Making Faces

Dimension Reduction and Image Recognition

Steve Crow

Goals

Disclaimers

1

List of commonly mispronounced Python packages self.Python

submitted 3 years ago by **michael_david** | dunderinit

What are some commonly mispronounced python packages or closely related libraries other than the below:

- pypi = "pie-pee-eye" or "pip-ee"
- uwsgi = "you-wiz-gee"
- gunicorn = "jee-unicorn"
- nginx = "engine-ex"
- numpy = "num-pie" (debatable)

30 comments share save hide give award report crosspost

Law of 2 Feet

If a session just isn’t doing it for you, don’t feel obligated to stay. The speakers won’t mind, and you should do what you can to get the most out of CodeMash. Simply head for the door and find another session to go to.

Machine Learning

A method of data analysis that automates model building.

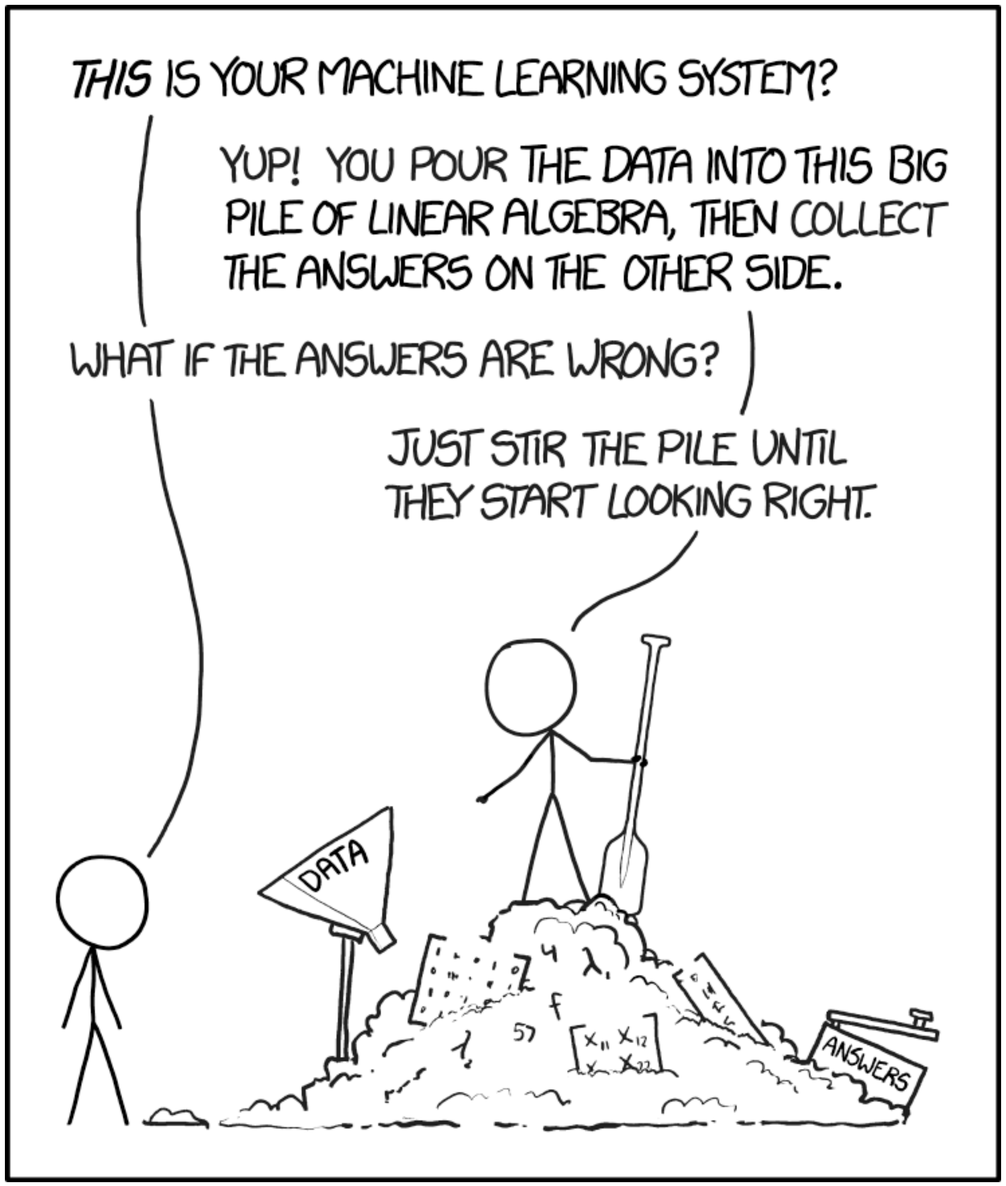
Steve Crow
@cr0wst

Training Set

A set of example data used for learning.

(Semi) Supervised Learning

Uses a combination of input and output data in the **training set** to build a mathematical model.



xkcd.com/1838/

Steve Crow
@cr0wst

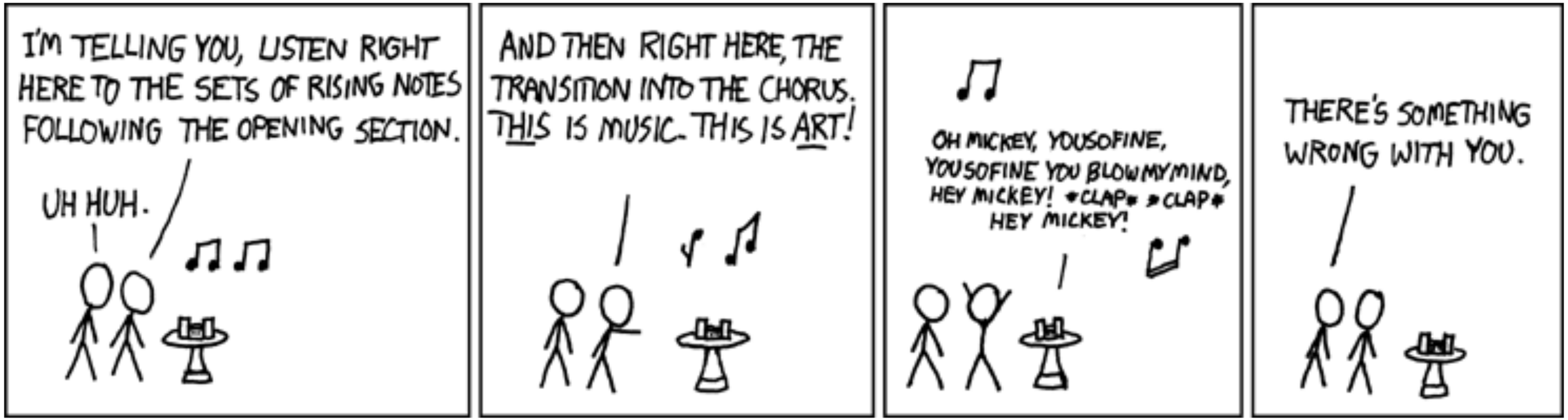
Unsupervised Learning

Uses only inputs in the dataset and finds structures or grouping in the data.

Your friend is having a birthday!


WHAT KIND OF MUSIC DO YOU LISTEN TO?
OH, A MIX OF THINGS. SOME CLASSIC ROCK LIKE BOSTON, BUT THEN OF COURSE QUEEN AND BOWIE, JOAN JETT...
DEFINITELY, WE NEED MORE OF THOSE SOUNDS.
BUT THERE'S SOME GREAT NEWER STUFF TOO, LIKE FRANZ FERDINAND, THE DONNAS, AND AUDIOSLAVE.
SOMETIMES THEY'RE A LITTLE MUCH FOR ME. I GO MORE FOR THINGS LIKE THE ARCADE FIRE, SOMETIMES MIXING SOME ELECTRONIC SOUNDS LIKE POSTAL SERVICE.
OH YEAH--HAVE YOU EVER CHECKED OUT FREEZEPOP?
MHM! SYNTH POP CAN BE FUN, BUT AT THE SAME TIME, I AGREE THAT SOMETIMES YOU JUST NEED TO BLAST SOME METALLICA.
WHO?
... METALLICA.
ARE THEY NEW?
I SOUND PRETTY KNOWLEDGABLE ABOUT MUSIC UNTIL PEOPLE FIGURE OUT THAT I'M JUST NAMING BANDS FROM GUITAR HERO.

How do we know which songs to pick?


I'M TELLING YOU, LISTEN RIGHT HERE TO THE SETS OF RISING NOTES FOLLOWING THE OPENING SECTION.
UH HUH.
AND THEN RIGHT HERE, THE TRANSITION INTO THE CHORUS. THIS IS MUSIC. THIS IS ART!
OH MICKEY, YOUSOFINE, YOUSOFINE YOU BLOW MY MIND, HEY MICKEY! *CLAP* *CLAP* HEY MICKEY!
THERE'S SOMETHING WRONG WITH YOU.

Musical Features

acousticness, danceability, energy, instrumentalness, liveness, loudness, speachiness, valence, tempo, time signature, key, duration…

Which features should we use?

Principal Component Analysis

A process for compressing, extracting features, and reducing dimensions.

Steve Crow
@cr0wst

Feature Elimination

Reduce the feature space by eliminating features.

Feature Extraction

Create new features as a combination of existing features in an attempt to reduce any sort of redundancy.

When to use PCA?

Use **Principal Component Analysis** when you want to make sure your variables are independent of one another.

When to use PCA?

Use **Principal Component Analysis** when you are okay with making your independent variables less interpretable.

Let’s try it out!

Song Features

Spotify has an API that allows us to access song features. The features that we are going to isolate are **energy** and **loudness**.

Step 1: Training Set

Python Data Analysis Library

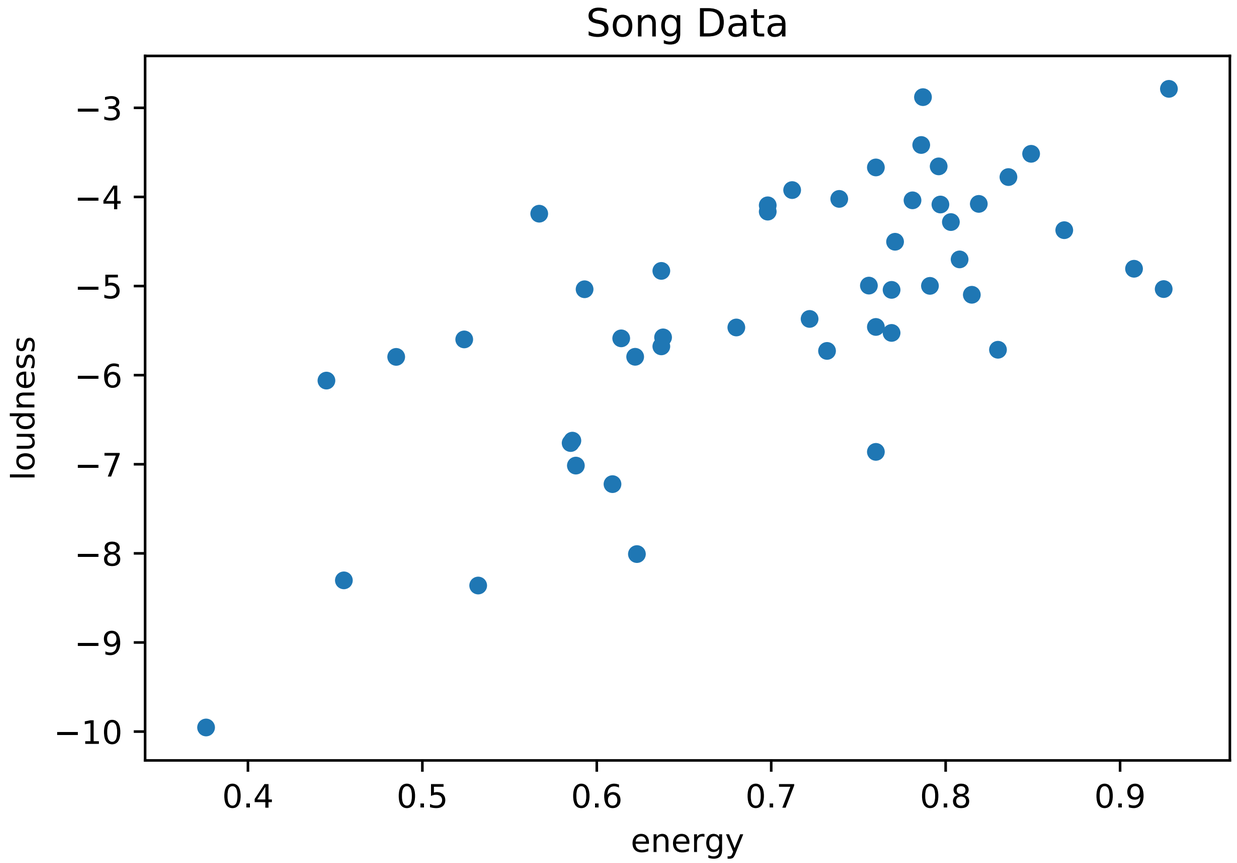
```
import pandas as pd

songs = pd.read_csv('songdata.csv')
```

Read CSV Into a DataFrame

Step 1: Training Set

name	danceability	energy	key	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo
Love The Way You Lie	0.749	0.925	10	-5.034	0.2270	0.24100	0.000000	0.5200	0.641	86.989
Hymn For The Weekend - Seeb Remi	0.565	0.849	0	-3.516	0.0517	0.00868	0.000005	0.1200	0.427	102.034
Uptown Funk	0.856	0.609	0	-7.223	0.0824	0.00801	0.000081	0.0344	0.928	114.988
We Can't Stop	0.613	0.622	1	-5.794	0.0334	0.00882	0.000000	0.3700	0.484	80.003
This Is What You Came For	0.630	0.928	9	-2.787	0.0331	0.19900	0.124000	0.1480	0.465	123.963

Step 1: Training Set



Normalization

Feature scaling, or mean normalization.

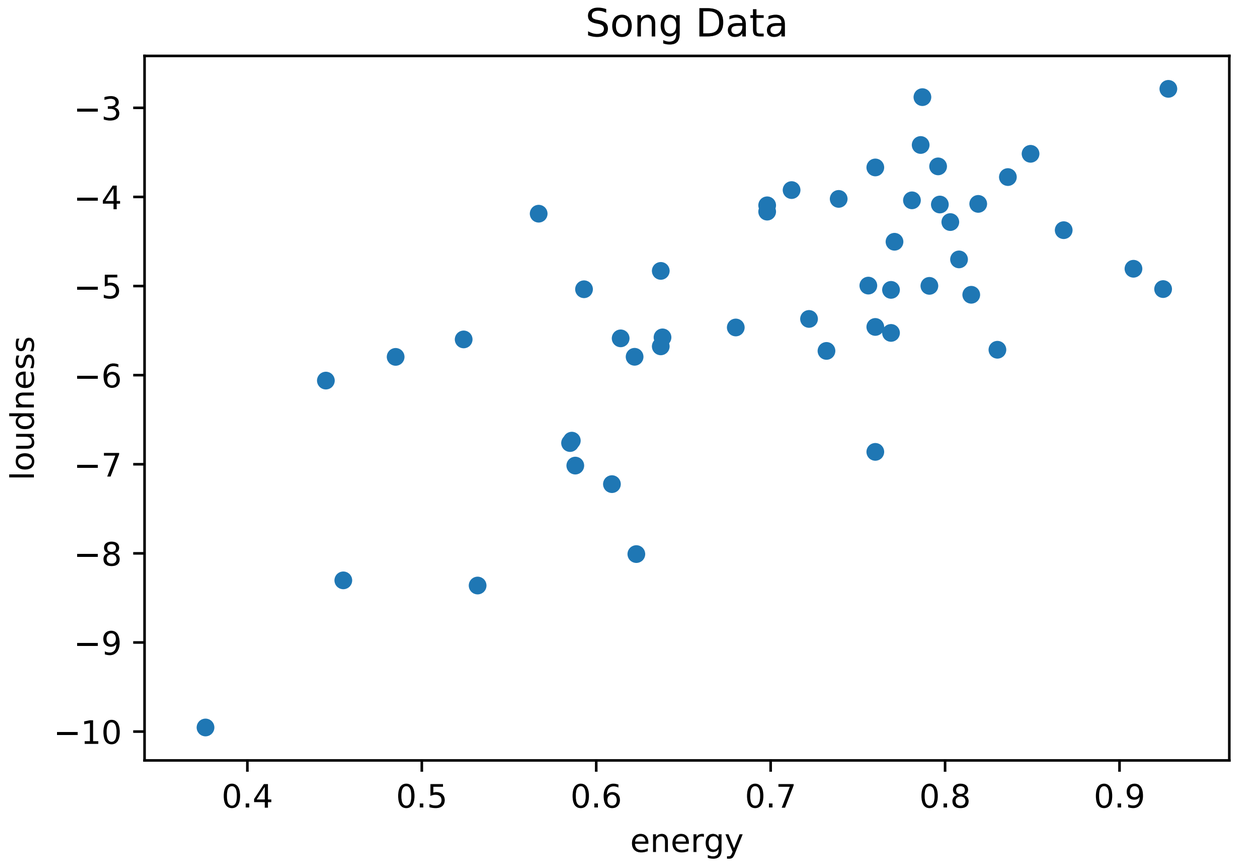
Step 2: Normalize

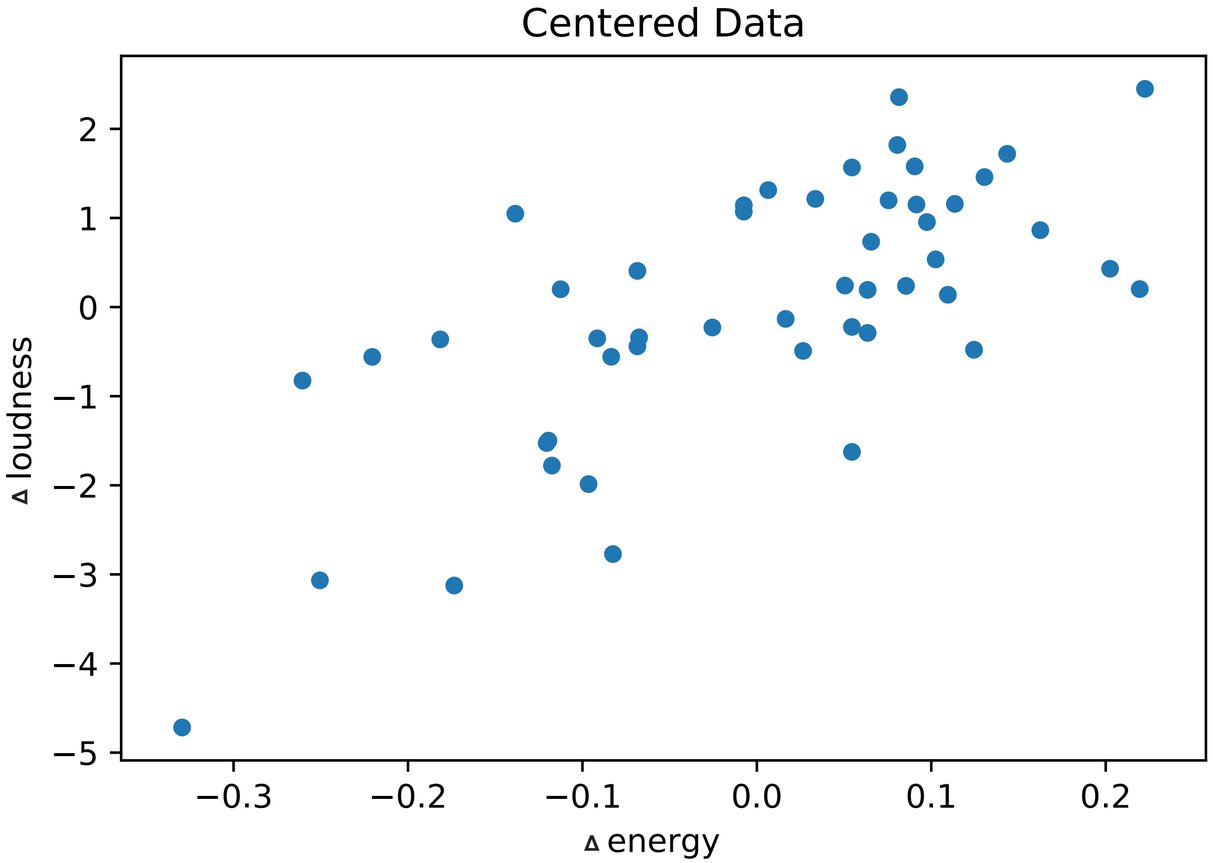
Average of Each Feature

```
songs = songs.sub(songs.mean())
```

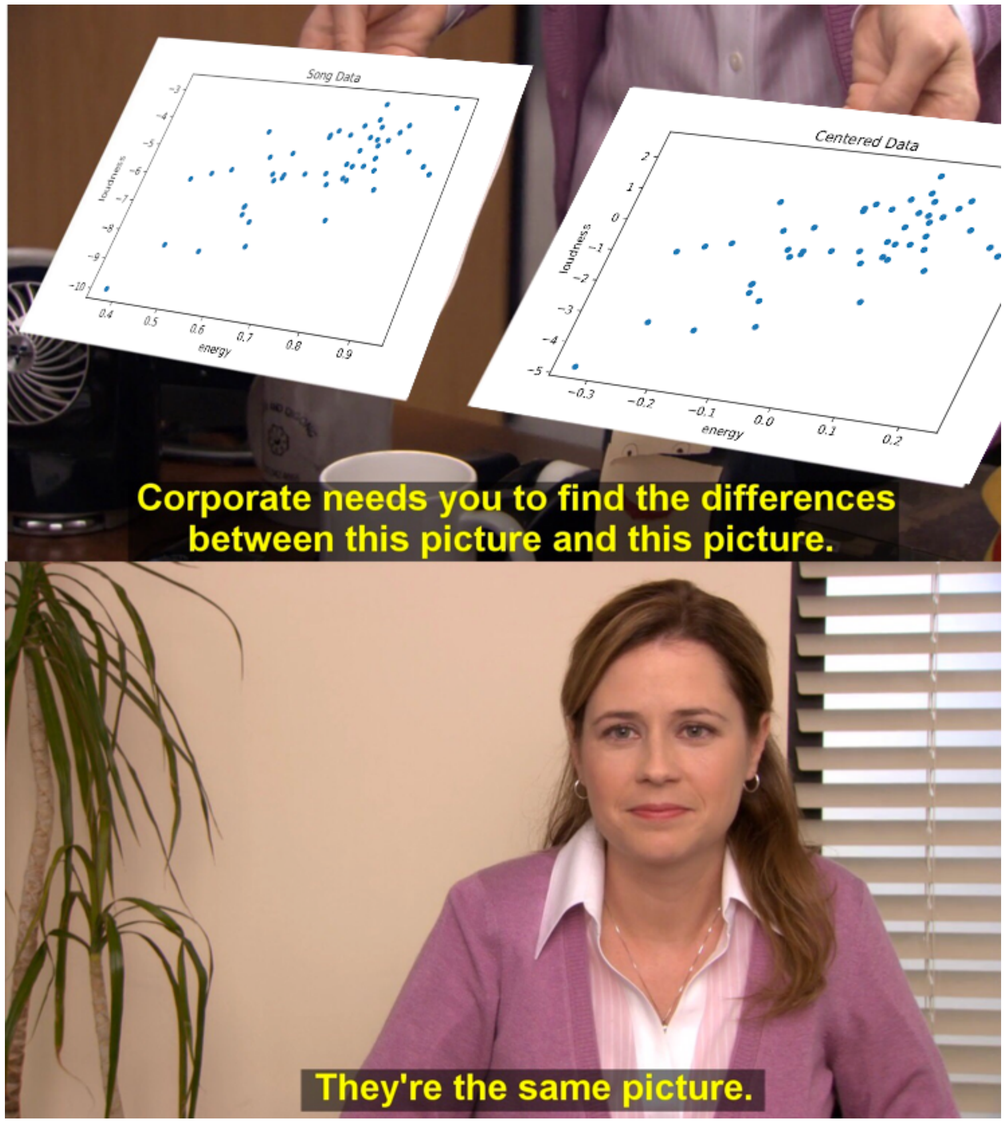
Perform Element Subtraction

Step 2: Normalize

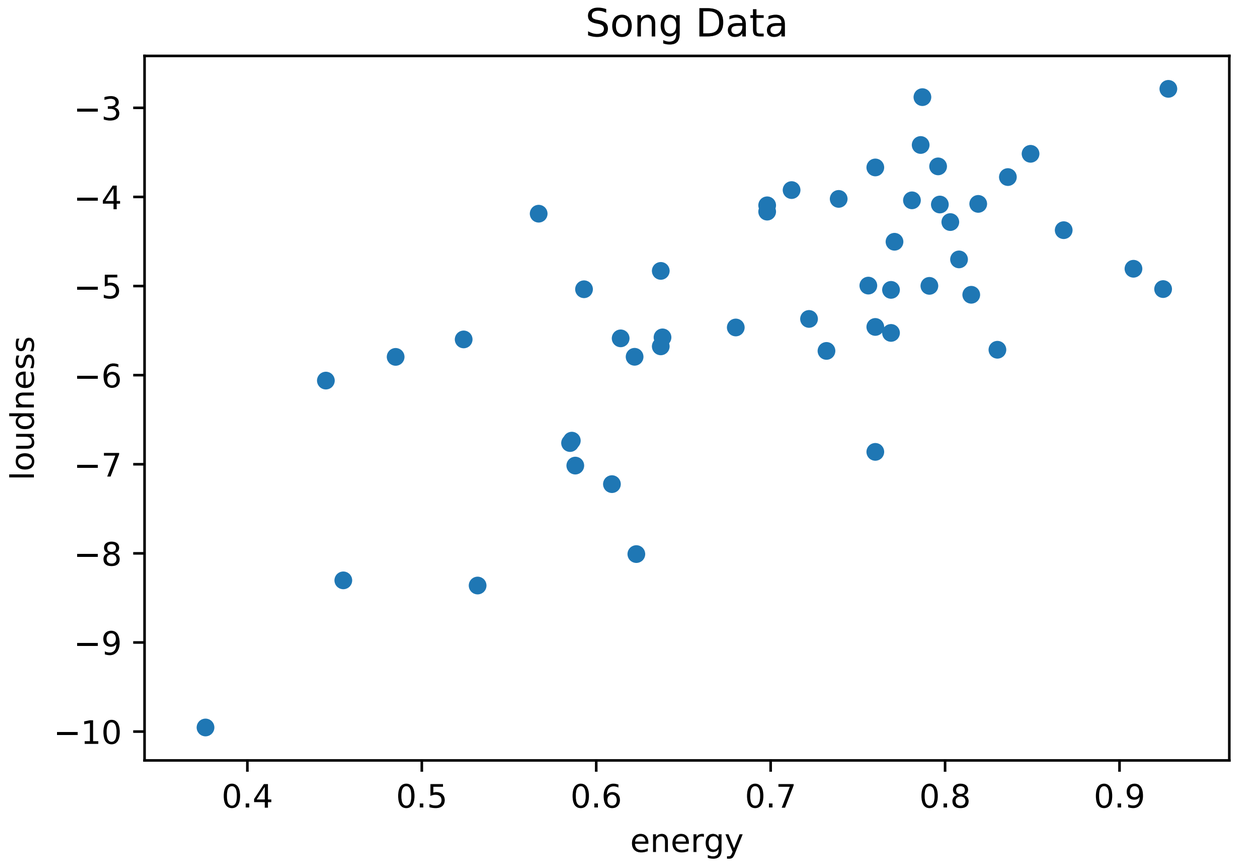


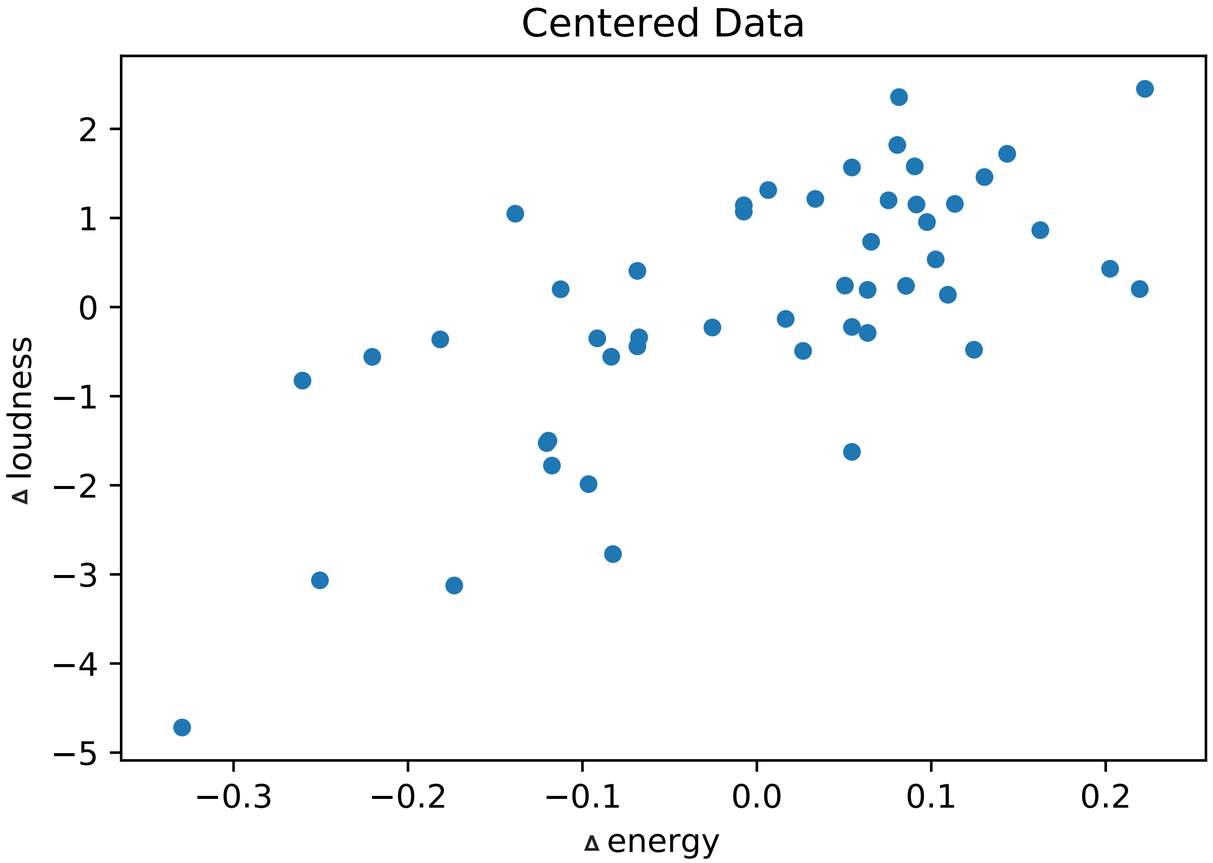


Steve Crow
@cr0wst


Song Data
loudness
energy
Centered Data
loudness
energy
Corporate needs you to find the differences between this picture and this picture.
They're the same picture.

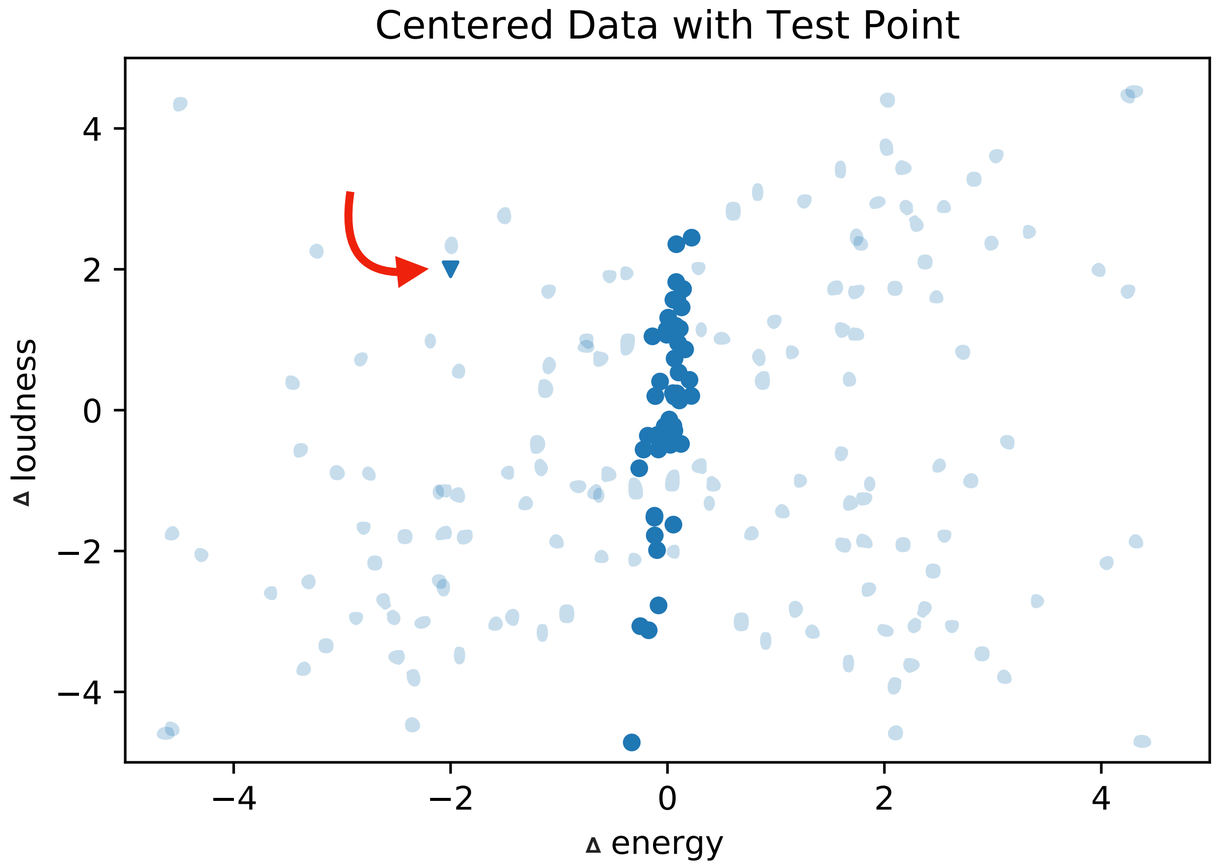
Step 2: Normalize



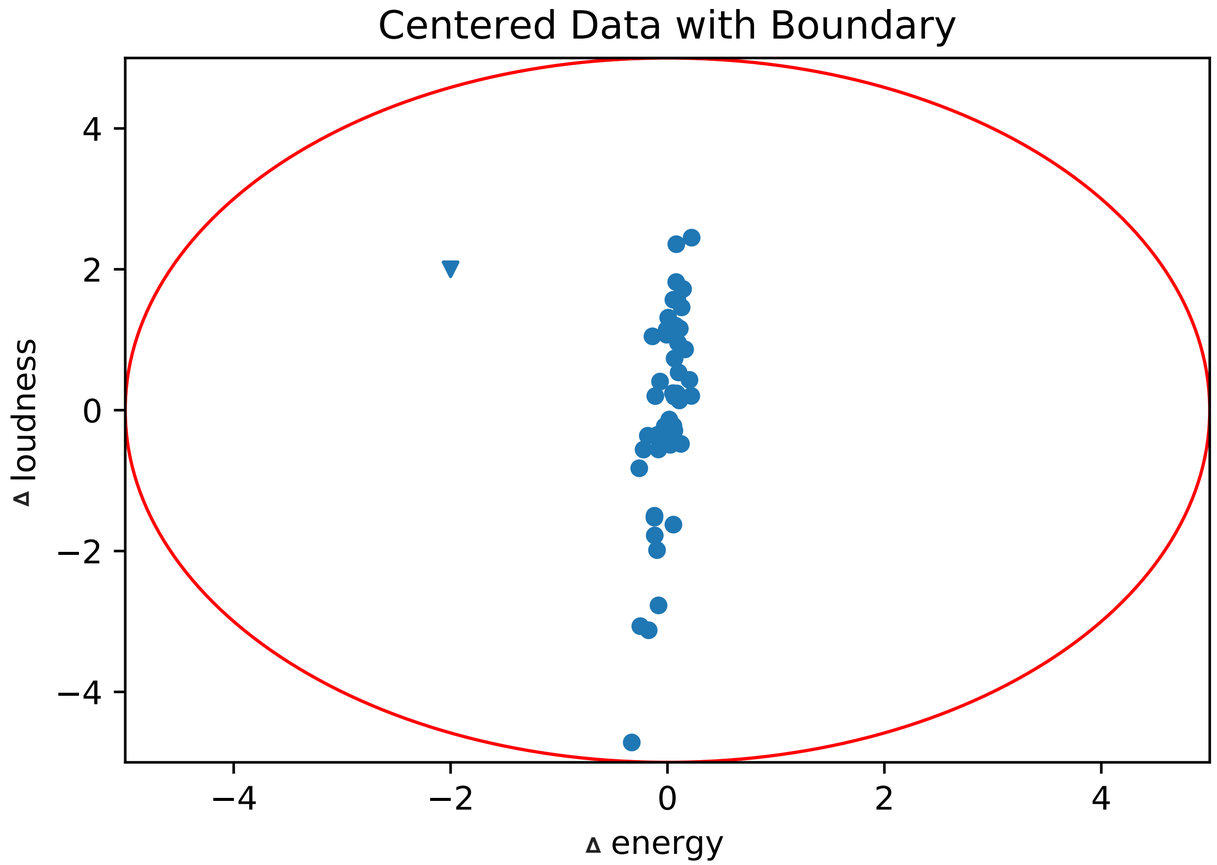


Steve Crow
@cr0wst

Add this to our playlist?



Create a Boundary



Does this song belong?

```
def is_playlistable(energy, loudness, radius):
    return math.sqrt(energy**2 + loudness**2) <= radius
```

$$\sqrt{energy^2 + loudness^2} \leq radius$$

Steve Crow
@cr0wst

Is this song playlistable?

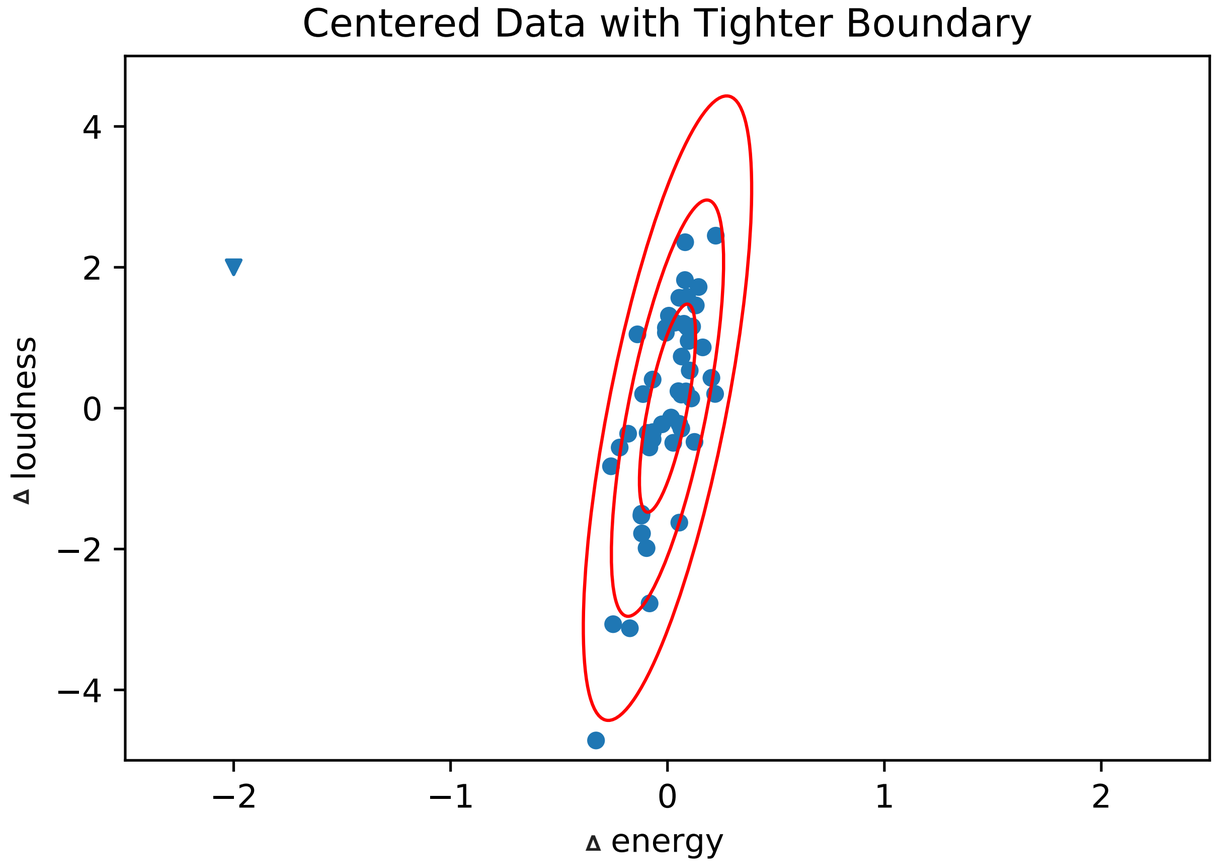
~~Yes~~.

Should it be though?

Easy isn’t always best.

Instead of a circle, maybe we can find a better shape.

Create a Better Boundary

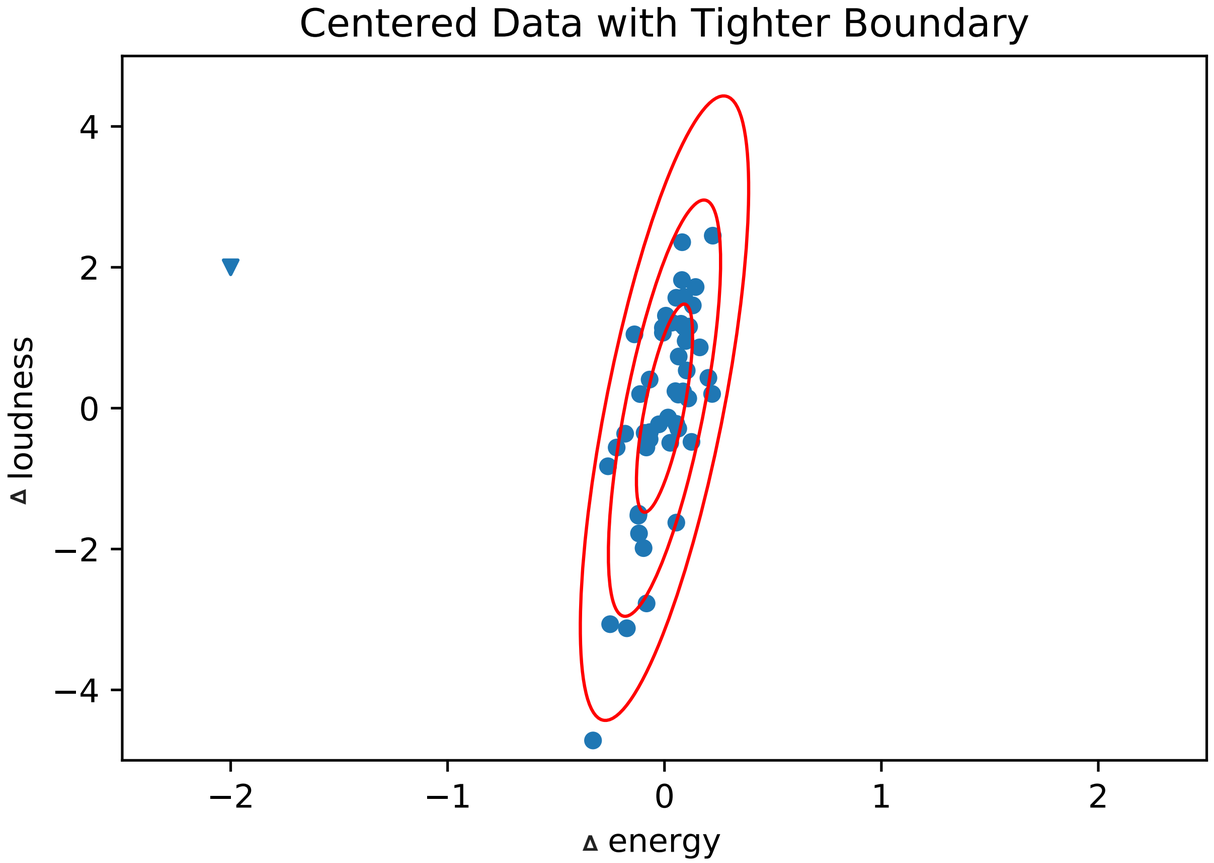


Steve Crow
@cr0wst

Prediction Ellipse

The axes of the ellipse point to the direction of least and most **variance**.

Variance


Centered Data with Tighter Boundary
4
2
0
−2
−4
▵ loudness
−2
−1
0
1
2
▵ energy

Singular Value Decomposition

One of the most known and widely used matrix decompositions. All matrices have a Singular Value Decomposition and it is commonly used in compression, denoising, and data reduction.

Singular Value Decomposition

$$A = U\Sigma V^T$$

Steve Crow
@cr0wst

Singular Value Decomposition

Python Scientific Computing Package

```
import numpy as np

u, s, vh = np.linalg.svd(
    songs.div(np.sqrt(len(songs))),
    full_matrices = False
)
```

$$\frac{M}{\sqrt{n}} = U\Sigma V^T$$

Singular Value Decomposition

Columns Point in Direction of Greatest to Least Variance
(The columns are the Principal Components)

```
u, s, vh = np.linalg.svd(
    songs.div(np.sqrt(len(songs))),
    full_matrices = False
)
```

Diagonal Matrix of PCA Weights
(Helps Tell How Necessary a Component Is)

Singular Value Decomposition

S

1.46530621
0.09094177

Singular Values
(How important is each PC?)

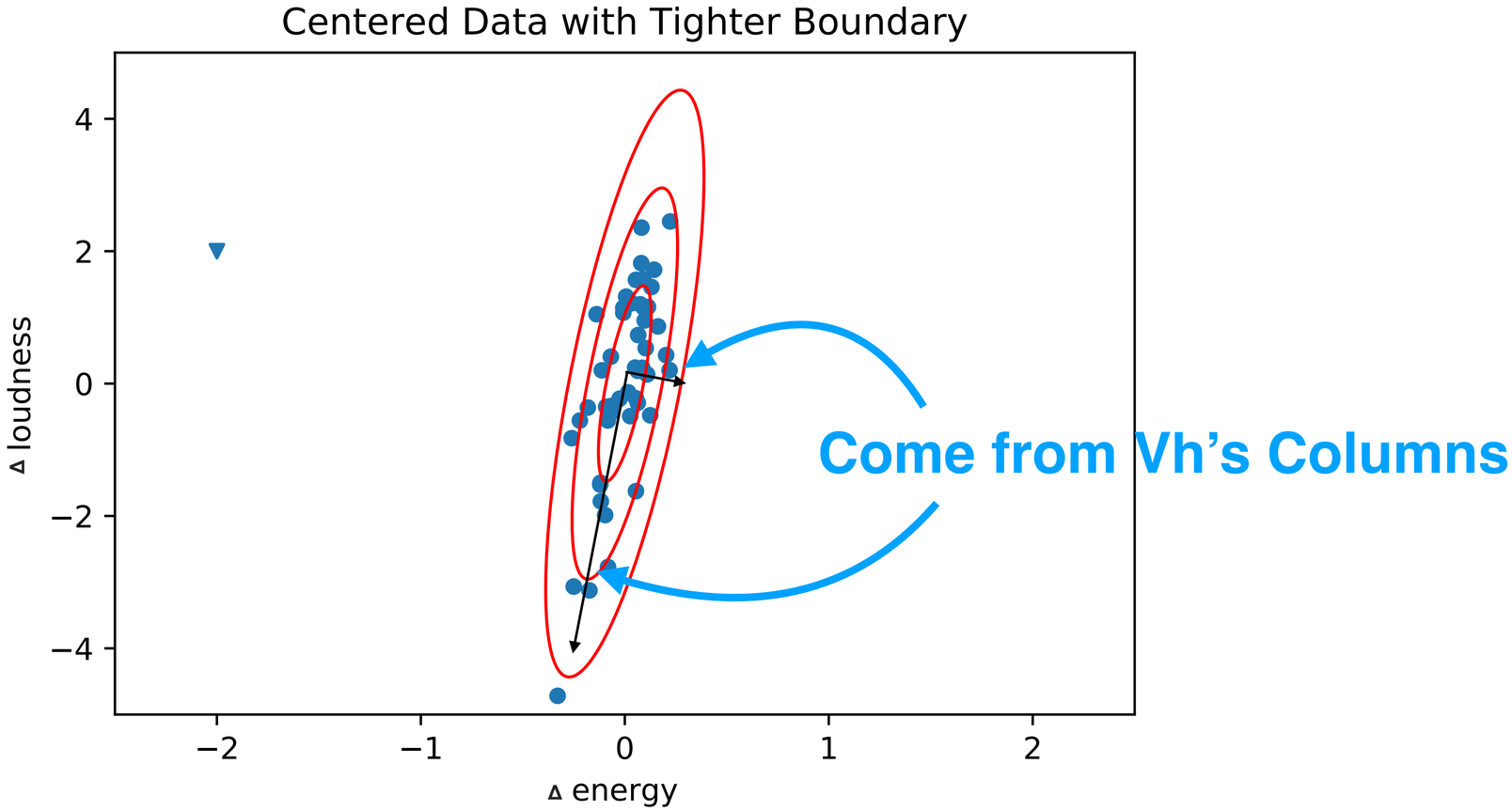
Vh

-0.0616217	-0.99809958
-0.99809958	0.0616217

Principal Components
(Columns are Vectors of Ellipse)

Steve Crow
@cr0wst

Singular Value Decomposition



Now how do we use the Principal Components?

Projections

```
song_proj = songs.dot(vh.T)
```

$$P = MV$$

(This is called projection)

Projections

	Δ energy	Δ loudness	P1	P2
Love The Way You Lie	0.220	0.202	-0.215	0.207
Hymn For The Weekend - Seeb Remi	0.144	1.720	-1.726	0.037
Uptown Funk	-0.096	-1.987	1.989	0.026
We Can't Stop	-0.083	-0.558	0.562	-0.049
This Is What You Came For	0.223	2.449	-2.458	0.071

All this math!

What’s the point?

Remember!

Principal Component Analysis can help with two things.

Feature Extraction

Create new features as a combination of existing features in an attempt to reduce any sort of redundancy.

Principal Components

	Δ energy	Δ loudness	P1	P2
Love The Way You Lie	0.220	0.202	-0.215	0.207
Hymn For The Weekend - Seeb Remi	0.144	1.720	-1.726	0.037
Uptown Funk	-0.096	-1.987	1.989	0.026
We Can't Stop	-0.083	-0.558	0.562	-0.049
This Is What You Came For	0.223	2.449	-2.458	0.071

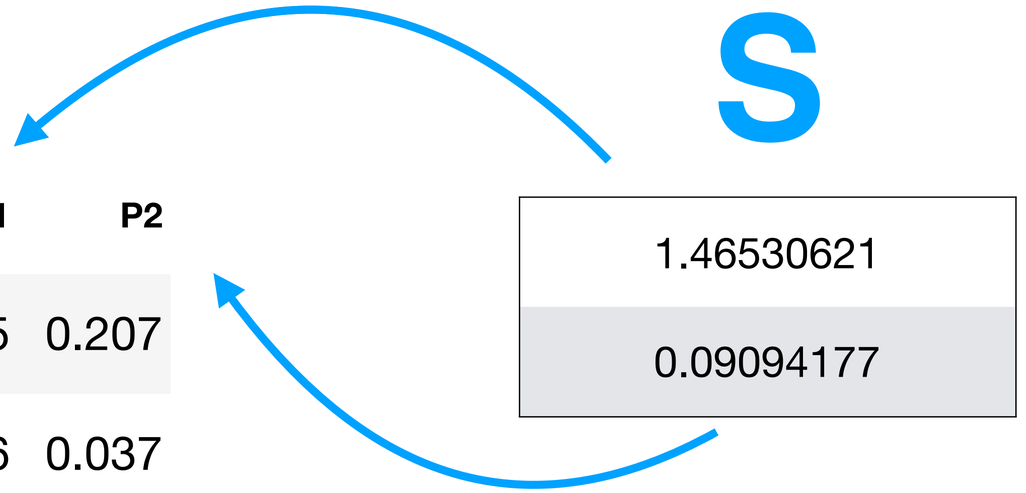
Warning!

Use **Principal Component Analysis** when you are okay with making your independent variables less interpretable.

Feature Elimination

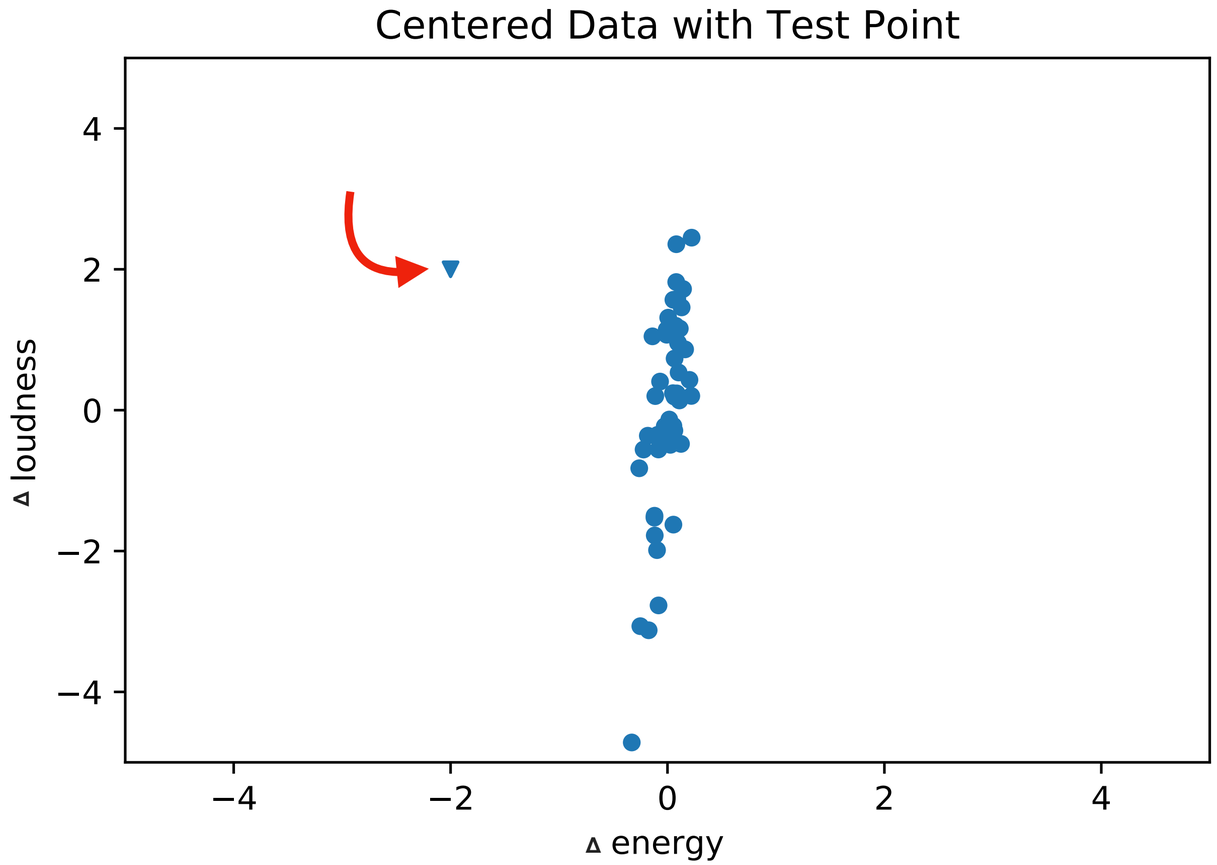
Reduce the feature space by eliminating features.

Principal Components

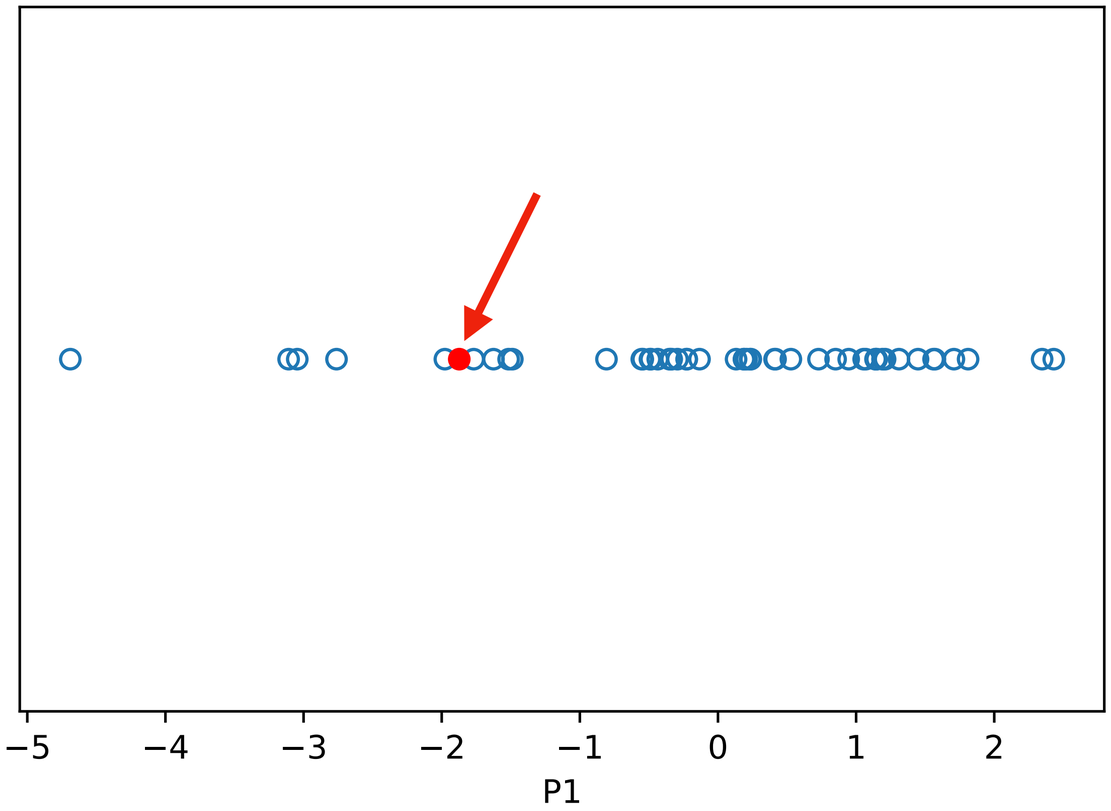


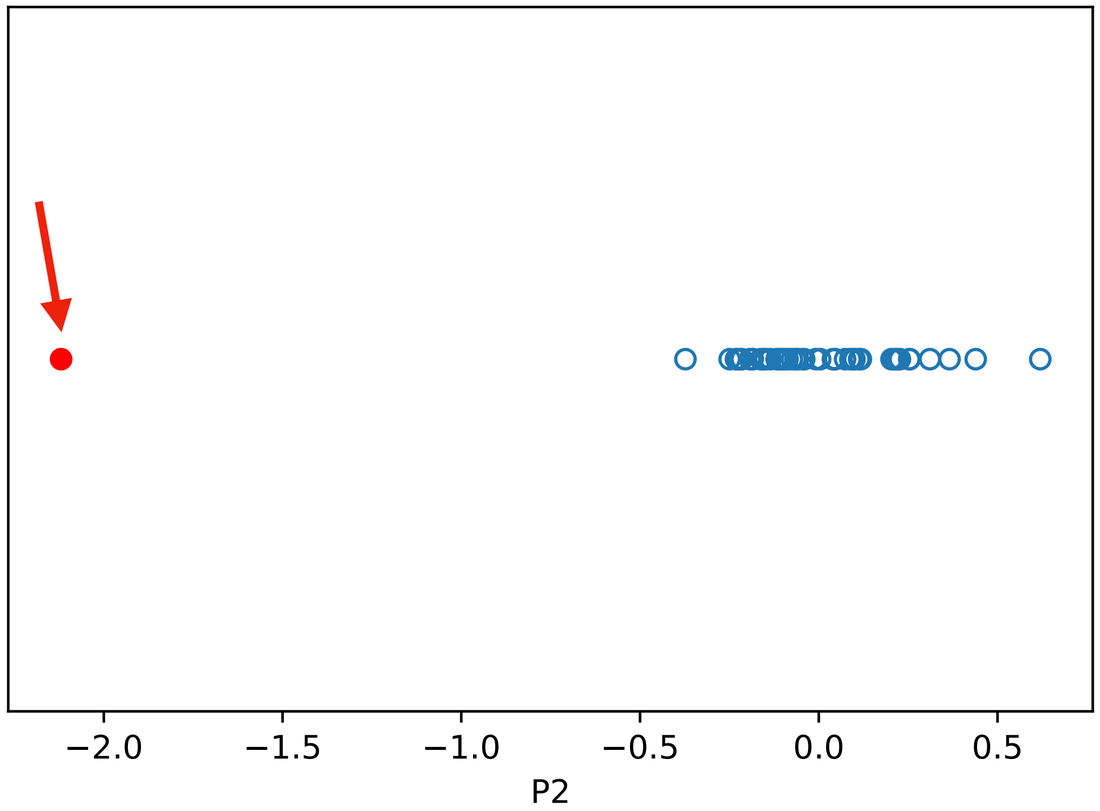
	P1	P2
Love The Way You Lie	-0.215	0.207
Hymn For The Weekend - Seeb Remi	-1.726	0.037
Uptown Funk	1.989	0.026
We Can't Stop	0.562	-0.049
This Is What You Came For	-2.458	0.071

Does this song belong?



Does this song belong?

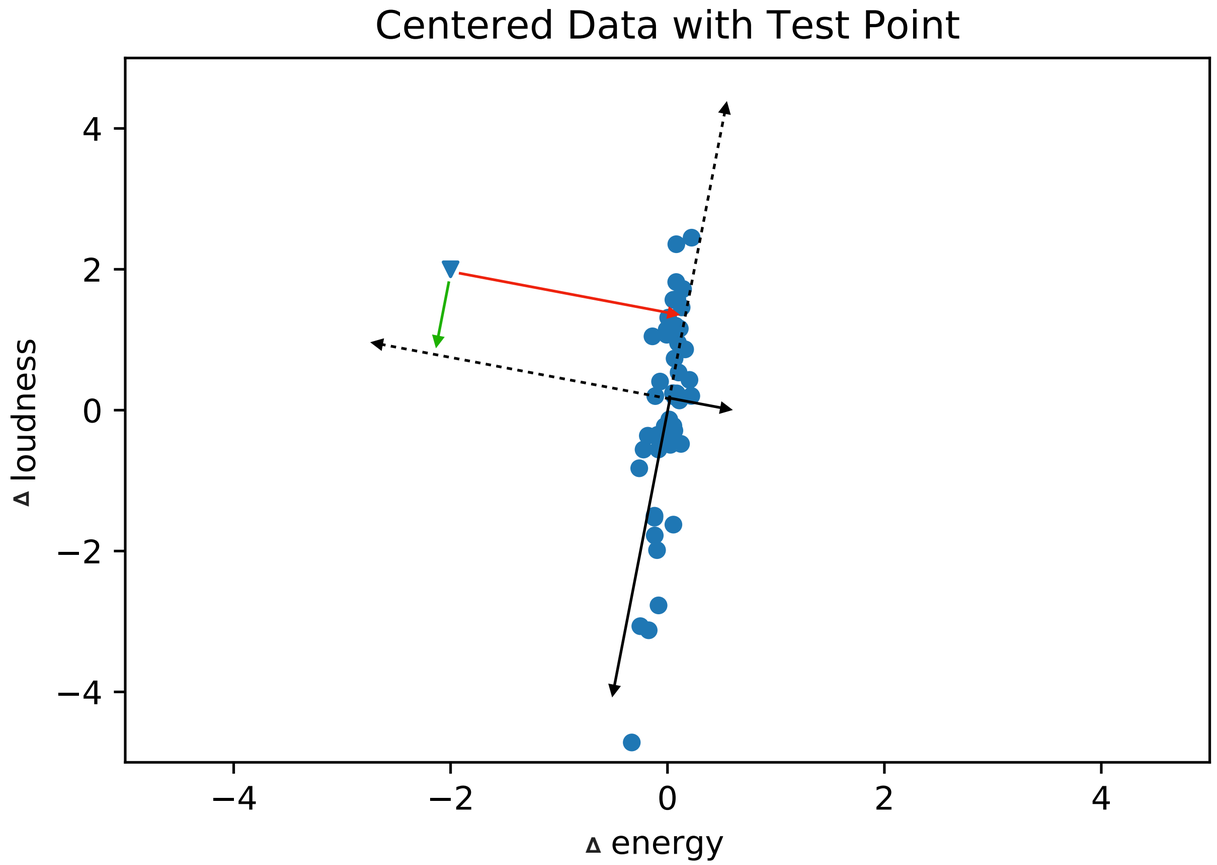




Steve Crow
@cr0wst

Making Faces
#CodeMash

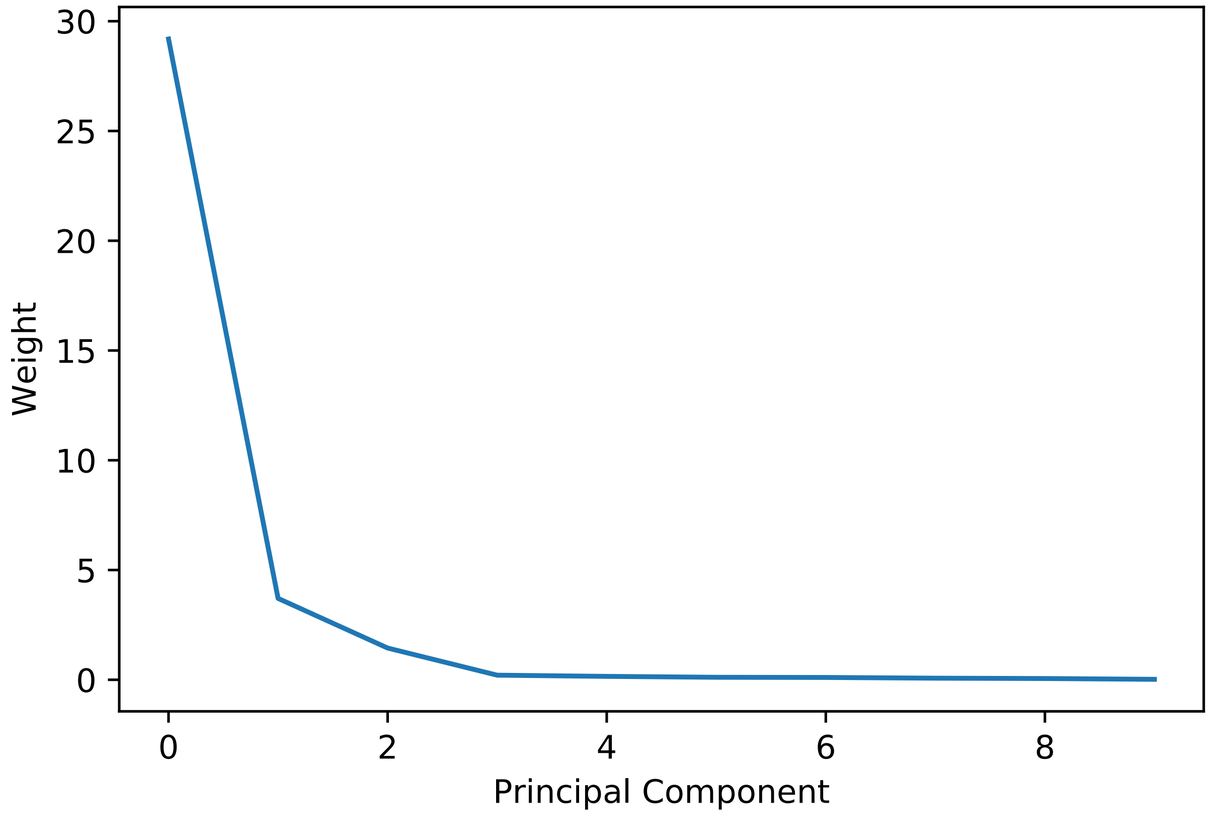
Does this song belong?



Steve Crow
@cr0wst

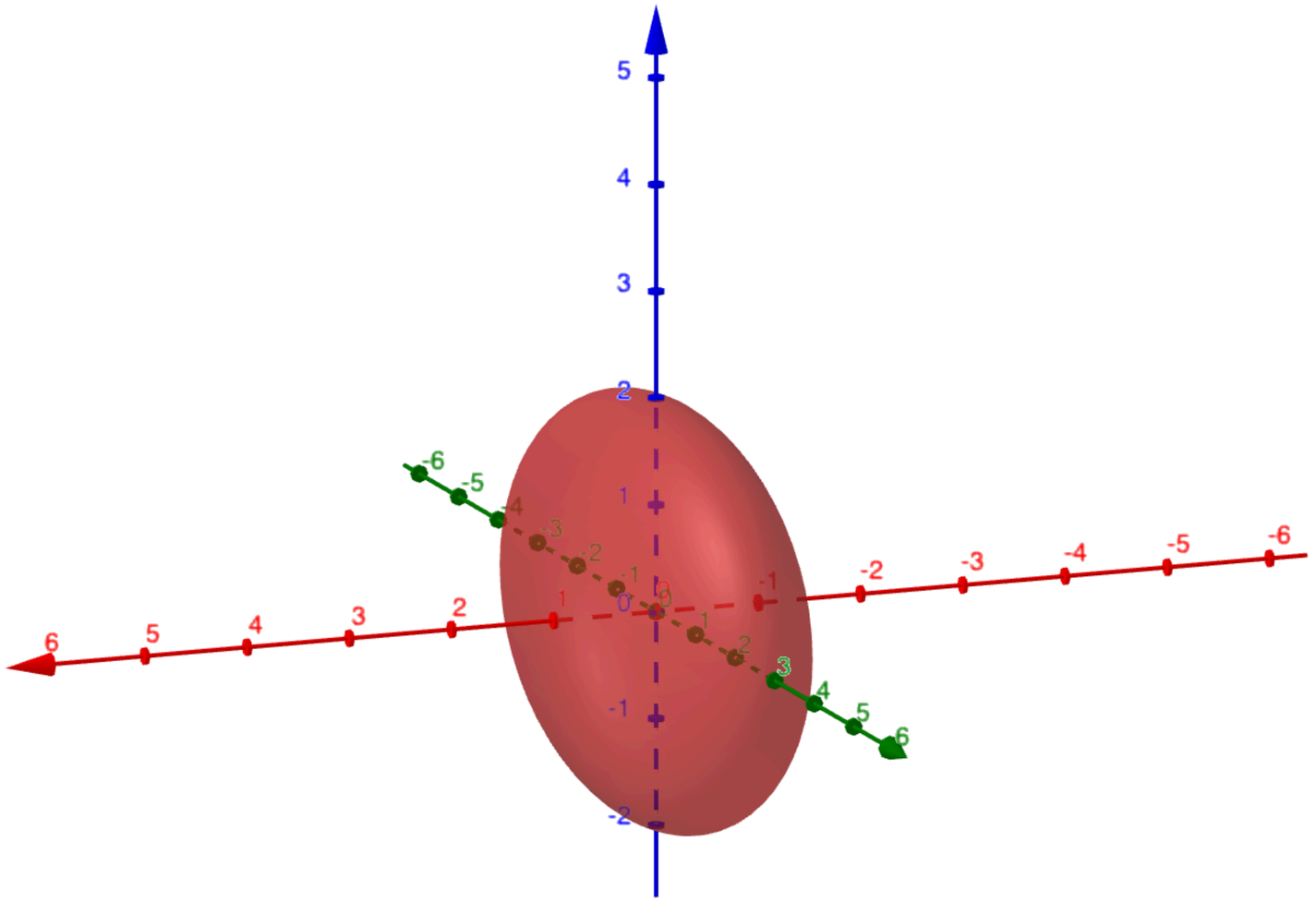
Two dimensions isn’t a lot.

Principal Component Weights



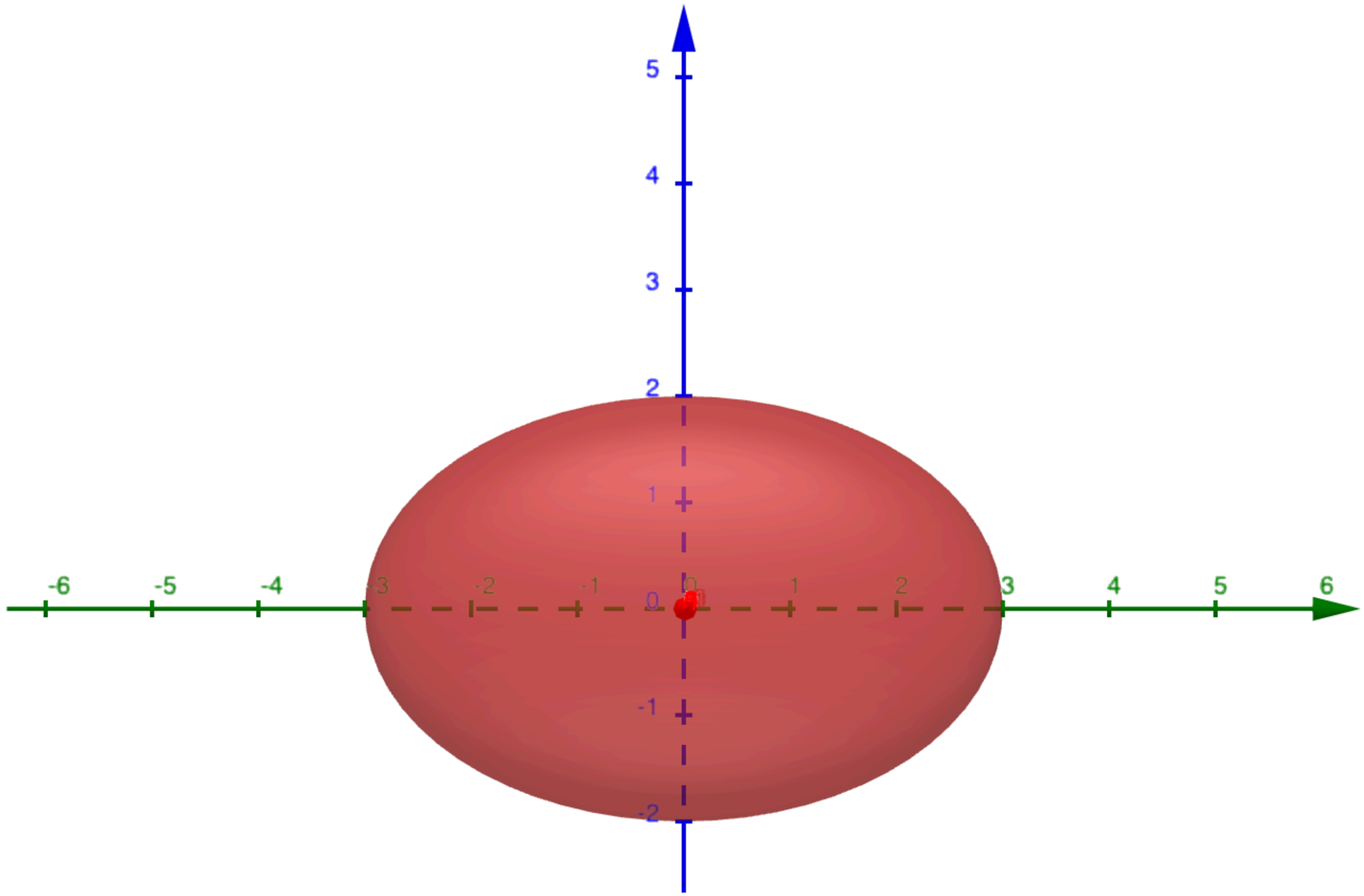
Principal Components

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Love The Way You Lie	-34.886	4.654	0.073	-0.137	-0.035	0.361	-0.148	0.034	-0.005	0.010
Hymn For The Weekend - Seeb Remi	-19.846	-5.380	1.421	0.218	0.075	-0.024	-0.069	0.037	-0.045	0.024
Uptown Funk	-6.864	-5.274	-2.193	-0.344	0.202	-0.100	0.016	0.050	0.018	-0.015
We Can't Stop	-41.859	-4.324	-1.005	0.185	0.171	0.128	-0.114	-0.146	0.103	-0.023
This Is What You Came For	2.070	3.603	2.575	0.054	-0.118	0.045	0.107	0.061	-0.143	-0.078



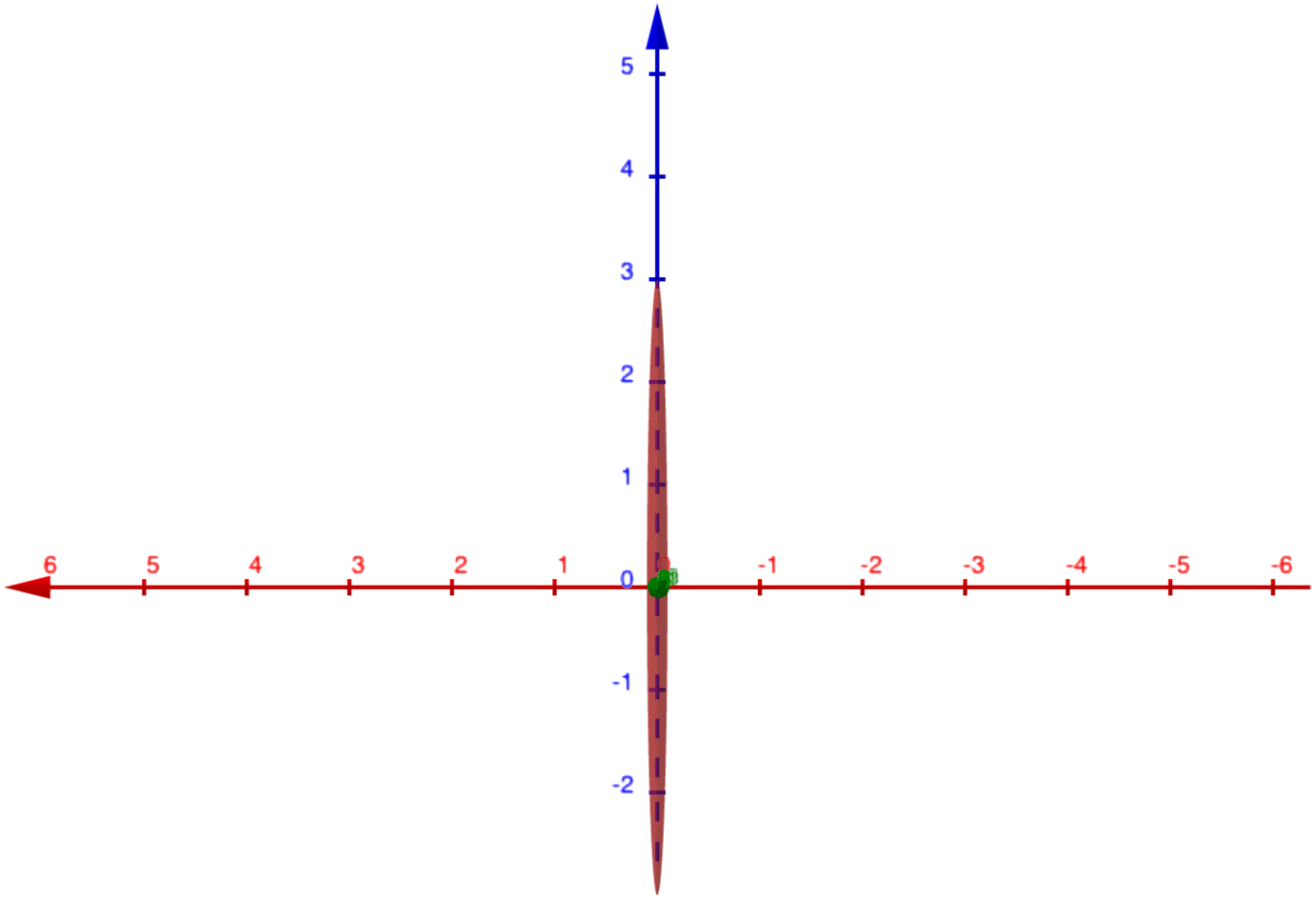
Steve Crow
@cr0wst

Making Faces
#CodeMash



Steve Crow
@cr0wst

Making Faces
#CodeMash



Steve Crow
@cr0wst

Making Faces
#CodeMash

Summary of Steps

Step 1: Build the Training Set

```
import pandas as pd

songs = pd.read_csv('songdata.csv')
```

Step 2: Normalize the Data

```
songs = songs.sub(songs.mean())
```

Step 3: Singular Value Decomposition

```
import numpy as np

u, s, vh = np.linalg.svd(
    songs.div(np.sqrt(len(songs))),
    full_matrices = False
)
```

Step 4: Calculate the Projections

```
song_proj = songs.dot(vh.T)
```

All Together

```
import numpy as np
import pandas as pd

# Step 1
songs = pd.read_csv('songdata.csv')

# Step 2
songs = songs.sub(songs.mean())

# Step 3
u, s, vh = np.linalg.svd(
    songs.div(np.sqrt(len(songs))),
    full_matrices = False
)

# Step 4
song_proj = songs.dot(vh.T)
```

The value of PCA is in its ability to reduce the number of dimensions.

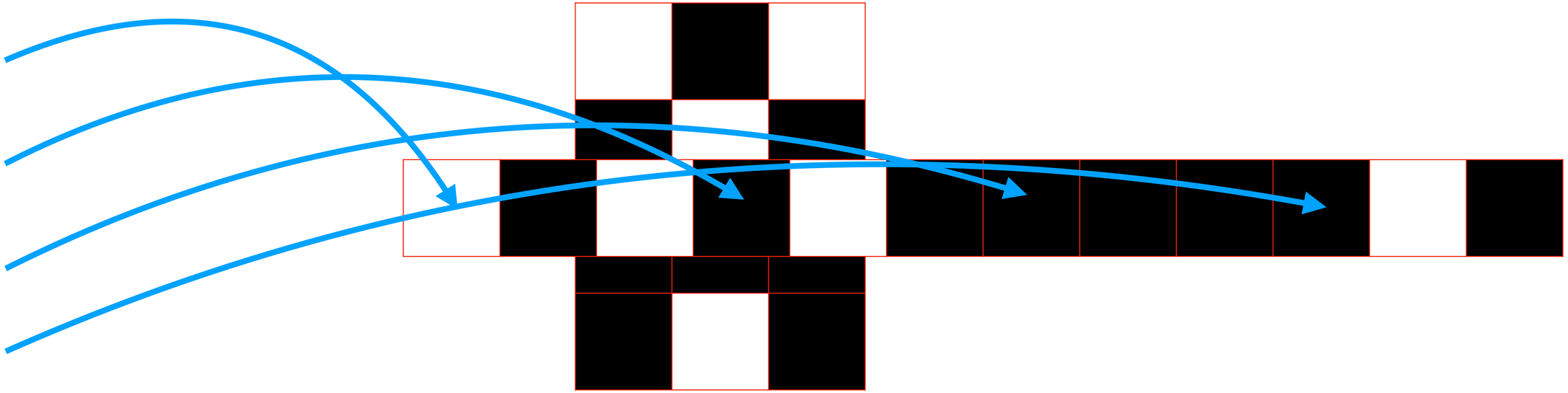
Face Dimension Reduction

Finally!

Images as Vectors

No convenient API this time!

Images as Vectors



A 4 by 3 image becomes a 1 by 12 image.

Each pixel represents a **single feature.**

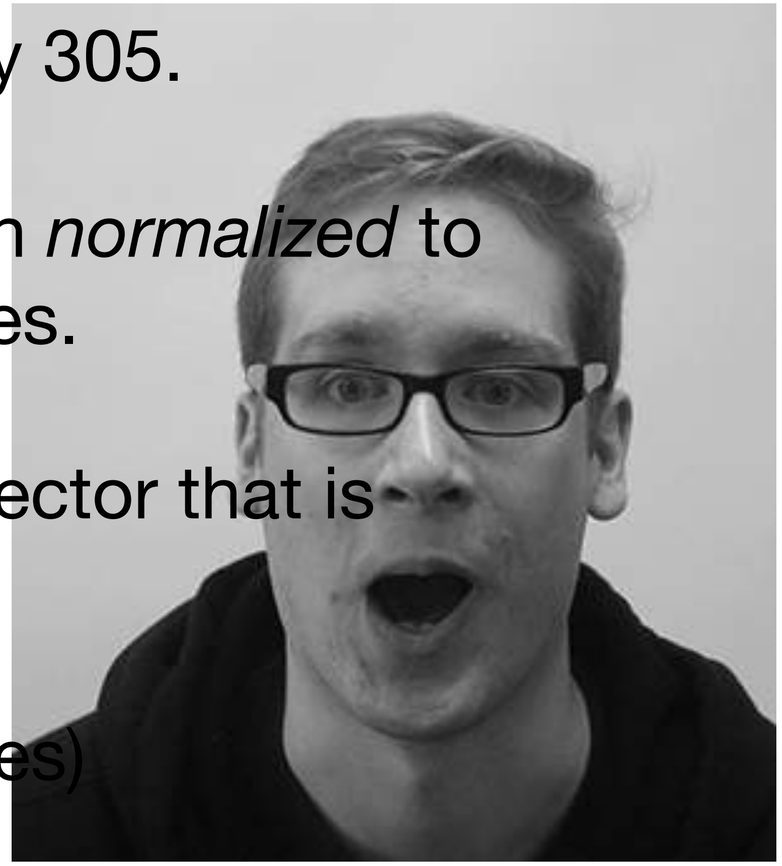
Face Images

Each image is 342 by 305.

Each image has been *normalized* to the best of my abilities.

This will produce a vector that is 104,310 pixels long.

(That’s a lot of features)



The Training Set

Average Face

Normalize the Training Set

Steve Crow
@cr0wst

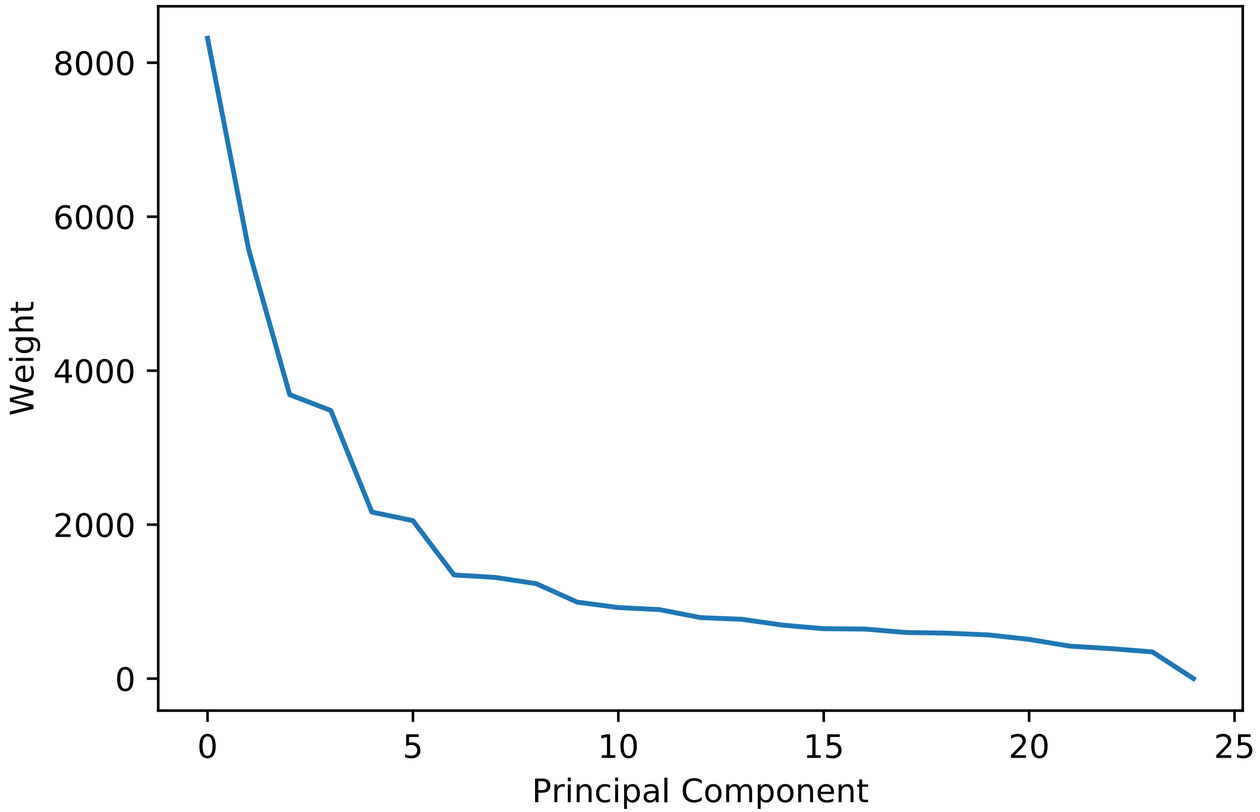
Normalize the Training Set

Steve Crow
@cr0wst

Principal Component Vectors

Steve Crow
@cr0wst

Principal Component Weights



Why only 25 components?

I thought it was equal to the number of features.

Having only 25 components can be a good thing!

Projection

Projection

Steve Crow
@cr0wst

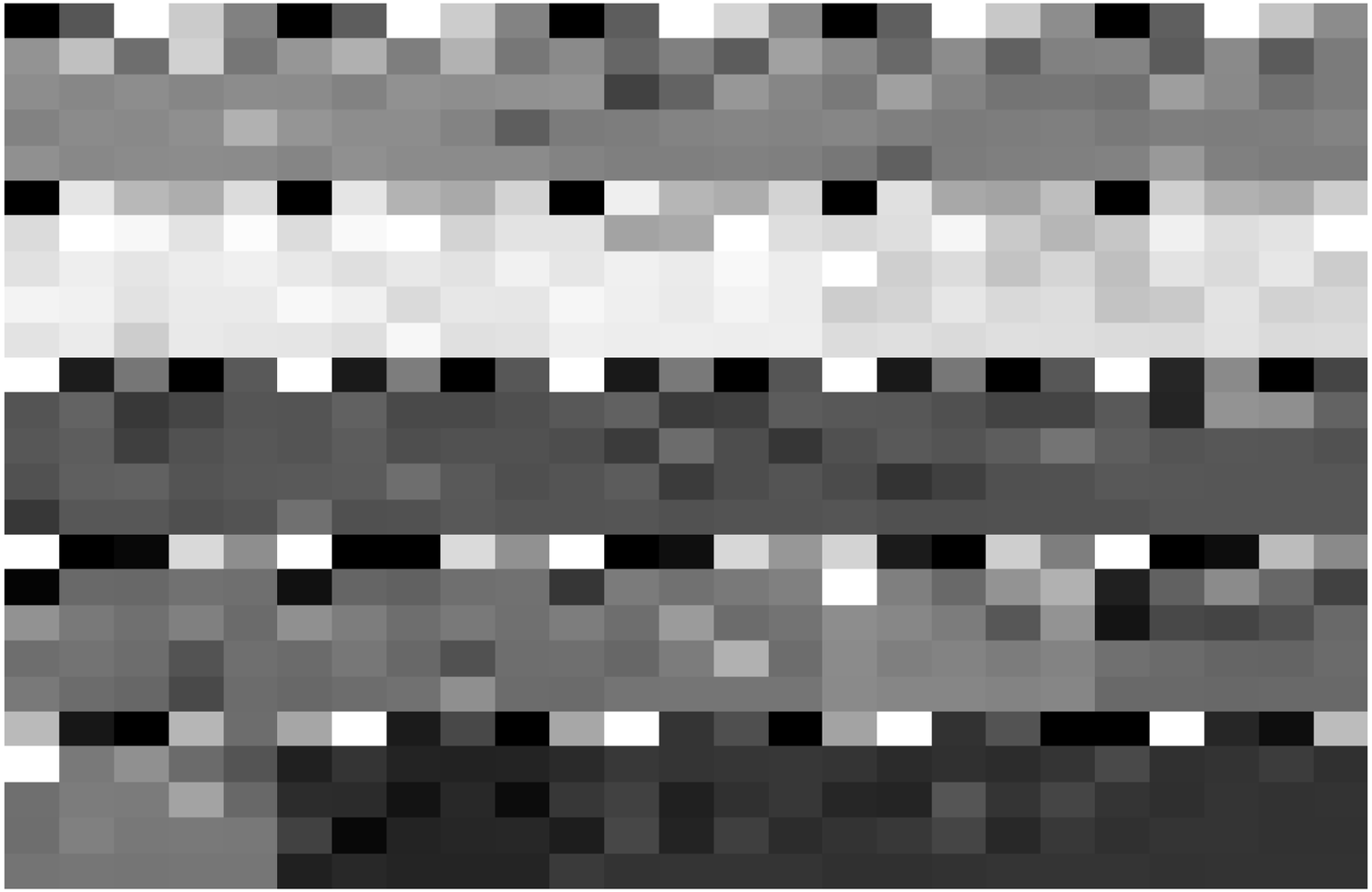
Projection

Steve Crow
@cr0wst

Projection

Steve Crow
@cr0wst

Projection



Steve Crow
@cr0wst

Let’s Test the Model

Image in Set

Person in Set

Not in Set

Project the Test Subjects

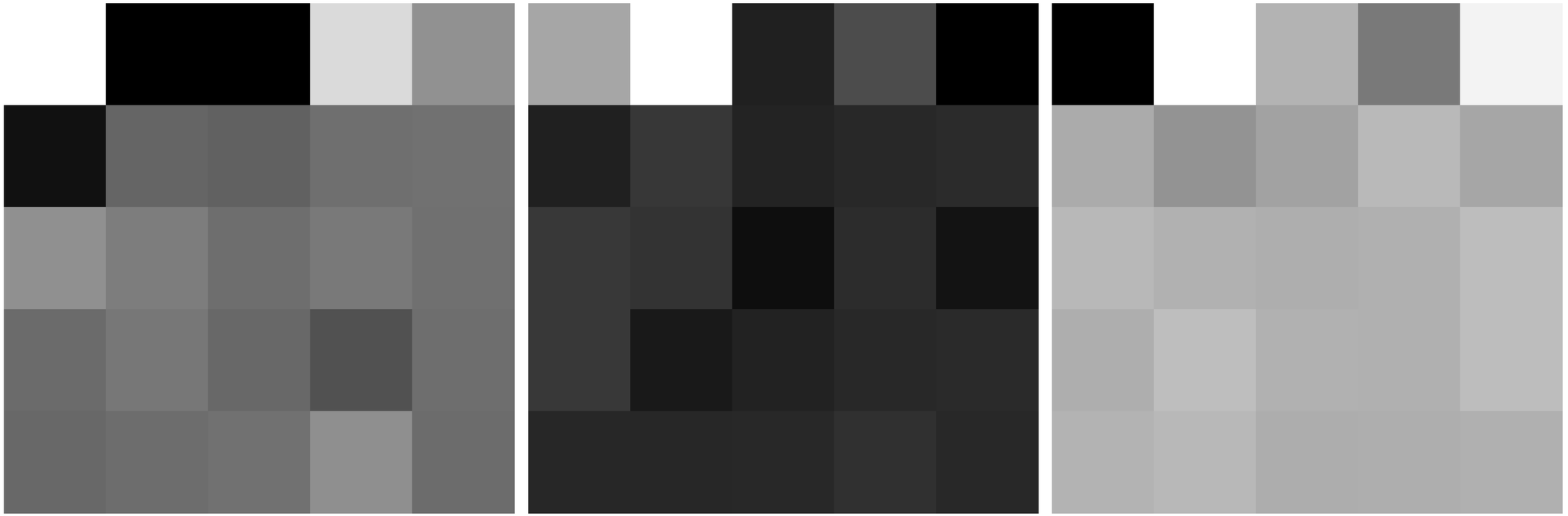


Image in Set

Person in Set

Not in Set

Compare the Projections

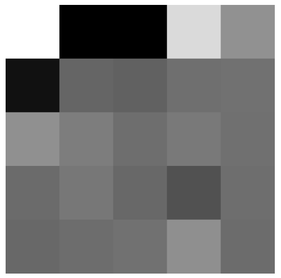
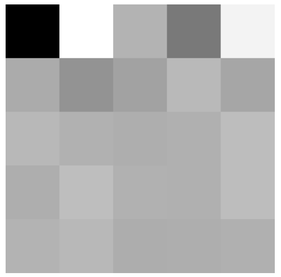
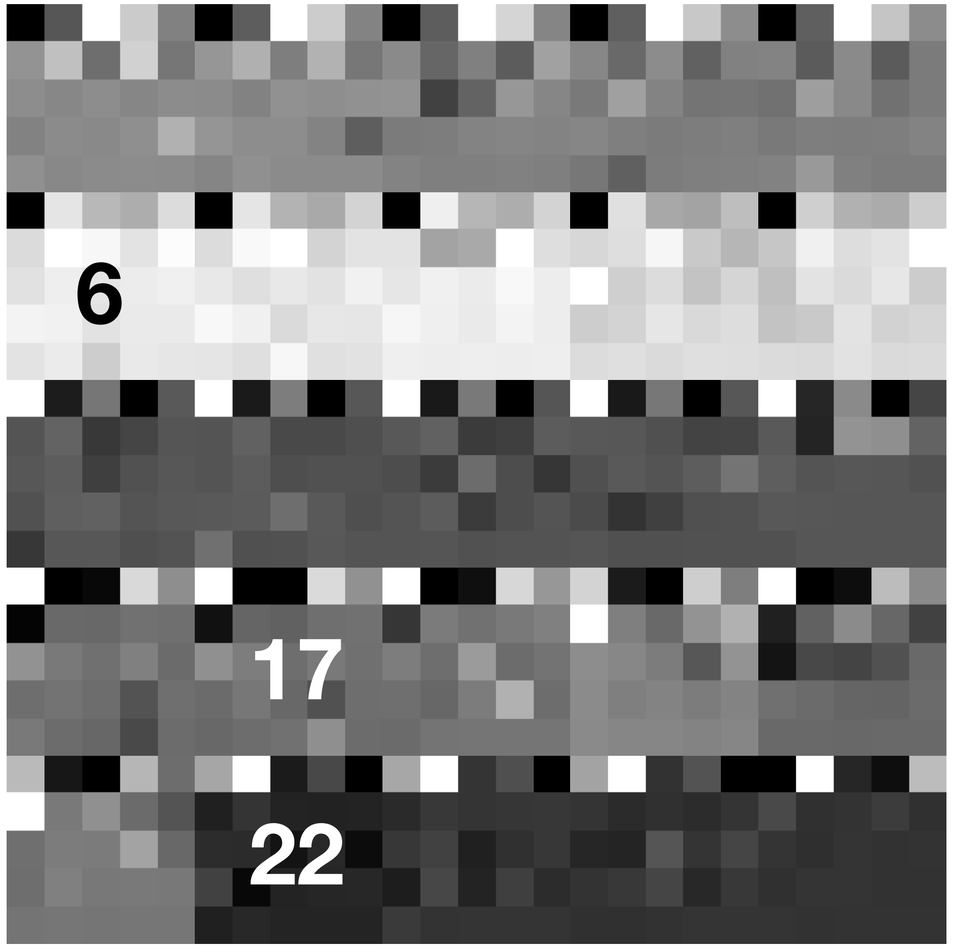


Image in Set
Closest to 17
Distance: 0

Person in Set
Closest to 22
Distance: 1452.54



Person Not in Set
Closest to 6
Distance: 9224.23



Results

Image in Set
Closest to 17
Distance: 0

Person in Set
Closest to 22
Distance: 1452.54

Person Not in Set
Closest to 6
Distance: 9224.23

Here’s the Process

Step 1: Build the Training Set

Glob is for getting all the file names.

```
import glob
import numpy as np
import cv2

images = []
for path in sorted(glob.glob('training_set/*.jpg')):
    images.append(cv2.imread(path, cv2.IMREAD_GRAYSCALE).flatten())

training_set = np.array(images)
```

Open Source Computer Vision Library

Read Images as Greyscale

Turn Each Image into a Row Vector

Step 2: Normalize the Data

```
training_set = training_set - training_set.mean(axis = 0)
```

Find the average row (image)

Step 3: Singular Value Decomposition

```
import numpy as np

u, s, vh = np.linalg.svd(
    training_set / np.sqrt(len(training_set))),
    full_matrices=False
)
```

Step 4: Calculate the Projections

```
face_proj = training_set.dot(vh.T)
```

All Together

```
import glob
import numpy as np
import cv2

# Step 1
images = []
for path in sorted(glob.glob('training_set/*.jpg')):
    images.append(cv2.imread(path, cv2.IMREAD_GRAYSCALE).flatten())

training_set = np.array(images)

# Step 2
training_set = training_set - training_set.mean(axis=0)

# Step 3
u, s, vh = np.linalg.svd(
    training_set / np.sqrt(len(training_set))),
    full_matrices = False
)

# Step 4
face_proj=training_set.dot(vh.T)
```

Final Thoughts

Links and Stuff

https://towardsdatascience.com/a-one-stop-shop-for-principal- component-analysis-5582fb7e0a9c

https://machinelearningmastery.com/singular-value- decomposition-for-machine-learning/

https://github.com/cr0wst/svd-and-pca

Steve Crow

Software Developer (NinjaCat.io)

@cr0wst

https://smcrow.net