Generalized Low Rank Models: Basics and Applications

Thursday, January 28, 2021

STARS Analytics and Artificial Intelligence

Julio Cárdenas-Rodríguez, Ph.D.

Principal Data Scientist

jcardenasrdz@uhc.com

United Healthcare®

Contents

Introduction to Generalized Linear Models:

some Linear Algebra and Optimization

Applications to the Penguins data set:

Clustering of features

Clustering of observations

Classification in the reduced space

Regression using features in the reduced space

The challenges of our data

age	gender	state	diabetic	prescriptions count	mbr type	• • •
67	F	AZ	1	?	A	• • •
67	F	NC	1	10	A	• • •
?	M	WA	0	2	W	• • •
29	?	IN	0	1000	D	• • •
⋮	⋮	⋮	⋮	⋮	⋮	

Structural

- Categorical features with very large dimensionality (zip code, prov NPI, gpi codes, etc)
- Missing value represent different things depending on the feature (age vs STAR measure)
- Mixed of data types: continuous, Boolean, categorical, ordinal, counts
- Large to very large data sets

Analytical

- Quantify the difference between to observations that differ only in their categorical data
- Avoid redundancy between features used for predictive / descriptive models
- How to impute missing entries

A general view of Low Rank Models

given: A, $k \ll m, n$

find: $X \in \mathbf{R}^{m \times k}$, $Y \in \mathbf{R}^{k \times n}$ for which

$$\underset{\text{numeric}}{\overset{k}{\begin{bmatrix} X \end{bmatrix}}} \; \overset{\text{numeric}}{\begin{bmatrix} \quad Y \quad \end{bmatrix}} \approx \underset{\text{data table}}{\begin{bmatrix} \quad A \quad \end{bmatrix}}$$

numeric | numeric | data table

i.e., $x_i y_j \approx A_{ij}$, where

$$\begin{bmatrix} X \end{bmatrix} = \begin{bmatrix} \text{—}x_1\text{—} \\ \vdots \\ \text{—}x_m\text{—} \end{bmatrix} \qquad \begin{bmatrix} \quad Y \quad \end{bmatrix} = \begin{bmatrix} | & & | \\ y_1 & \cdots & y_n \\ | & & | \end{bmatrix}$$

interpretation:

- X and Y are (compressed) representation of A
- $x_i^T \in \mathbf{R}^k$ is a point associated with example i
- $y_j \in \mathbf{R}^k$ is a point associated with feature j
- inner product $x_i y_j$ approximates A_{ij}

A numerical example of Low Rank Models: PCA (seen as an optimization problem)

PCA:

This is a least squares solution, just like linear regression

$$\text{minimize} \quad \|A - XY\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n (A_{ij} - x_i y_j)^2$$

with variables $X \in \mathbf{R}^{m \times k}$, $Y \in \mathbf{R}^{k \times n}$

- (analytical) solution via SVD of $A = U \Sigma V^T$:

$$X = U_k \Sigma_k^{1/2} \quad Y = \Sigma_k^{1/2} V_k^T$$

- (numerical) solution via alternating minimization

Generalized Low Rank Model (using alternating minimization)

MINIMIZE *only for Ω (observed data)*

$$\sum_{(i,j)\in\Omega} L_j(x_i y_j, A_{ij}) + \sum_{i=1}^{m} r_i(x_i) + \sum_{j=1}^{n} \tilde{r}_j(y_j)$$

Loss functions for each column

regularizer for X

regularizer for Y

Choose $L_j(x_i y_j, A_{ij})$ based on the data type:

Data type	possible loss functions
real	quadratic, absolute value, huber
boolean	logistic, hinge
integer	poisson
ordinal	small vs large, ordinal hinge
categorical	logit multinomial, one-vs-all

Regularizers

$$\sum_{(i,j)\in\Omega} L_j(x_i y_j, A_{ij}) + \sum_{i=1}^{m} r_i(x_i) + \sum_{j=1}^{n} \tilde{r}_j(y_j)$$

Loss functions for each column

regularizer for X

regularizer for Y

choose regularizers r, $\tilde{r}$ to impose structure:

structure	$r(x)$	$\tilde{r}(y)$
small	$\|x\|_2^2$	$\|y\|_2^2$
sparse	$\|x\|_1$	$\|y\|_1$
nonnegative	$\mathbf{1}(x \geq 0)$	$\mathbf{1}(y \geq 0)$
clustered	$\mathbf{1}(\mathbf{card}(x) = 1)$	0

Recovery of known matrix factorization methods

$$\sum_{(i,j)\in\Omega} L_j(x_i y_j, A_{ij}) + \sum_{i=1}^{m} r_i(x_i) + \sum_{j=1}^{n} \tilde{r}_j(y_j)$$

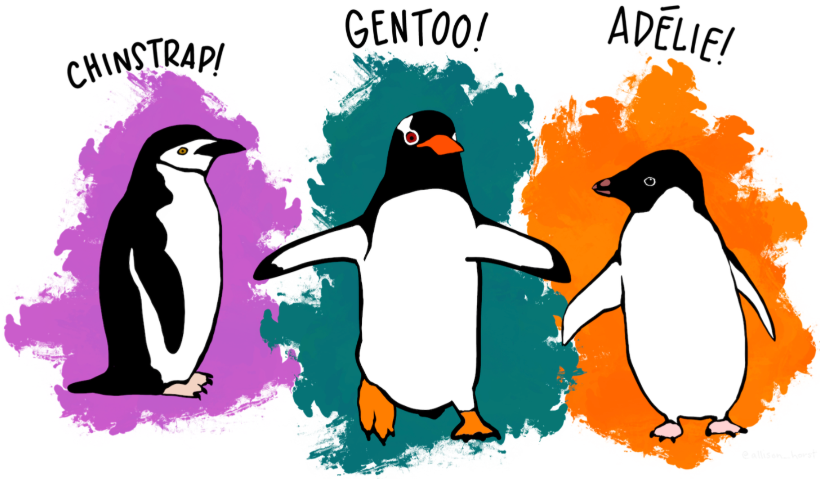
Loss functions for each column

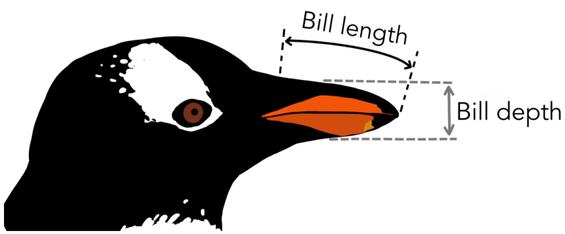
regularizer for X

regularizer for Y

X loss	Y loss	X reg	Y reg	name
quadratic	quadratic	0	0	PCA
quadratic	quadratic	NonNegConstraint	Non Neg Constraint	NNMF
quadratic	quadratic	0	Unit One Sparse Constraint	K-means
huber	huber	quadratic	quadratic	robust PCA

Applications to the Penguins data set

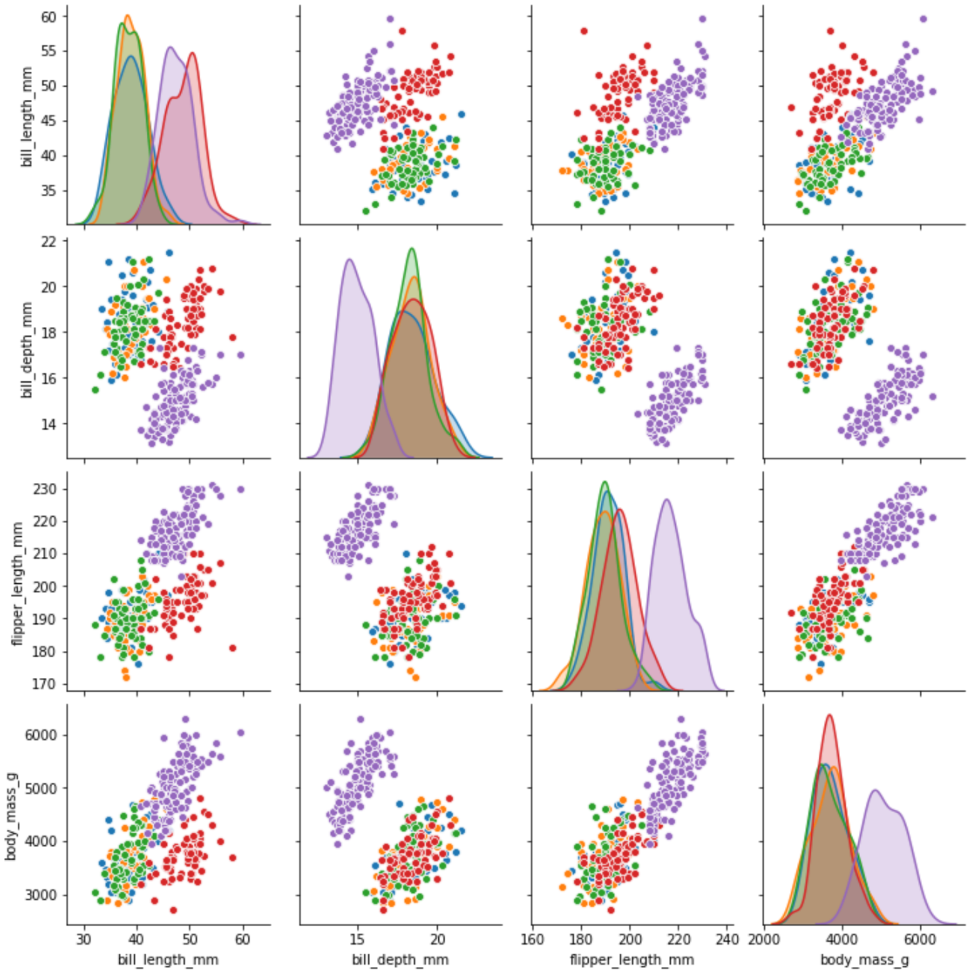




Feature	Description
species	categorical with 3 levels
island	categorical with 3 levels
bill_length_mm	numerical
bill_depth_mm	numerical
flipper_length_mm	numerical
body_mass_g	numerical
sex	boolean

https://github.com/allisonhorst/palmerpenguins

Applications to the Penguins data set

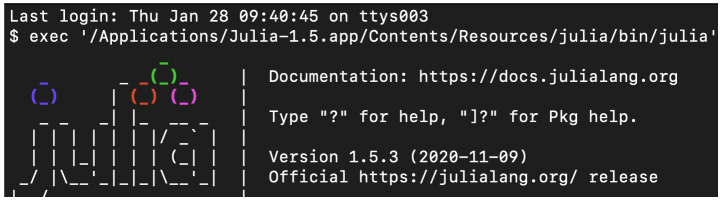


Questions:

- Can we cluster ***features*** based on their similarity? → Y
- Can we cluster ***penguins*** based on their similarity? → X
- Can we classify with only two features ? -→ cols of X)

$$\begin{bmatrix} X \end{bmatrix} \begin{bmatrix} \quad Y \quad \end{bmatrix} \approx \begin{bmatrix} \qquad A \qquad \end{bmatrix}$$

https://github.com/allisonhorst/palmerpenguins



Low Rank Model of the Penguins Data Set

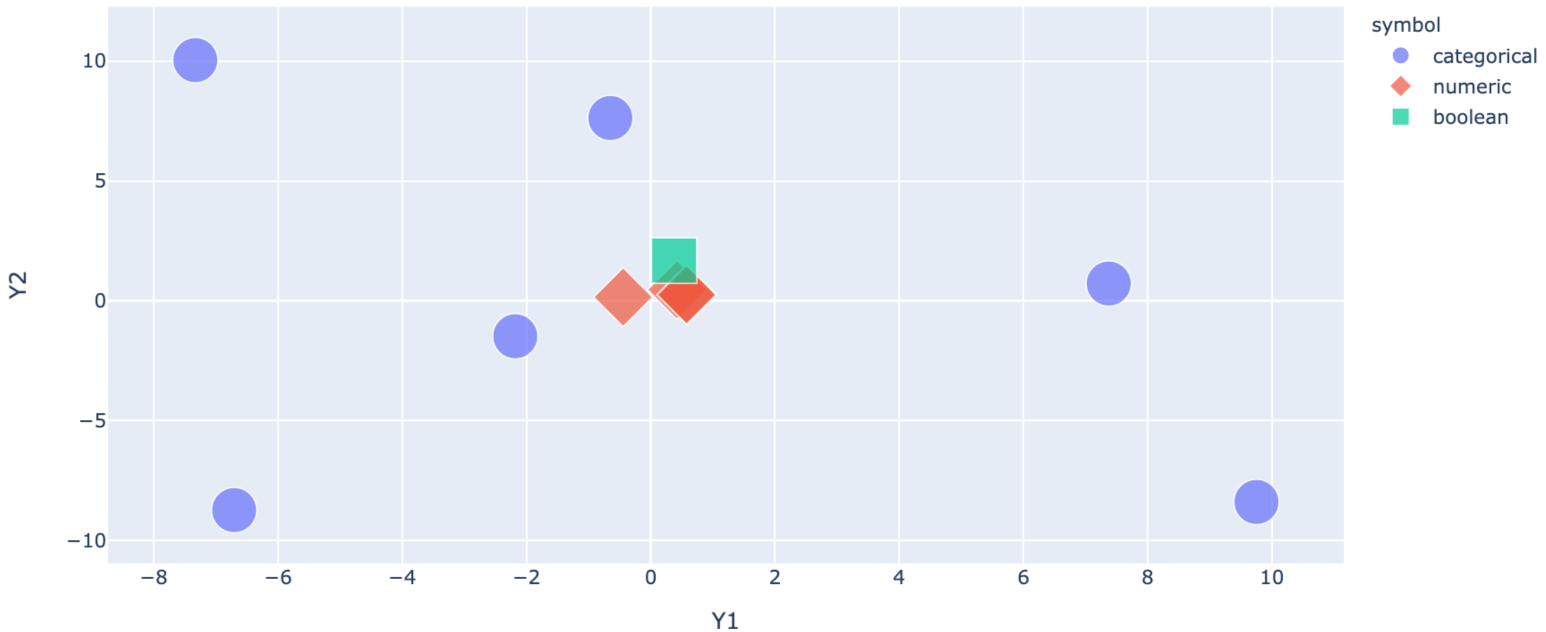
k = 2 p = 11 n = 7

$$m = 344 \quad \begin{bmatrix} X \end{bmatrix} \begin{bmatrix} \quad Y \quad \end{bmatrix} \approx \begin{bmatrix} \quad A \quad \end{bmatrix} \quad m = 344$$

Feature	Loss	hot encoded col size
species	MultinomialLoss	3
island	MultinomialLoss	3
bill_length_mm	QuadLoss	N/A
bill_depth_mm	QuadLoss	N/A
flipper_length_mm	QuadLoss	N/A
body_mass_g	QuadLoss	N/A
sex	LogisticLoss	N/A

Applications to the Penguins data set:

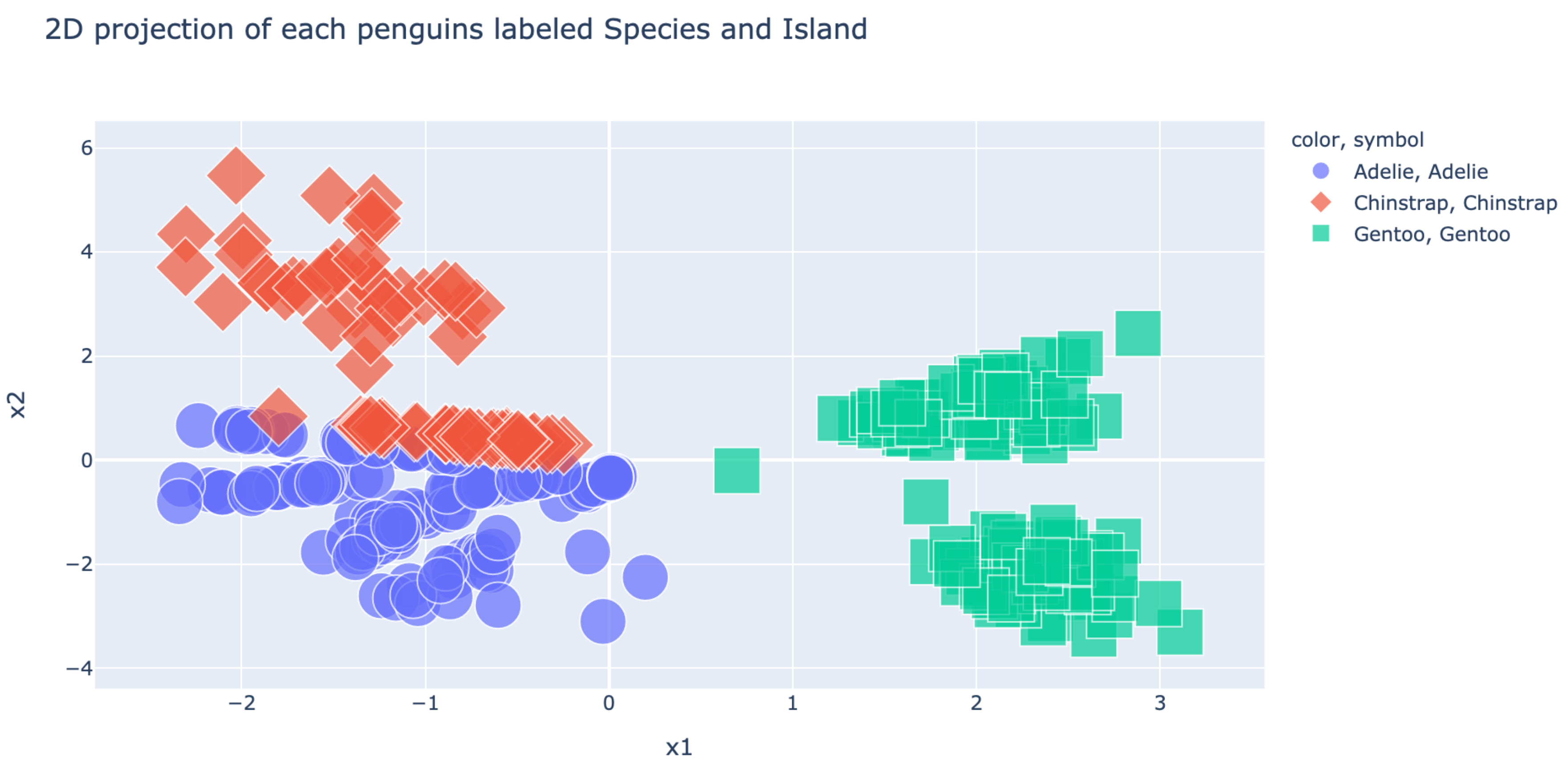
cluster ***features*** *based on their similarity*



[Click here for interactive plot]

Applications to the Penguins data set:

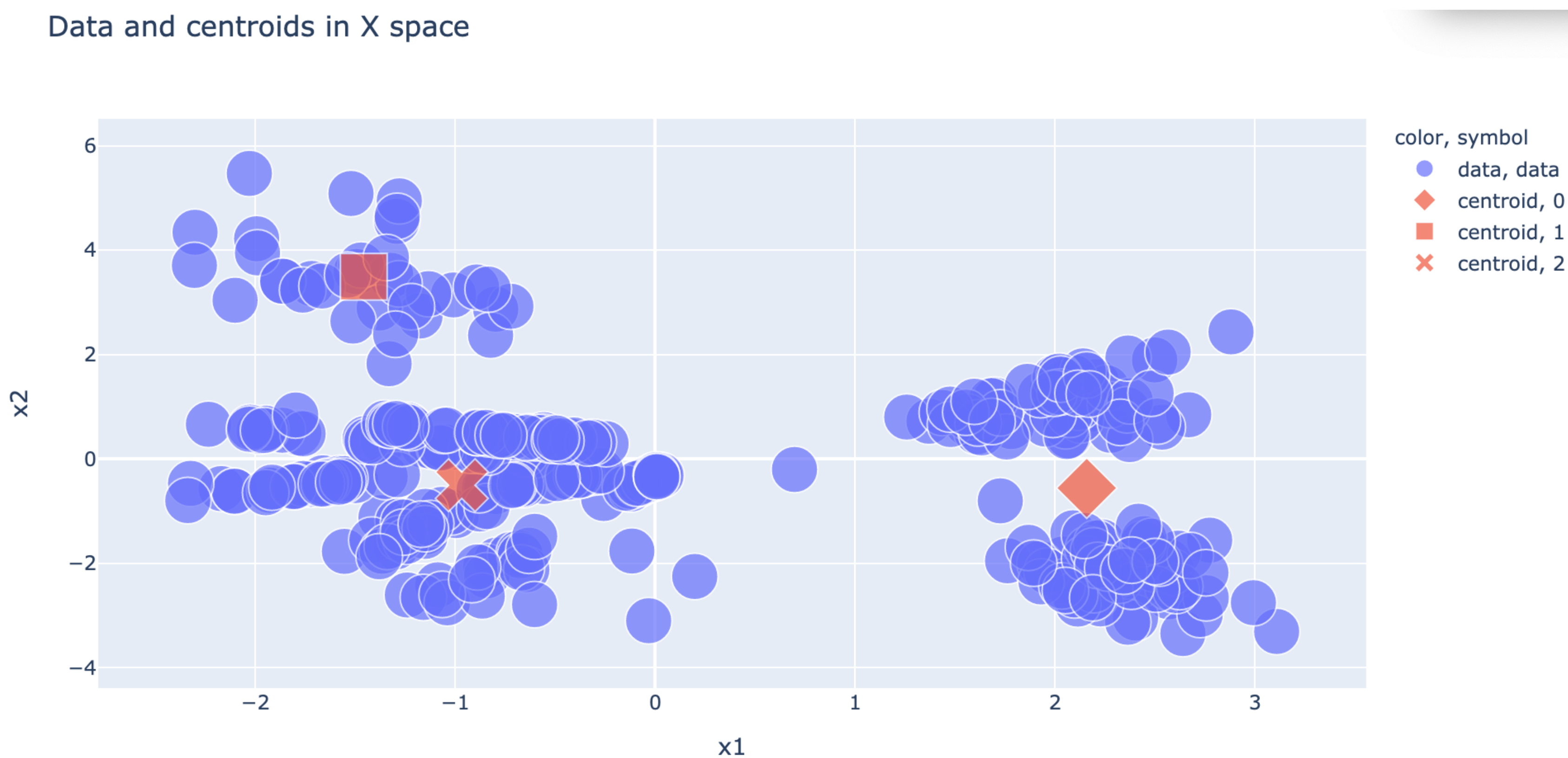
cluster <u>penguins</u> based on their similarity



Click here for interactive plot

Applications to the Penguins data set:

cluster penguins based on their similarity cont'



	x1	x2
centroid 01	2.158236	-0.557577
centroid 02	-1.457938	3.489020
centroid 03	-0.967619	-0.503565

Applications to the Penguins data set:

cluster ***penguins*** *based on their similarity cont'*

Xc

	x1	x2
centroid 01	2.158236	-0.557577
centroid 02	-1.457938	3.489020
centroid 03	-0.967619	-0.503565

Y
Loss functions
rescaling

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	male
centroid 01	2	1	48.0	15.0	217.0	5084	False
centroid 02	3	2	49.0	19.0	200.0	4226	True
centroid 03	1	3	40.0	18.0	191.0	3656	False

Applications to the Penguins data set:

cluster ***penguins*** *based on their similarity cont'*

Xc

	x1	x2
centroid 01	2.158236	-0.557577
centroid 02	-1.457938	3.489020
centroid 03	-0.967619	-0.503565

Xc * Y

```
0        species_Adelie
1     species_Chinstrap
2        species_Gentoo
3         island_Biscoe
4          island_Dream
5      island_Torgersen
6        bill_length_mm
7         bill_depth_mm
8     flipper_length_mm
9           body_mass_g
10             sex_MALE
```

	0	1	2	3	4	5	6	7	8	9	10
0	-9.598765	15.516327	-5.652068	25.731506	-21.418382	-3.877056	0.659424	-1.039494	1.140533	1.102101	-0.120372
1	-20.720394	-8.257501	27.549910	-43.525930	45.714942	-2.002966	0.972568	1.161620	-0.029700	0.029923	5.293447
2	10.890107	-7.495050	-3.211774	-5.206641	2.037486	2.857284	-0.638968	0.354812	-0.690692	-0.681611	-1.207964

Loss functions and rescaling

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	male
centroid 01	2	1	48.0	15.0	217.0	5084	False
centroid 02	3	2	49.0	19.0	200.0	4226	True
centroid 03	1	3	40.0	18.0	191.0	3656	False

Applications to the Penguins data set:

Classification with only two features

```
from sklearn.ensemble import RandomForestClassifier as RFC
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn import metrics

d = A_raw.merge(X[['x1','x2']],left_index=True, right_index=True)
d = d.dropna()

le = preprocessing.LabelEncoder()
Y_str = d.species
le.fit(Y_str)

Ydata = le.transform(Y_str)
Xdata = d[['x1','x2']]

X_train, X_test, y_train, y_test = train_test_split(Xdata, Ydata, test_size=0.33, random_state=42)

clf = RFC(n_estimators=2,max_depth=2)
clf.fit(X_train, y_train)

y1 = le.inverse_transform( y_test )
y2 = le.inverse_transform(clf.predict(X_test) )

print( metrics.classification_report(y1, y2) )
```

```
              precision    recall  f1-score   support

      Adelie       0.96      0.94      0.95        52
   Chinstrap       0.86      0.90      0.88        20
      Gentoo       1.00      1.00      1.00        38

    accuracy                           0.95       110
   macro avg       0.94      0.95      0.94       110
weighted avg       0.96      0.95      0.95       110
```

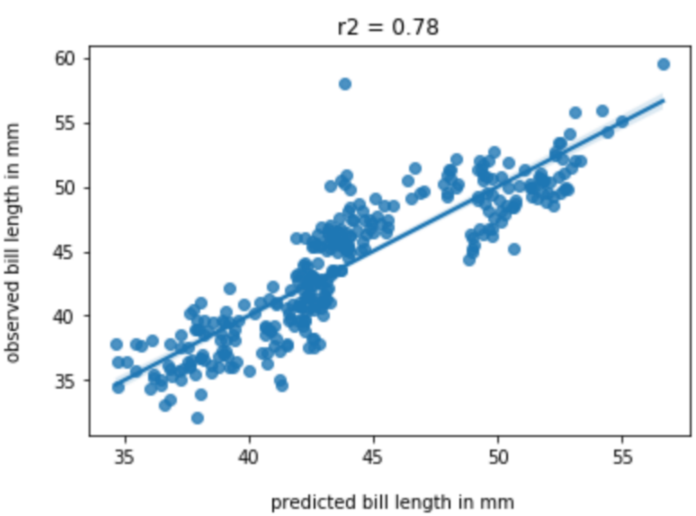
Linear Regression of `bill_length_mm` using `X` space matrix

```
# Xdata matrix with an intercept
Xdata = d[['x1','x2']].copy()
Xdata['c'] = 1

# outcome
Ydata = d['bill_length_mm']

# get coefficients of `b1 by pseudo inverse
b = np.linalg.pinv(Xdata) @ Ydata

pd.DataFrame(b,index=['x1 coeff','x2 coeffect','intercept'],columns=['linear model']).T
```

	x1 coeff	x2 coeffect	intercept
linear model	-2.304419	-2.480798	43.975264

```
# predict data
Yhat = Xdata @ b
sns.regplot(y=Ydata, x= Yhat)
plt.title('r2 = ' + str(metrics.r2_score(Ydata, Yhat))[0:4] );
plt.xlabel('\n predicted bill length in mm')
plt.ylabel('observed bill length in mm\n')
```

```
Text(0, 0.5, 'observed bill length in mm\n')
```



Applications to the Penguins data set:

Regression with only two features