

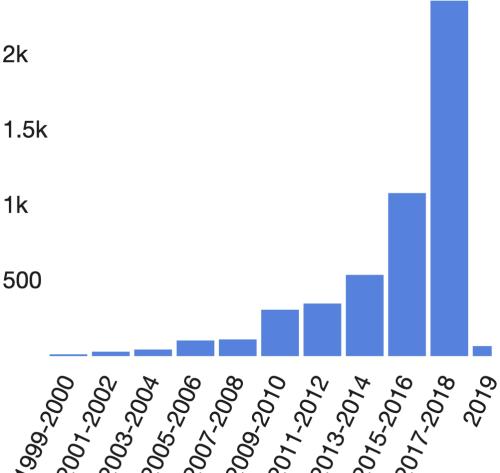
A Typical User's Ground-Level Perspective on Machine Learning in Astronomy

James R. A. Davenport



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

Number of ADS abstracts
including “machine learning”



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

The screenshot shows the scikit-learn Examples page. The top navigation bar includes Home, Installation, Documentation, Examples, and a Google Custom Search bar. The main content area is titled "Examples" and features a section for "Miscellaneous examples". Below this, there are several examples with small thumbnail images and titles:

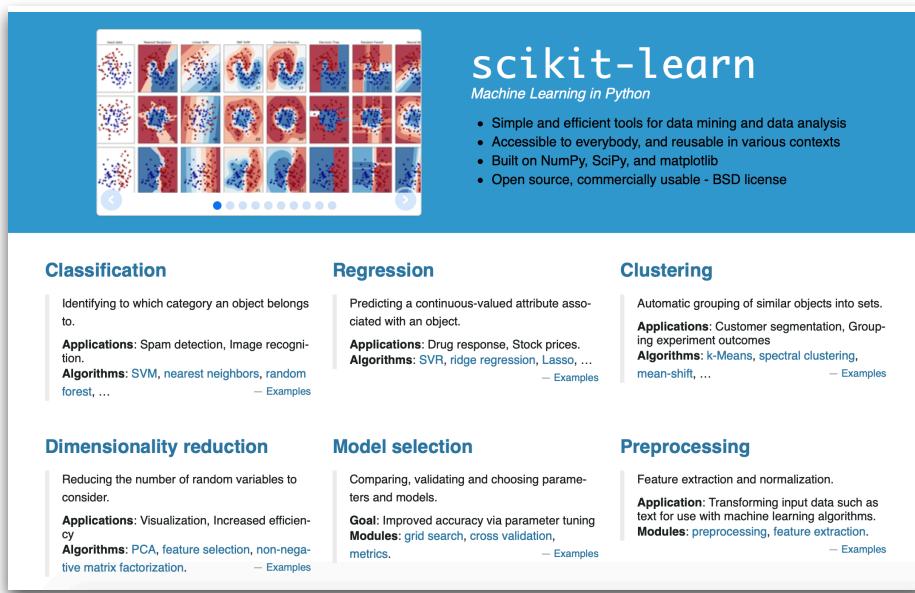
- Isotonic Regression
- Face completion with a multi-output estimators
- Multilabel classification
- Comparing anomaly detection algorithms for outlier detection on toy datasets
- Imputing missing values before building an estimator
- The Johnson-Lindenstrauss bound for embedding with random projections

The left sidebar lists various examples categorized under "Examples" and "Miscellaneous examples".

also astroML!
(see Brigitta's talk later)

The screenshot shows the homepage of the AstroML website. At the top, there is a navigation bar with links for Home, User Guide, Book Figures, Examples Plots, and a Google Custom Search bar. Below the navigation bar, there is a section titled "AstroML: Machine Learning and Data Mining for Astronomy" featuring several astronomical plots. To the left, there is a "News" sidebar with information about the textbook release in January 2014 and the software version 0.2 released in November 2013. There is also a "Links" sidebar with links to the mailing list and GitHub issue tracker. A "Videos" sidebar lists video talks from Scipy 2012 and 2013. A "Citing" sidebar provides instructions for citing the software. The main content area contains a detailed description of the AstroML module, its goals, and how to contribute. It also includes a "Downloads" section with links to the Python Package Index and GitHub.

Problems that ML is good for:



The screenshot shows the homepage of the scikit-learn website. At the top, there's a banner with a grid of small plots illustrating various machine learning models. Below the banner, the title "scikit-learn" is displayed in large white font, followed by "Machine Learning in Python" in smaller text. To the right of the title is a bulleted list of features:

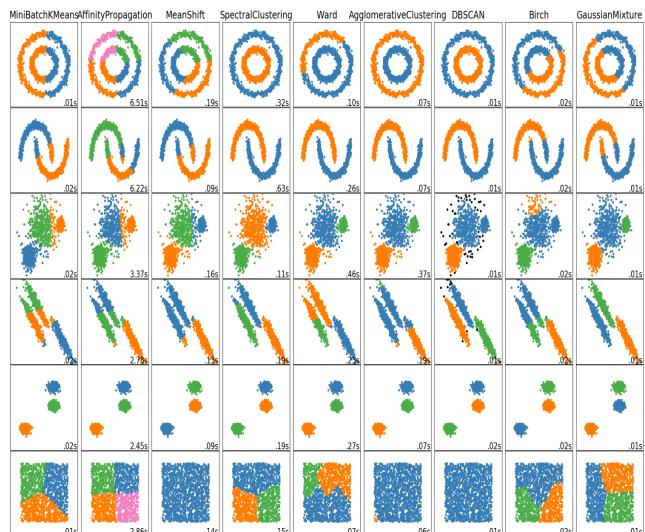
- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

The main content area is organized into several sections:

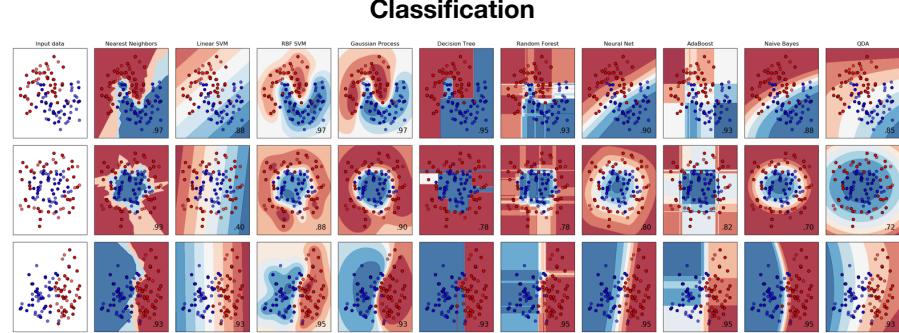
- Classification**: Describes identifying to which category an object belongs. Applications include spam detection and image recognition. Algorithms listed: SVM, nearest neighbors, random forest, etc.
- Regression**: Describes predicting a continuous-valued attribute associated with an object. Applications include drug response and stock prices. Algorithms listed: SVR, ridge regression, Lasso, etc.
- Clustering**: Describes automatic grouping of similar objects into sets. Applications include customer segmentation and grouping experiment outcomes. Algorithms listed: k-Means, spectral clustering, mean-shift, etc.
- Dimensionality reduction**: Describes reducing the number of random variables to consider. Applications include visualization and increased efficiency. Algorithms listed: PCA, feature selection, non-negative matrix factorization.
- Model selection**: Describes comparing, validating and choosing parameters and models. Goal: Improved accuracy via parameter tuning. Modules: grid search, cross validation, metrics.
- Preprocessing**: Describes feature extraction and normalization. Application: Transforming input data such as text for use with machine learning algorithms. Modules: preprocessing, feature extraction.

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

Clustering



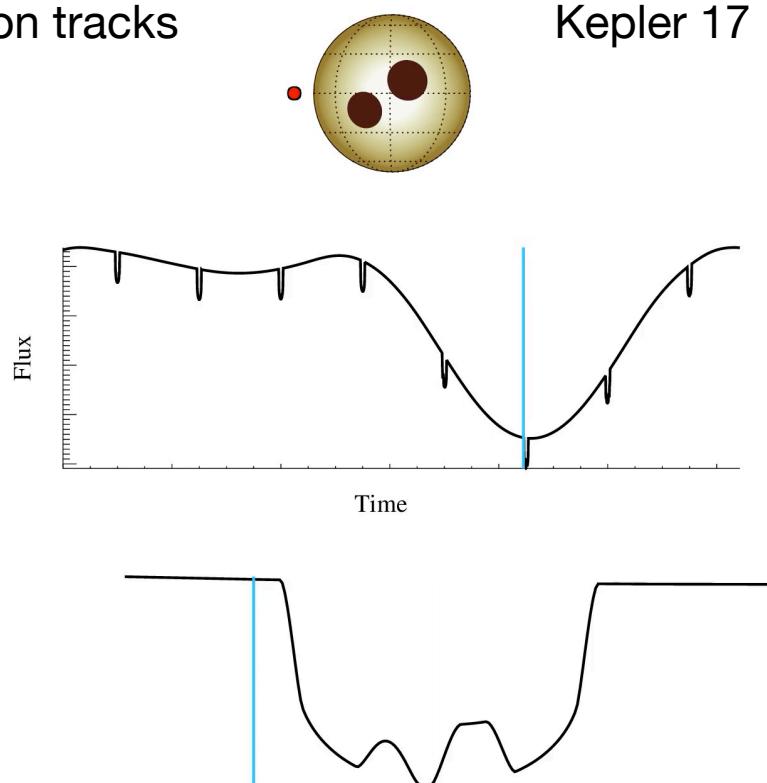
Classification



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

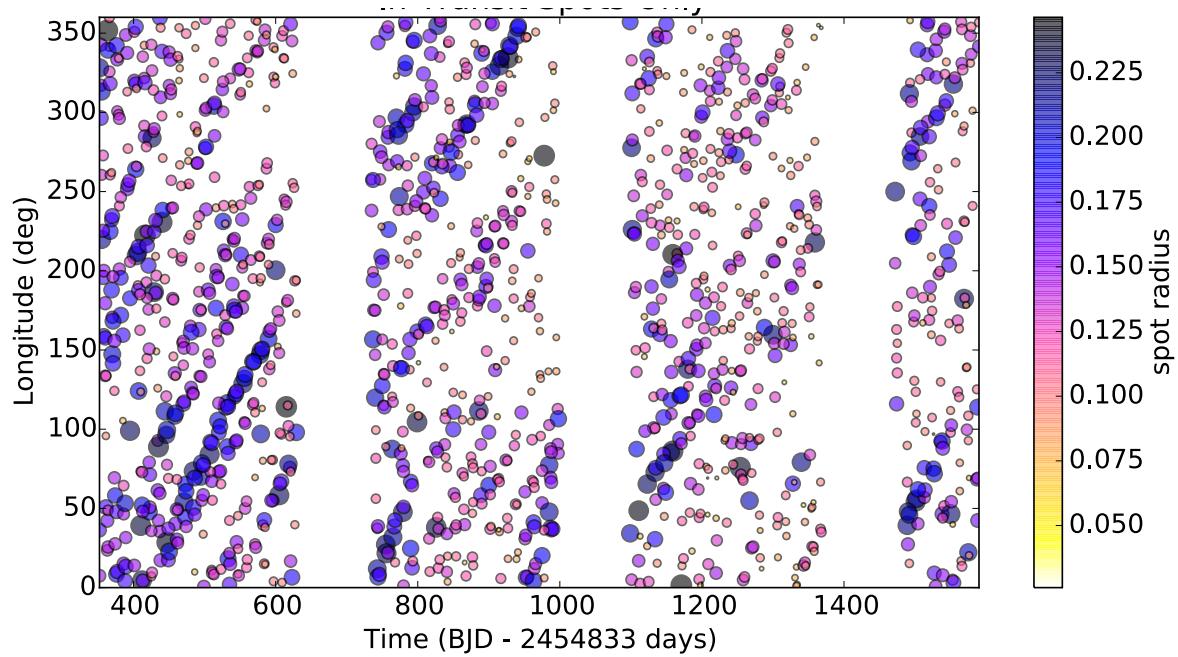
Example 1: Clustering starspot evolution tracks

Kepler 17



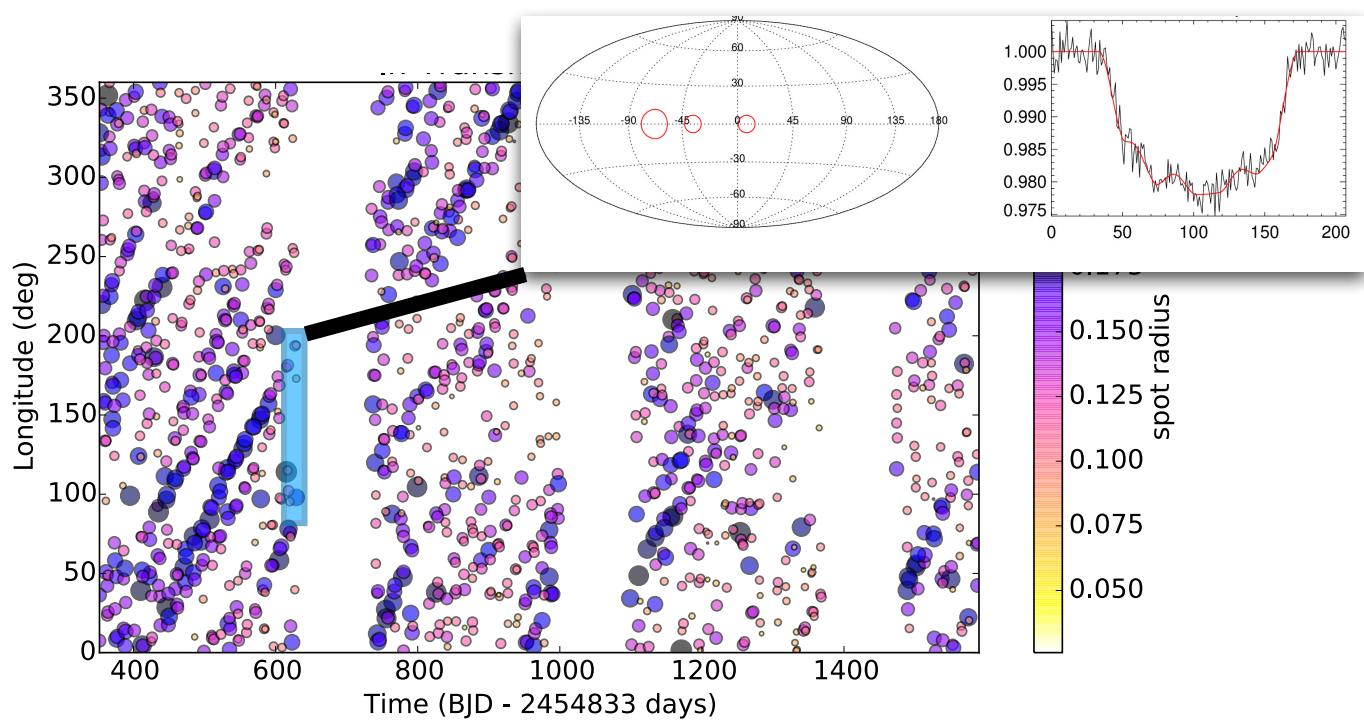
Example 1: Clustering starspot evolution tracks

Kepler 17



Example 1: Clustering starspot evolution tracks

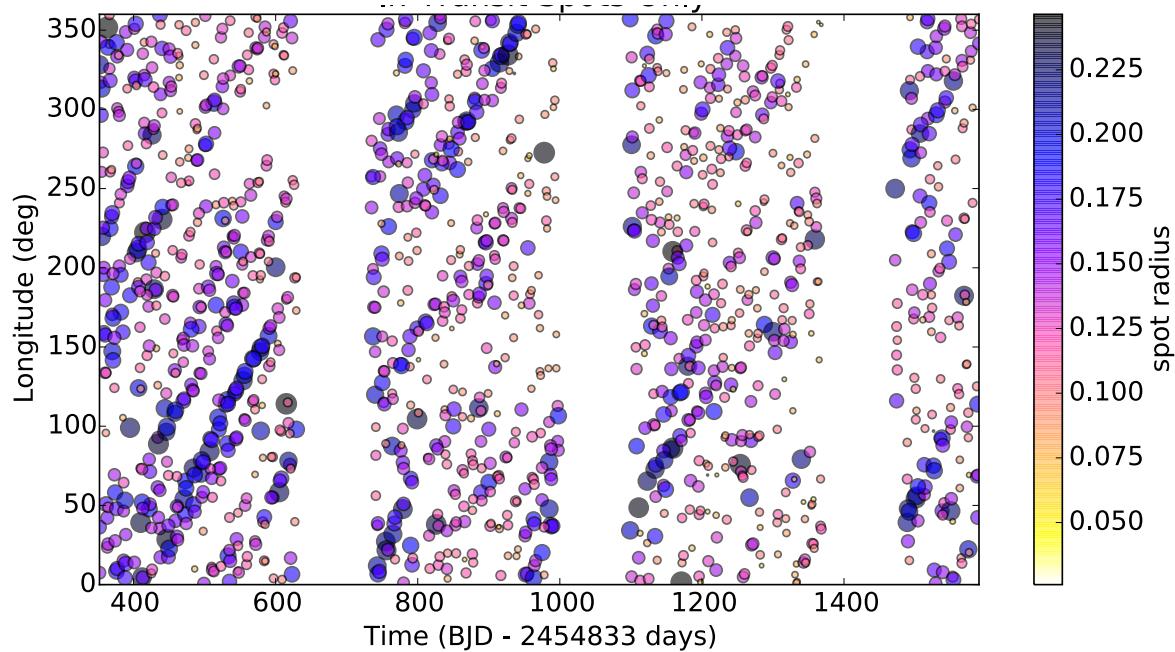
Kepler 17



Example 1: Clustering starspot evolution tracks

Kepler 17

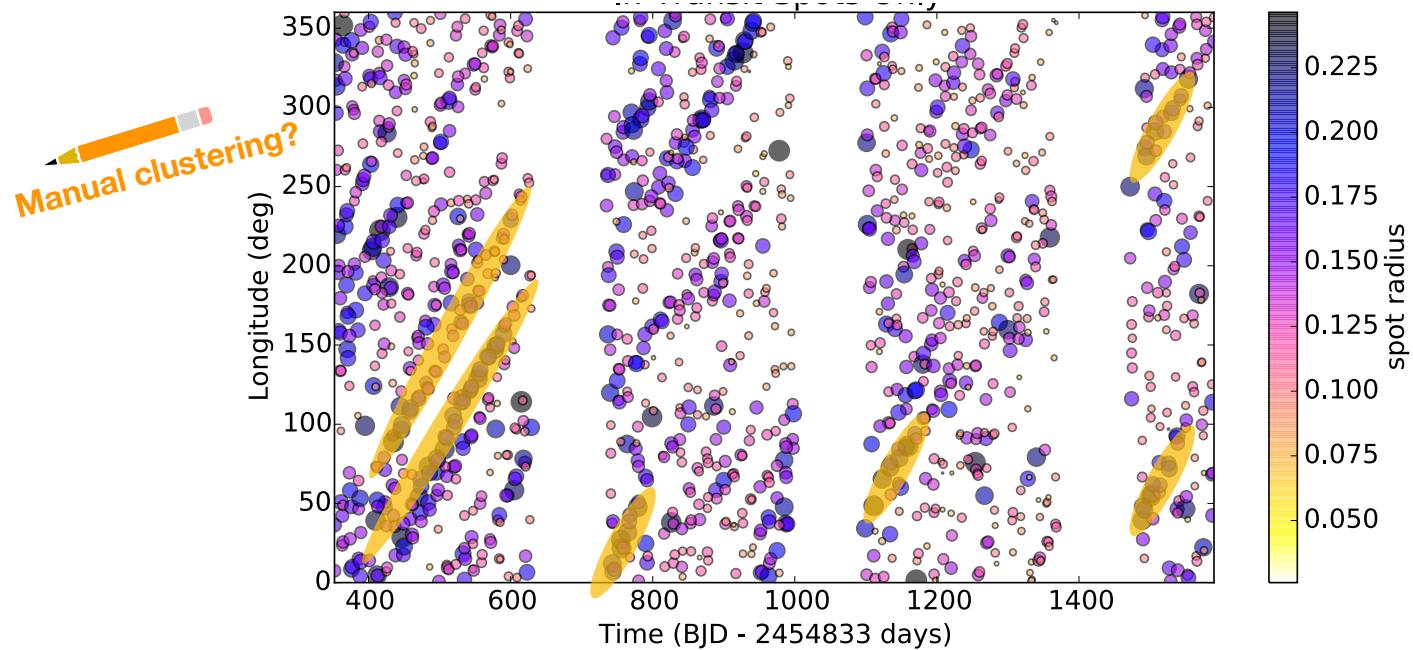
Question: Which tracks are starspots, how do they emerge/decay?



Example 1: Clustering starspot evolution tracks

Kepler 17

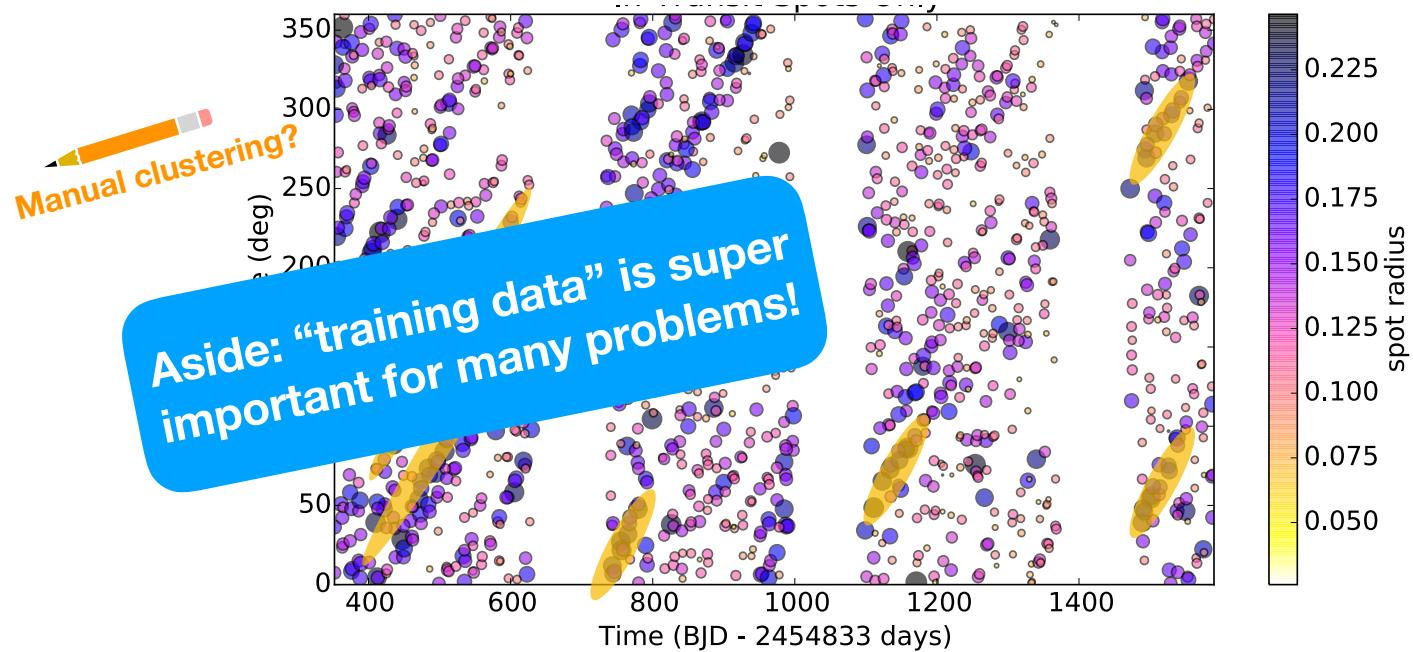
Question: Which tracks are starspots, how do they emerge/decay?



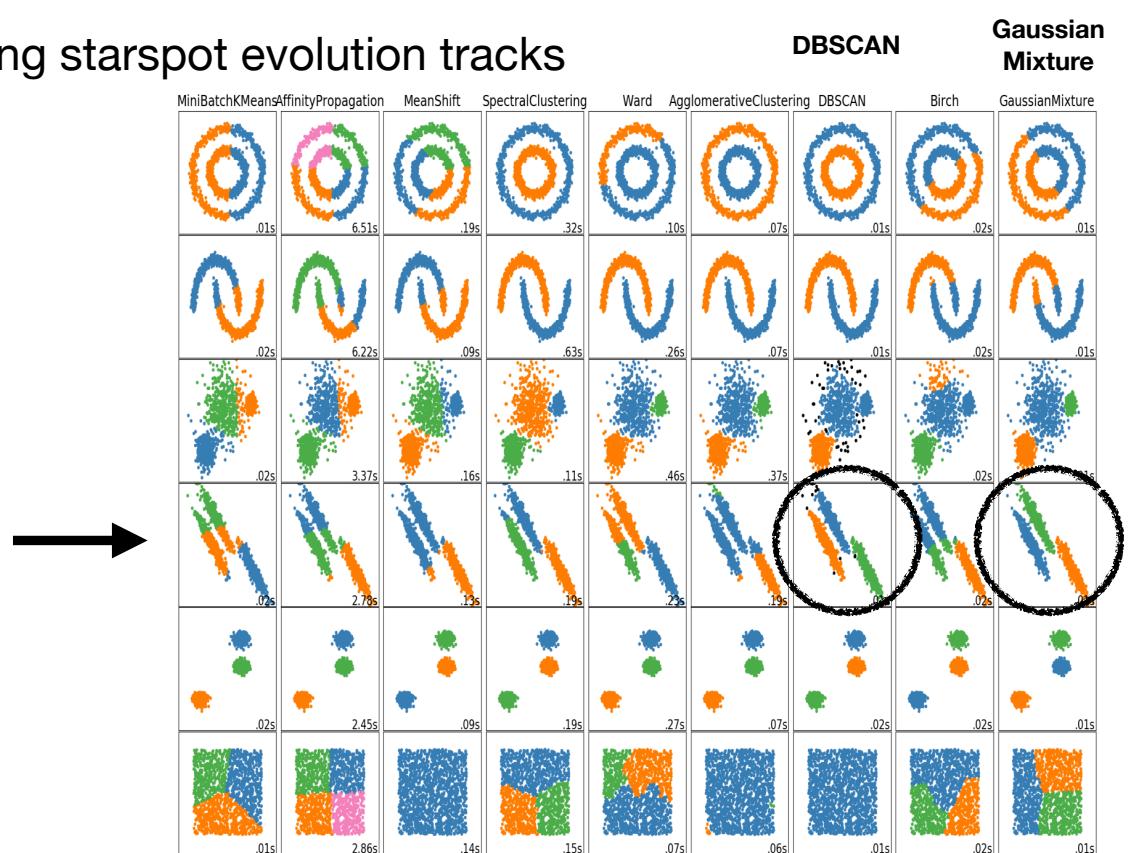
Example 1: Clustering starspot evolution tracks

Kepler 17

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

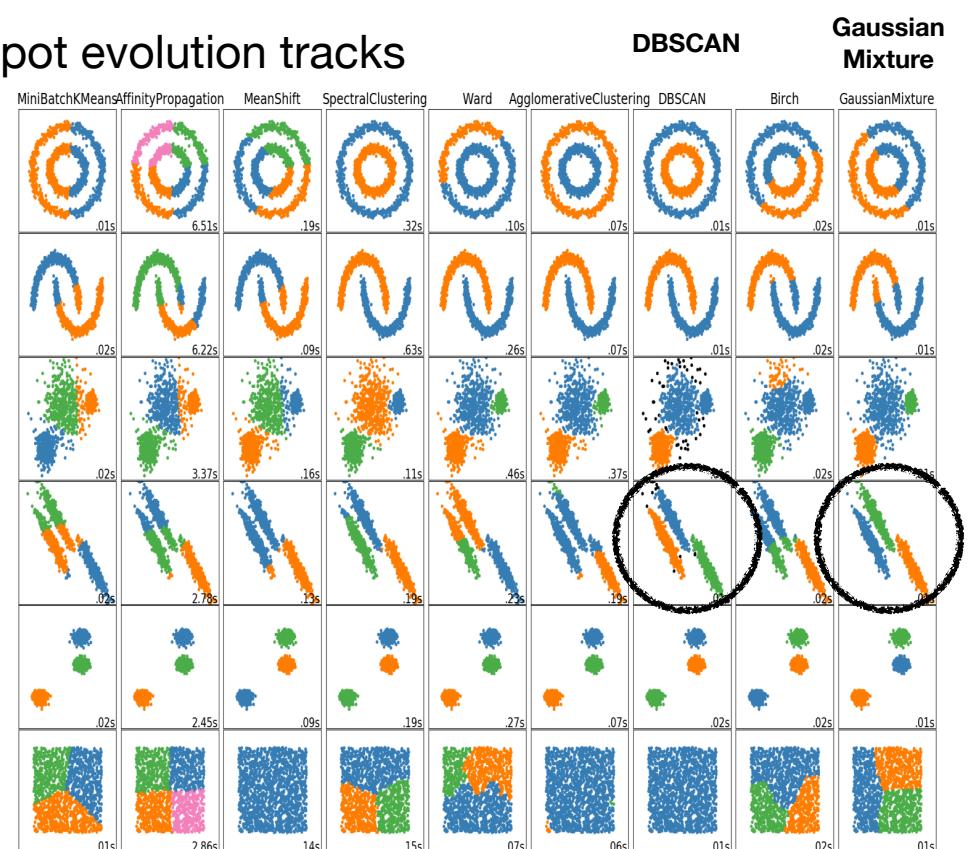


Example 1: Clustering starspot evolution tracks



Example 1: Clustering starspot evolution tracks

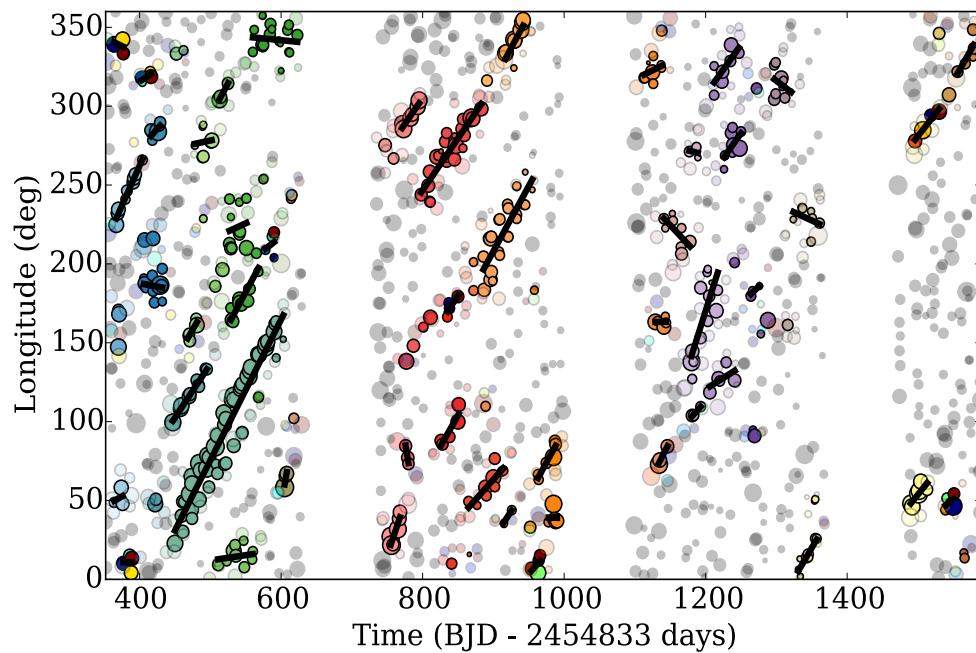
need to predefine Nclusters?



Example 1: Clustering starspot evolution tracks

Kepler 17

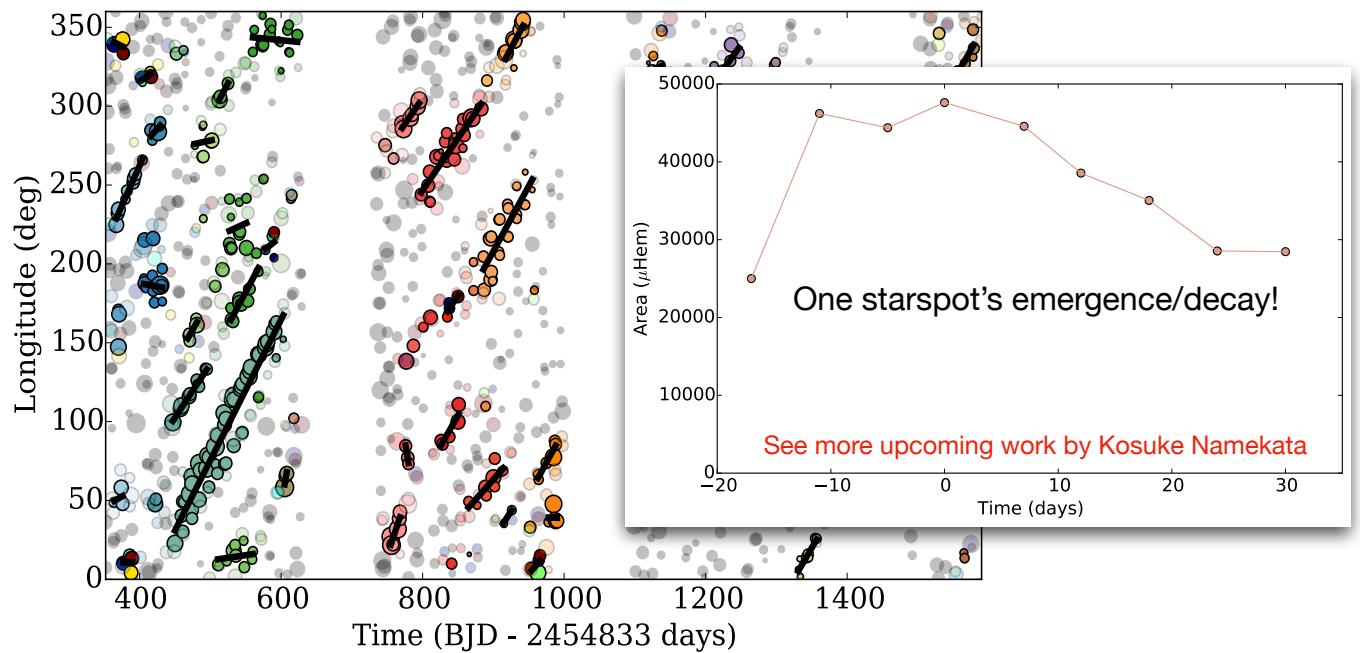
DBSCAN: Density-based spatial clustering of applications with noise



Example 1: Clustering starspot evolution tracks

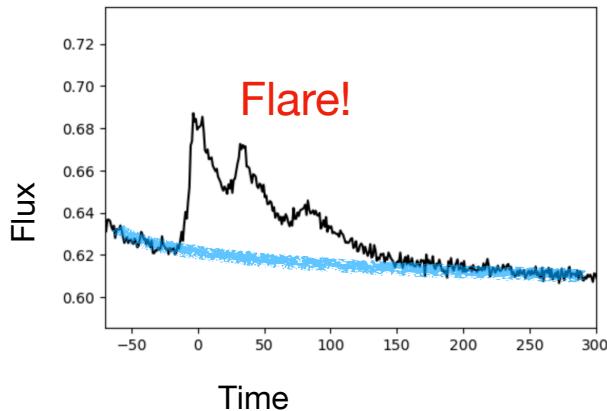
Kepler 17

DBSCAN: Density-based spatial clustering of applications with noise



Example 2: Modeling a complex stellar flare

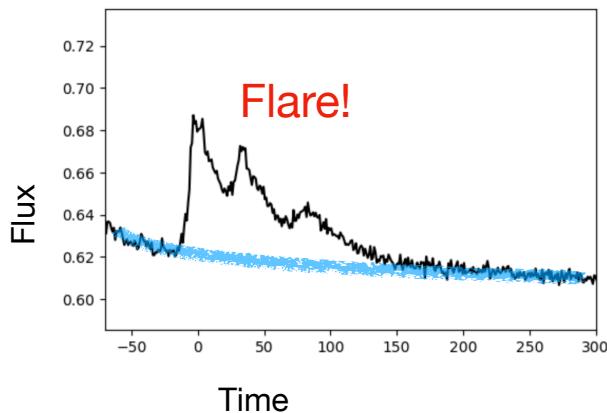
Question: Is there (quasi-) sinusoidal behavior in the flare decay?



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

Example 2: Modeling a complex stellar flare

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

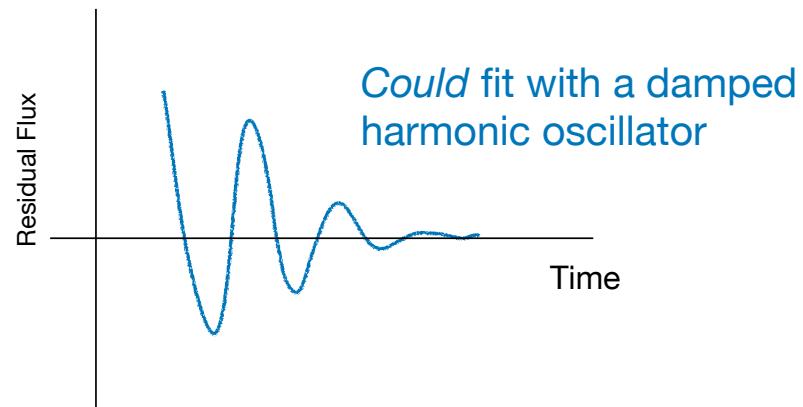
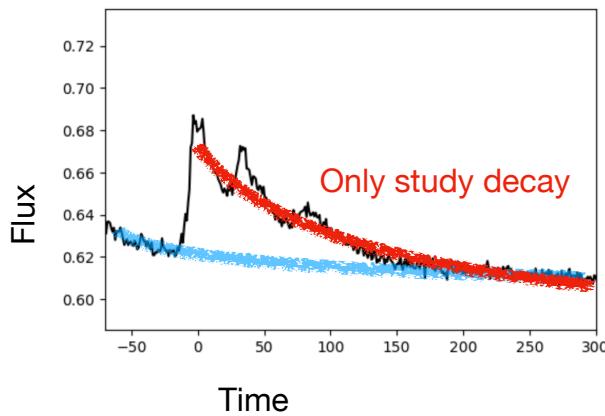


versus

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

Example 2: Modeling a complex stellar flare

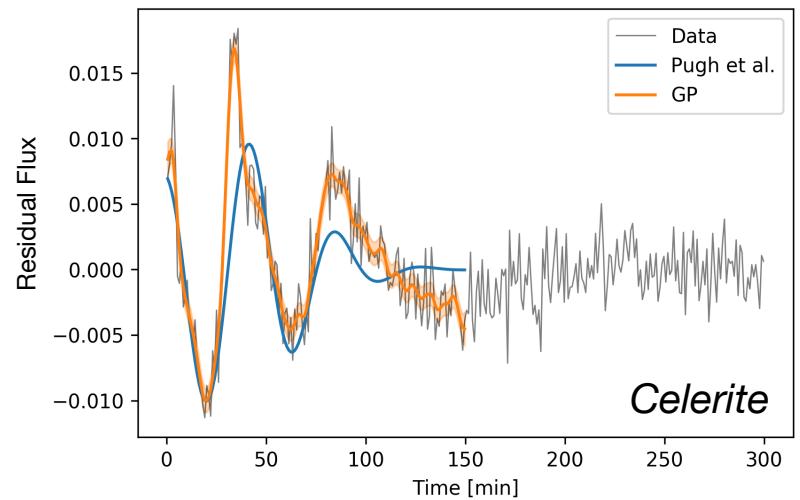
Question: Is there (quasi-) sinusoidal behavior in the flare decay?



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

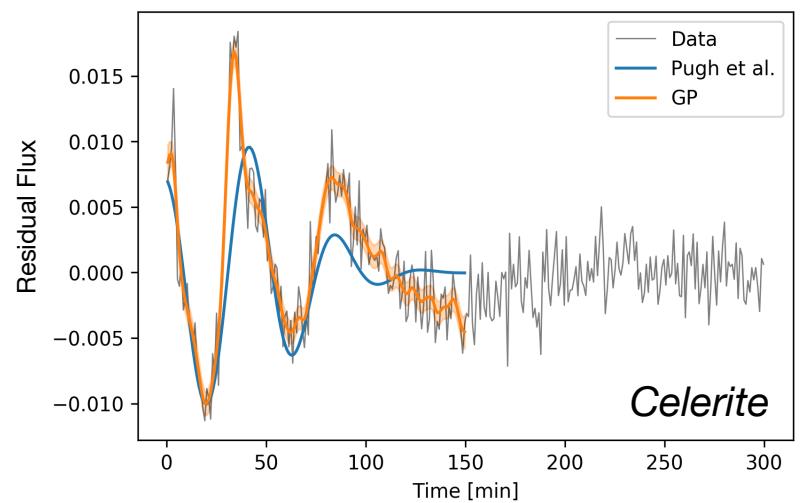
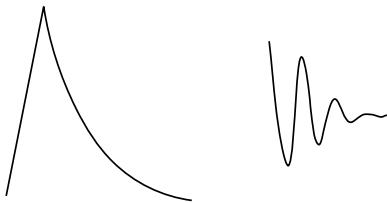


<https://github.com/RileyWClarke/QPP-GP>

Example 2: Modeling a complex stellar flare

Gaussian Process

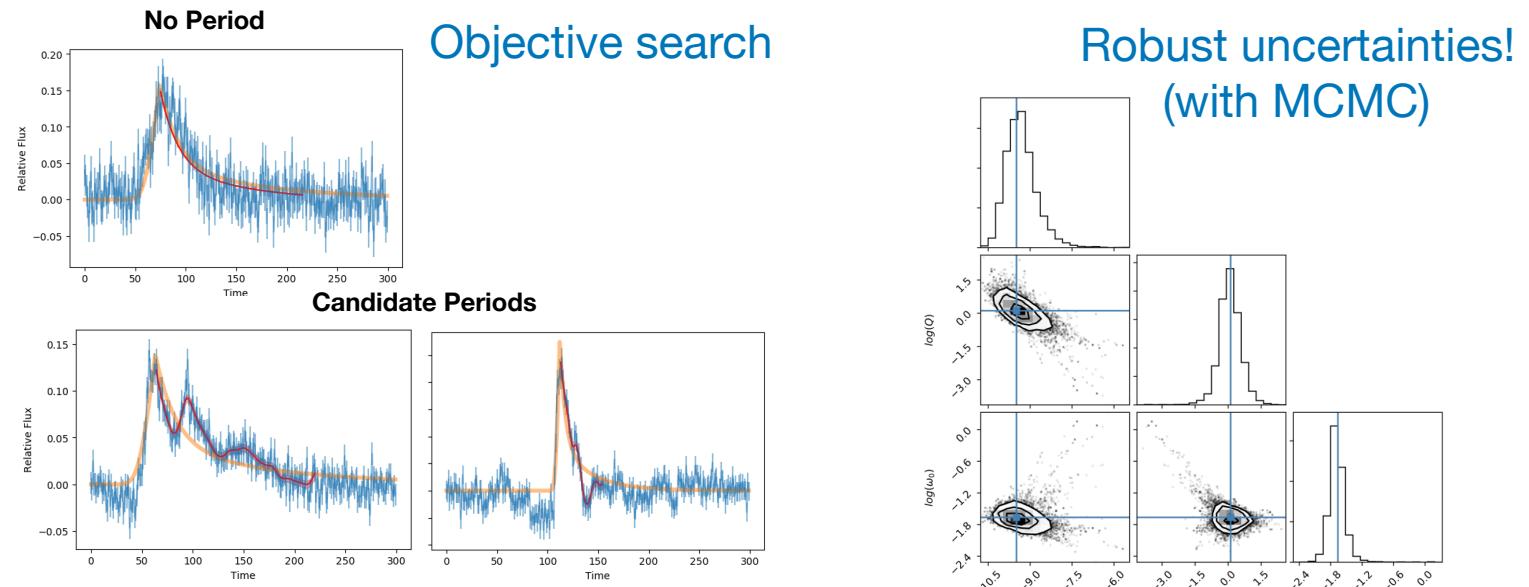
Use an exponential +
simple-harmonic-oscillator *kernel*



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

Example 2: Modeling a complex stellar flare

Gaussian Process

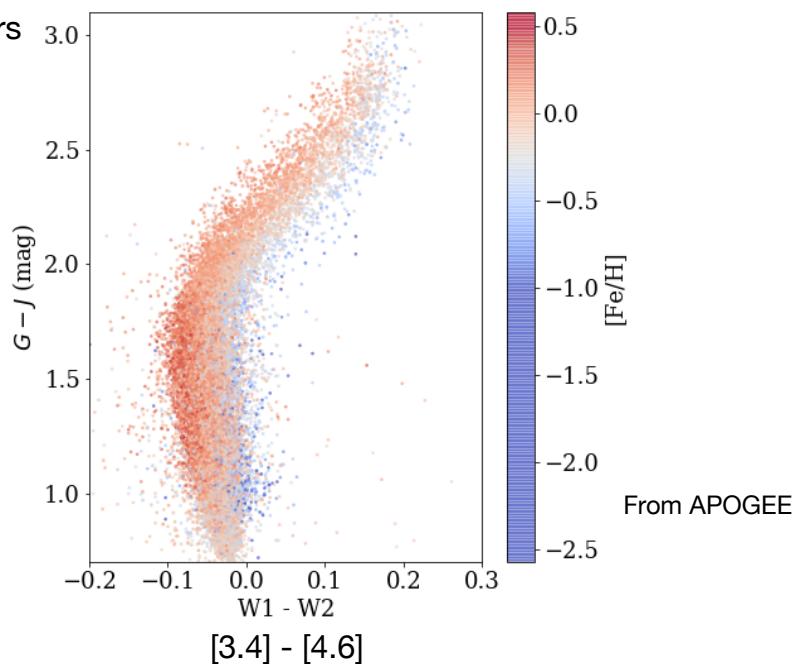


<https://github.com/RileyWClarke/QPP-GP>

Example 3: Modeling photometric metallicities

Observation: [Fe/H] gradient in stars

Gaia DR2 + (WISE + 2MASS) + APOGEE



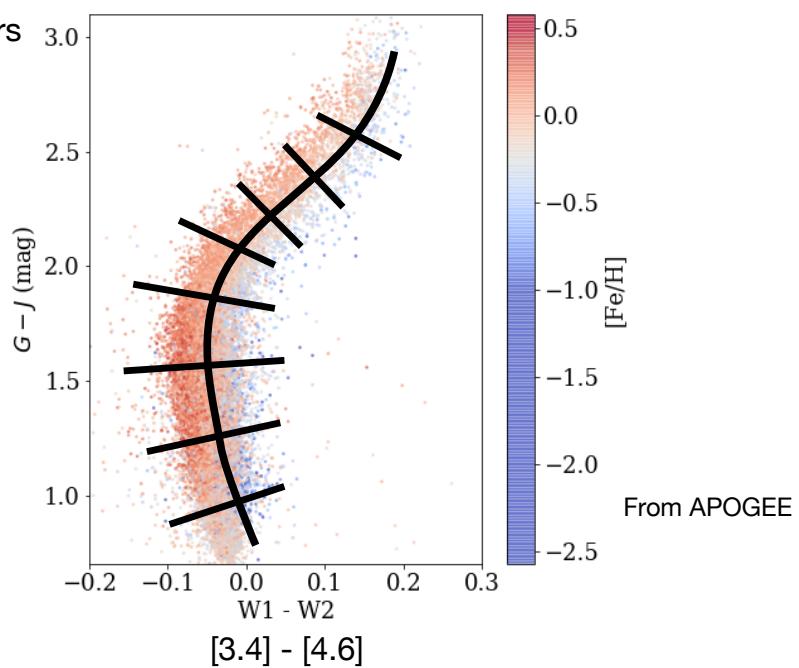
<https://github.com/jradavenport/ingot/>

jradavenport

Example 3: Modeling photometric metallicities

Observation: [Fe/H] gradient in stars

We could build a complex polynomial or spline model



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

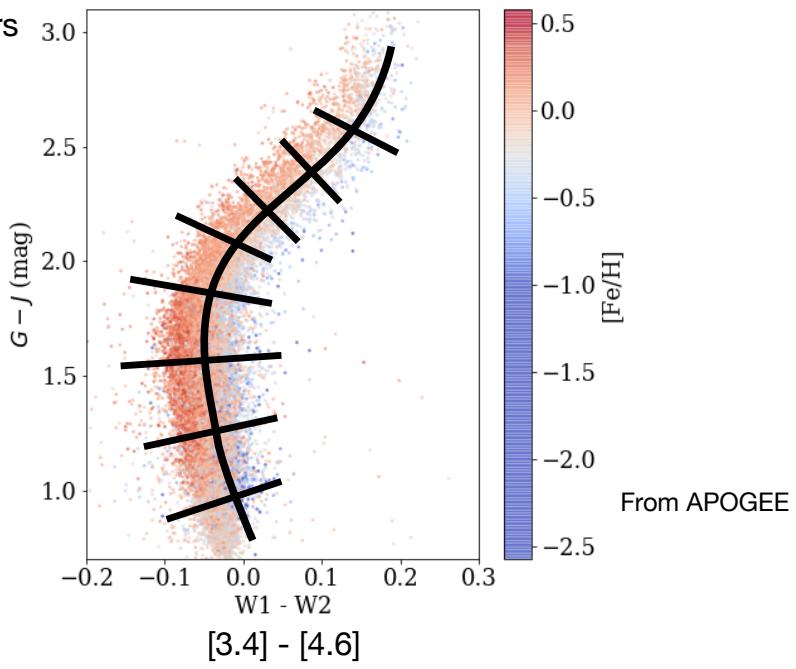
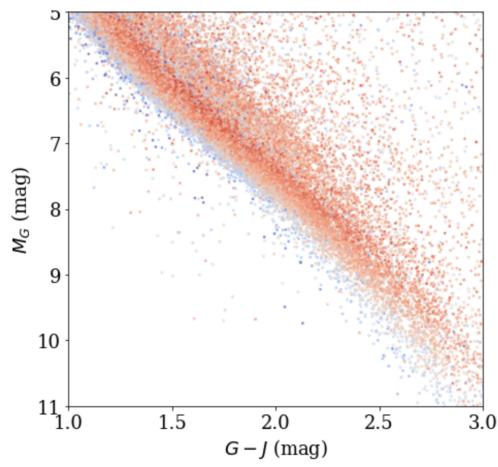
jradavenport

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!



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

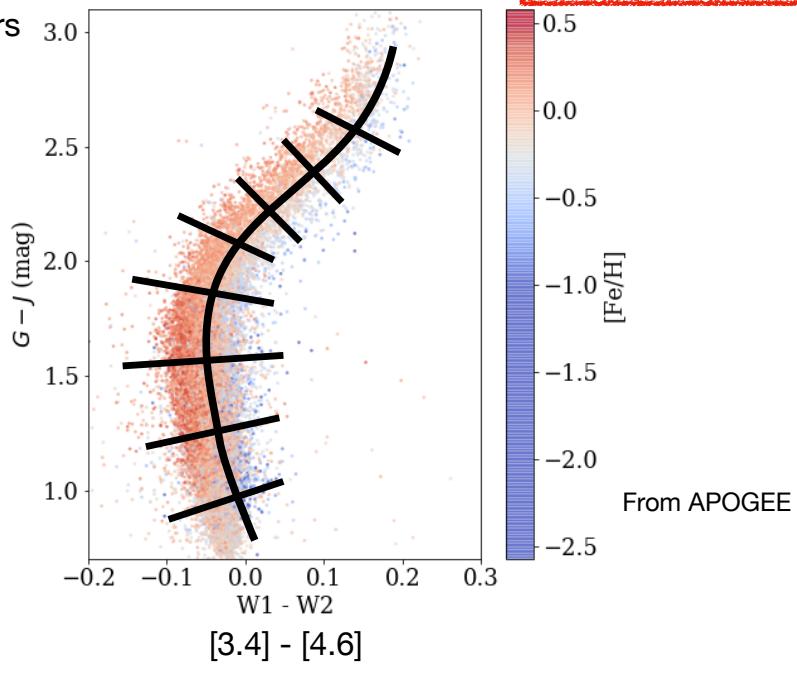
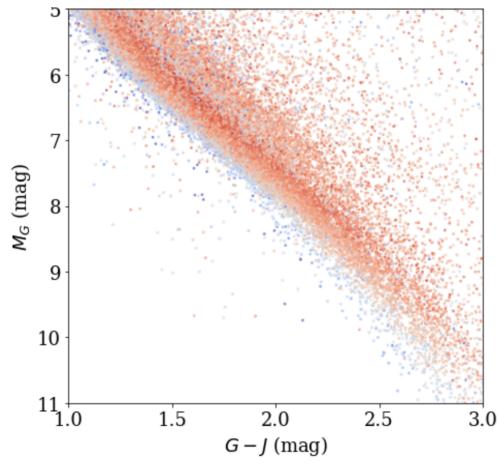
jradavenport

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!

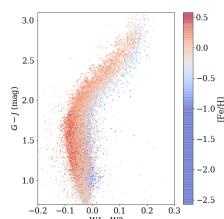


Or use a simple, flexible ML model!

Example 3: Modeling photometric metallicities

KNearestNeighbors

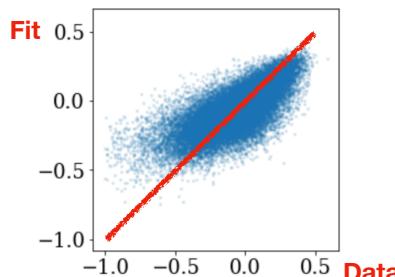
Xdata = ($G-J$, $W1-W2$)
Ydata = [Fe/H]



```
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)
```



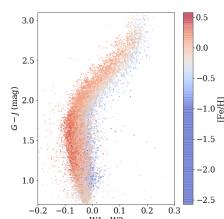
<https://github.com/jradavenport/ingot/>

jradavenport

Example 3: Modeling photometric metallicities

KNearestNeighbors

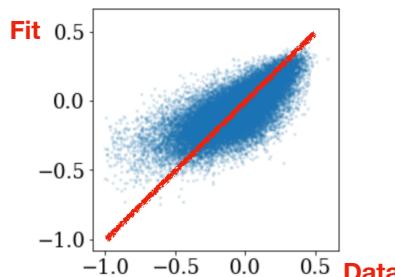
Xdata = ($G-J$, $W1-W2$)
Ydata = [Fe/H]



```
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)
```



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

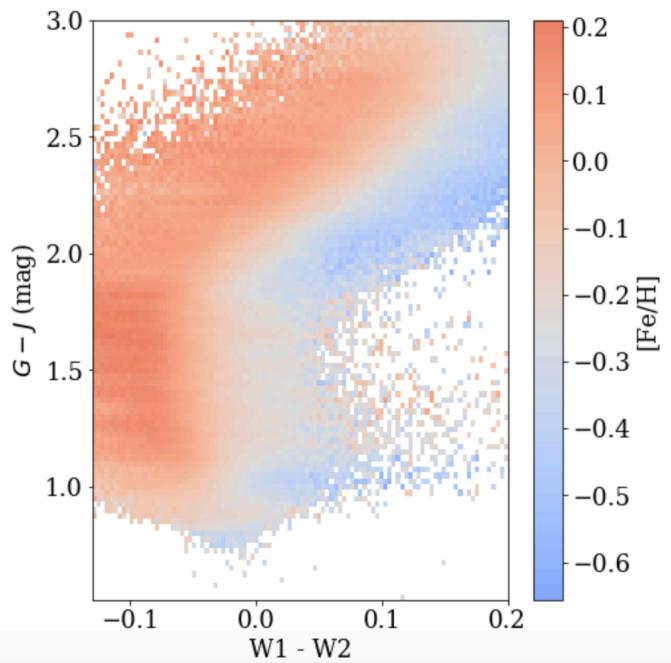
jradavenport

Example 3: Modeling photometric metallicities

KNearestNeighbors

**Result: a simple to use “surface”,
no tweaking for shape/order,
extend to additional dimensions easily**

1 Million new stars with no spectra



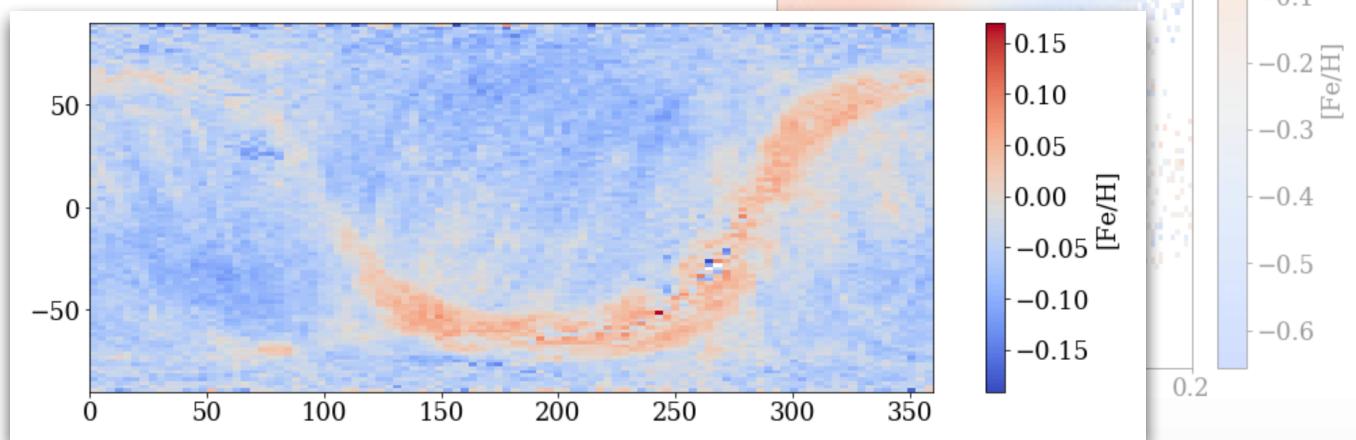
<https://github.com/jradavenport/ingot/>

jradavenport

Example 3: Modeling photometric metallicities

KNearestNeighbors

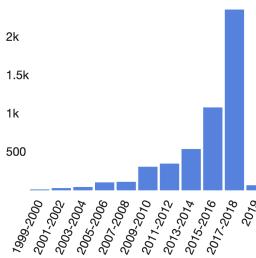
Result: a simple to use “surface”,
no tweaking for shape/order,
extend to additional dimensions easily



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

jradavenport

Conclusions



ML is easier and more “boring” than ever!

