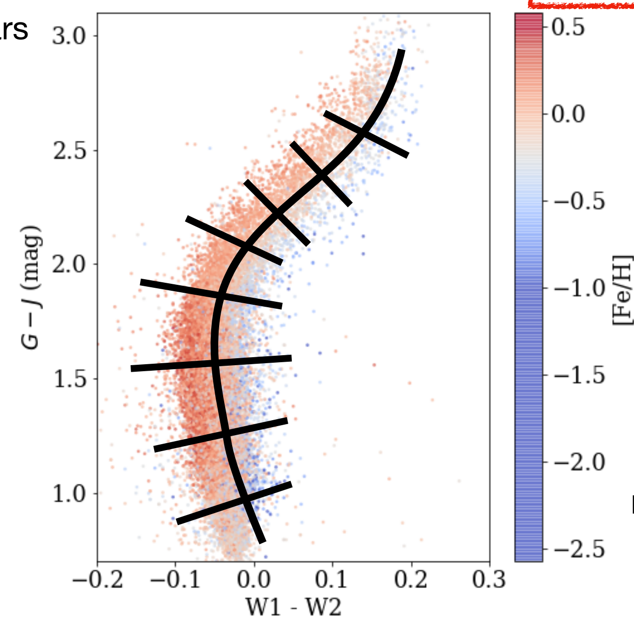
Or use a simple, flexible ML model!

Observation: [Fe/H] gradient in stars

We could build a complex polynomial or spline model

Tedious, and difficult to add additional dimensions!





[3.4] - [4.6]

From APOGEE

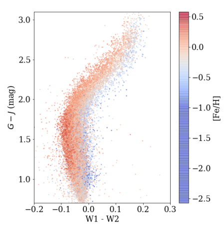
https://github.com/jradavenport/ingot/

jradavenport

Example 3: Modeling photometric metallicities

KNearestNeighbors

Xdata = (*G-J*, W1-W2)
Ydata = [Fe/H]

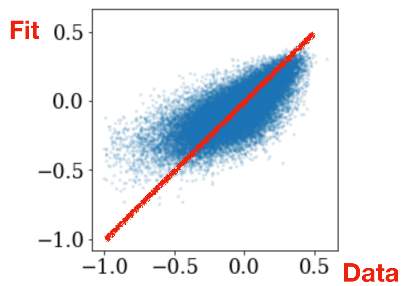



```
In [43]: model = KNeighborsRegressor(n_neighbors=5)

         model.fit(Xdata, Ydata)
         newY = model.predict(Xdata)

         plt.figure(figsize=(4,4))
         plt.scatter(Ydata, newY, s=3, alpha=0.14)
         # plt.xlim(-.5,.5)
         plt.ylim(plt.xlim())

Out[43]: (-1.0819262972727413, 0.6647018304720514)
```

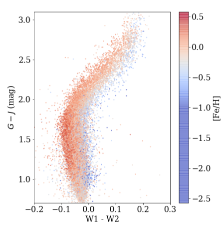


https://github.com/jradavenport/ingot/
jradavenport

Example 3: Modeling photometric metallicities

KNearestNeighbors

Xdata = (*G-J*, W1-W2)
Ydata = [Fe/H]

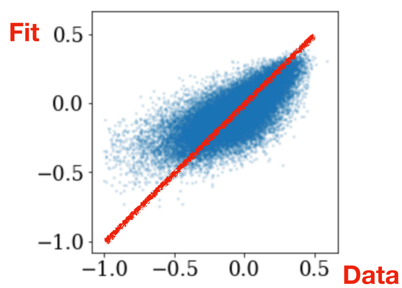



```
In [43]: model = KNeighborsRegressor(n_neighbors=5)

         model.fit(Xdata, Ydata)
         newY = model.predict(Xdata)

         plt.figure(figsize=(4,4))
         plt.scatter(Ydata, newY, s=3, alpha=0.14)
         # plt.xlim(-.5,.5)
         plt.ylim(plt.xlim())

Out[43]: (-1.0819262972727413, 0.6647018304720514)
```

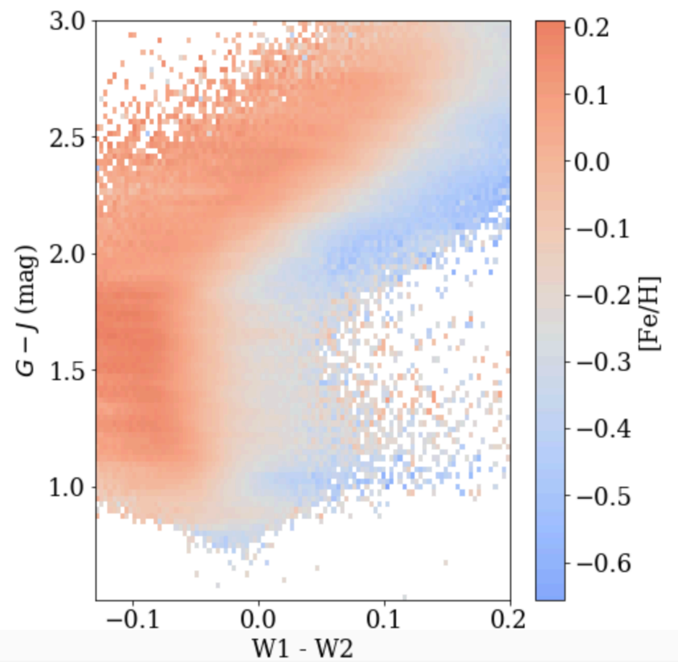


Example 3: Modeling photometric metallicities

KNearestNeighbors

<u>Result</u>: a simple to use "surface", no tweaking for shape/order, extend to addtional dimensions easily

1 Million new stars with no spectra

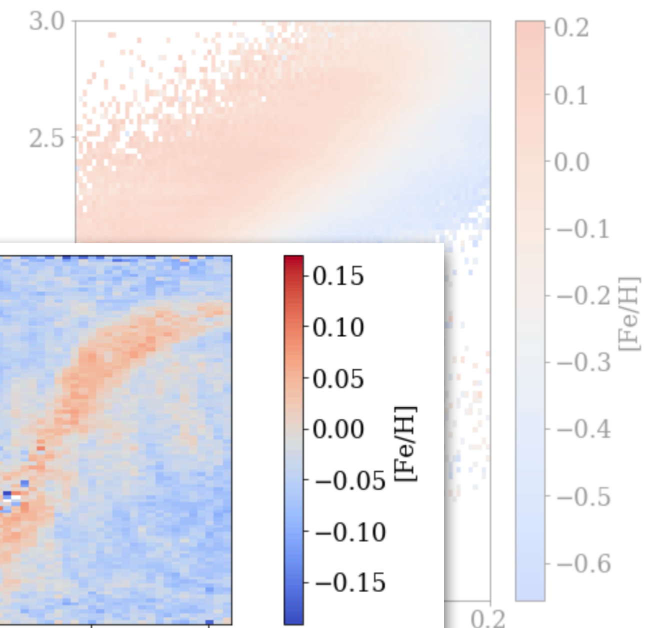


https://github.com/jradavenport/ingot/

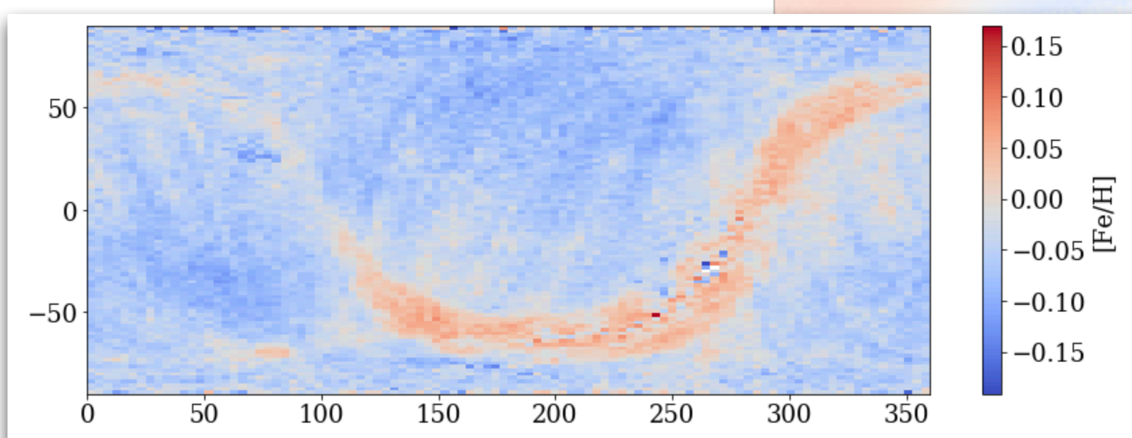
Example 3: Modeling photometric metallicities

KNearestNeighbors

1 Million new stars with no spectra

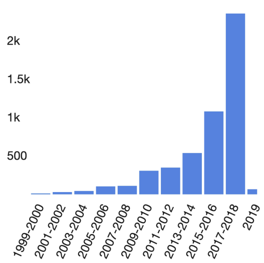


Result: a simple to use "surface", no tweaking for shape/order, extend to addtional dimensions easily



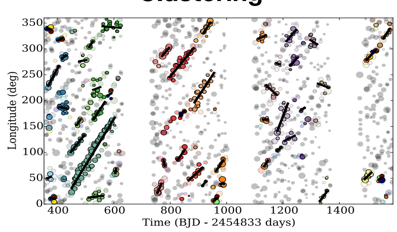
https://github.com/jradavenport/ingot/

Conclusions

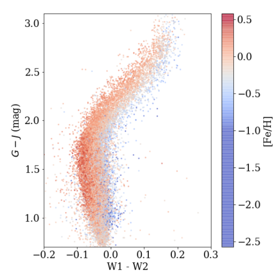


ML is easier and more “boring” than ever!

Clustering



Regression



GP’s