Berkeley MSSE
MOLECULAR SCIENCE & SOFTWARE ENGINEERING

Lecture 03:

Dimension Reduction and PCA

Markus Hohle

University California, Berkeley

Machine Learning Algorithms

MSSE 277B, 3 Units

Course Map

Lecture 1: Course Overview and Introduction to Machine Learning

Lecture 2: Bayesian Methods in Machine Learning

classic ML tools & algorithms

Lecture 3: Dimensionality Reduction: Principal Component Analysis

Lecture 4: Linear and Non-linear Regression and Classification

Lecture 5: Unsupervised Learning: Clustering and Gaussian Mixture Models

Lecture 6: Adaptive Learning and Gradient Descent Optimization Algorithms

Lecture 7: Introduction to Artificial Neural Networks - The Perceptron

ANNs/AI/Deep Learning

Lecture 8: Introduction to Artificial Neural Networks - Building Multiple Dense Layers

Lecture 9: Convolutional Neural Networks (CNNs) - Part I

Lecture 10: CNNs - Part II

Lecture 11: Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs)

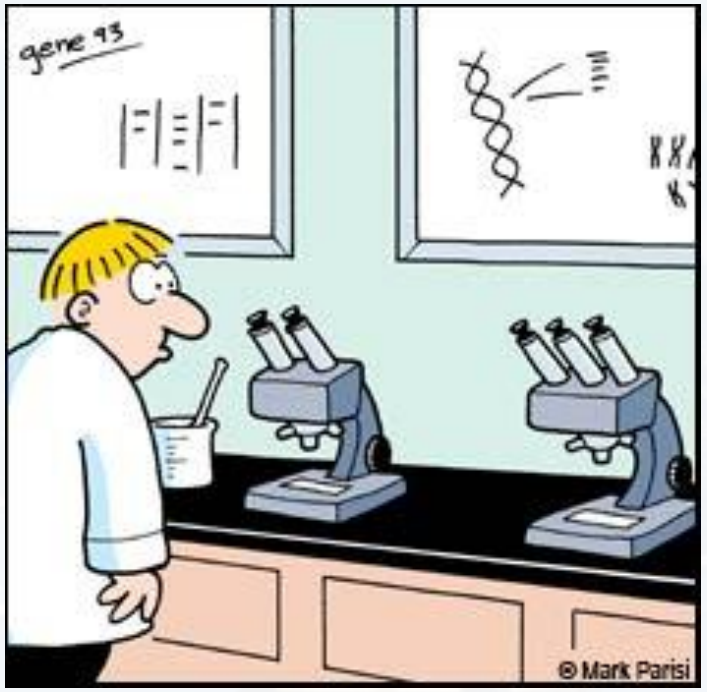
Lecture 12: Combining LSTMs and CNNs

Lecture 13: Running Models on GPUs and Parallel Processing

Lecture 14: Project Presentations

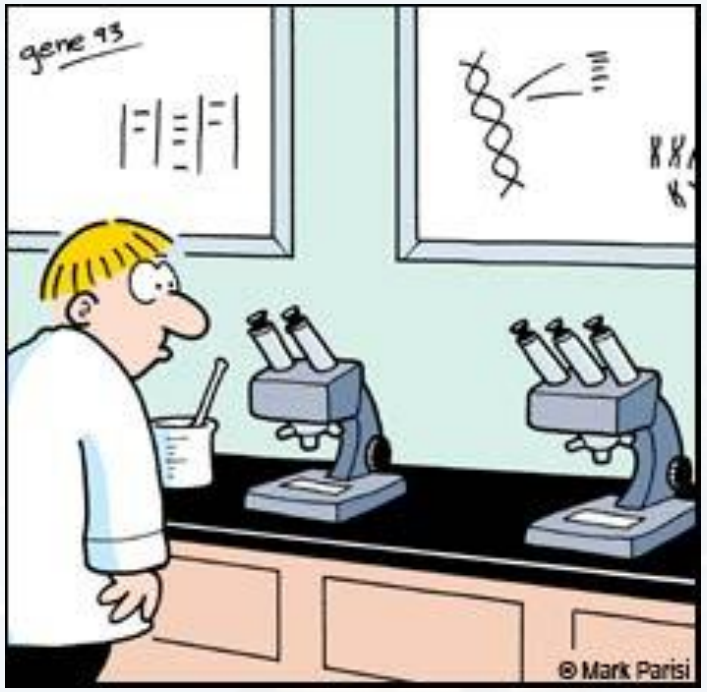
Lecture 15: Transformer

Lecture 16: GNN



Outline

- The Problem
- Mathematical formulation of the Problem
- Examples



Outline

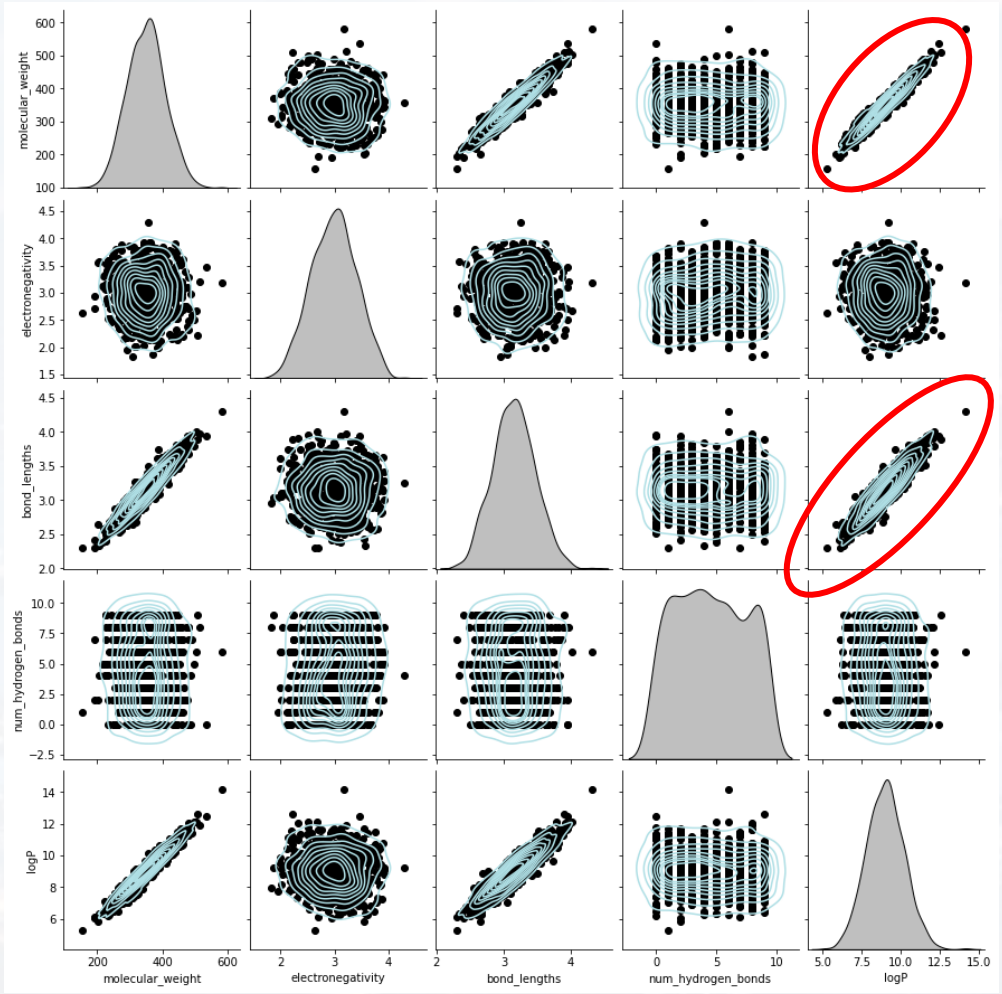
- The Problem
- Mathematical formulation of the Problem
- Examples

Dimension Reduction and PCA:

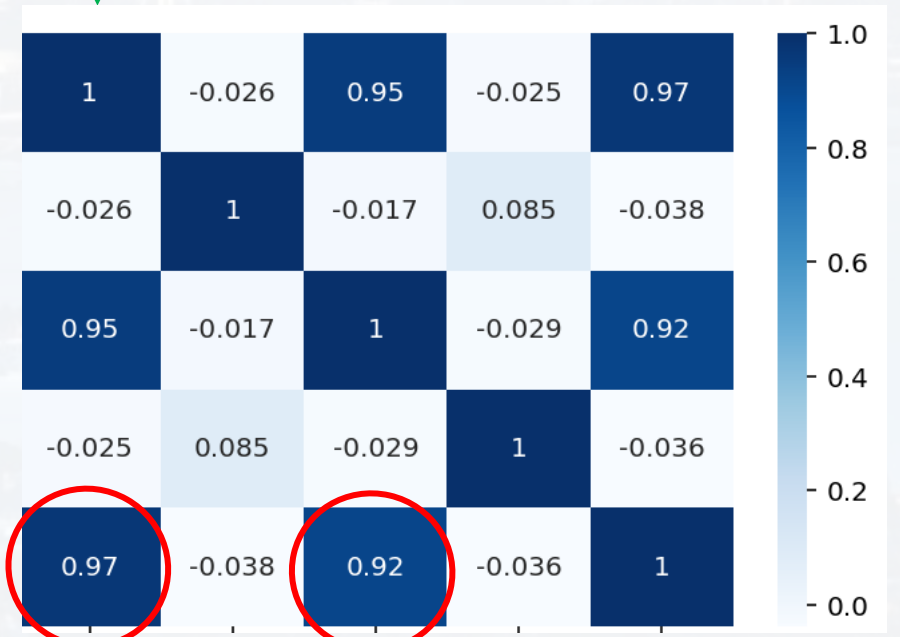
The Problem

some features correlate!

label	molecular_weight	electronegativity	bond_lengths	num_hydrogen_bonds	logP
Toxic	382.602	2.00269	3.61153	3	9.82666
Toxic	408.961	2.93626	3.47904	6	9.85889
Non-Toxic	239.548	2.71413	2.63922	8	6.75962
Non-Toxic	315.58	2.85598	2.86034	9	8.70674
Non-Toxic	282.521	2.83877	2.9664	1	7.8173



$$corr(x,y) = \frac{cov(x,y)}{\sqrt{var(x)var(y)}}$$

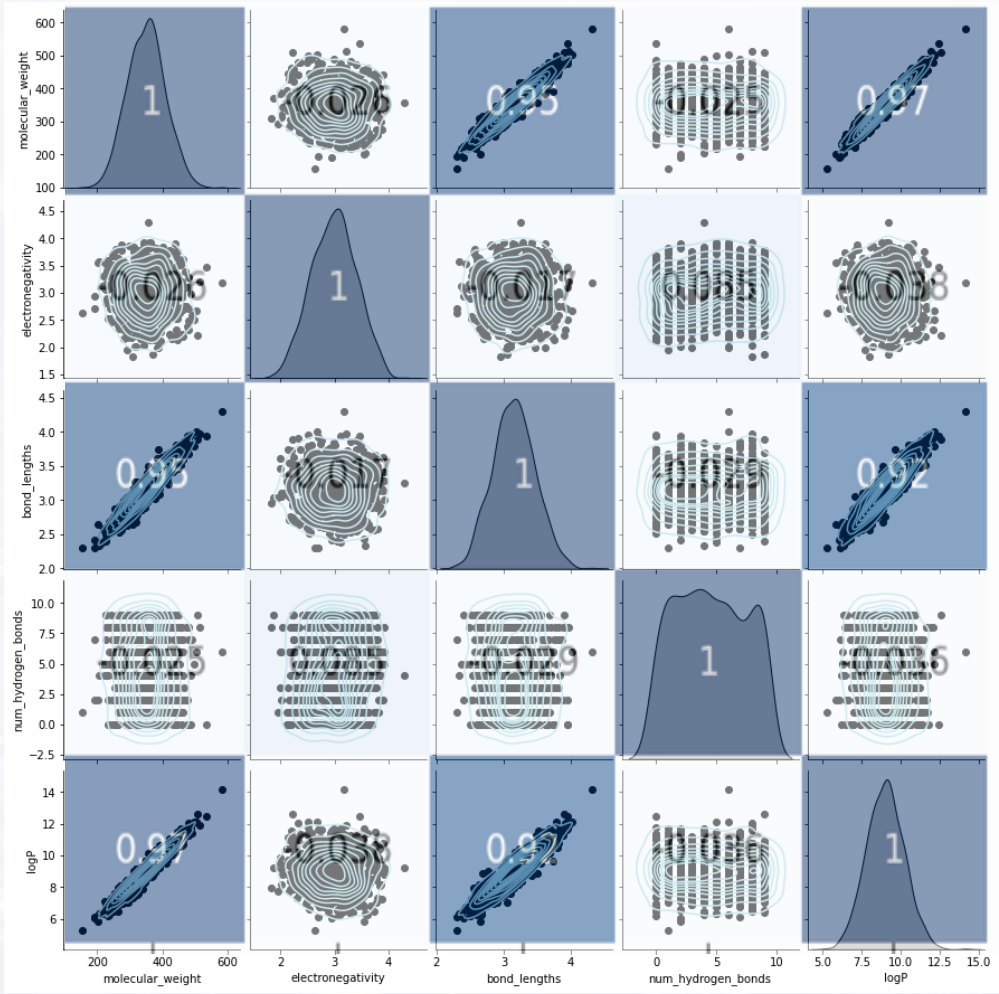


Dimension Reduction and PCA:

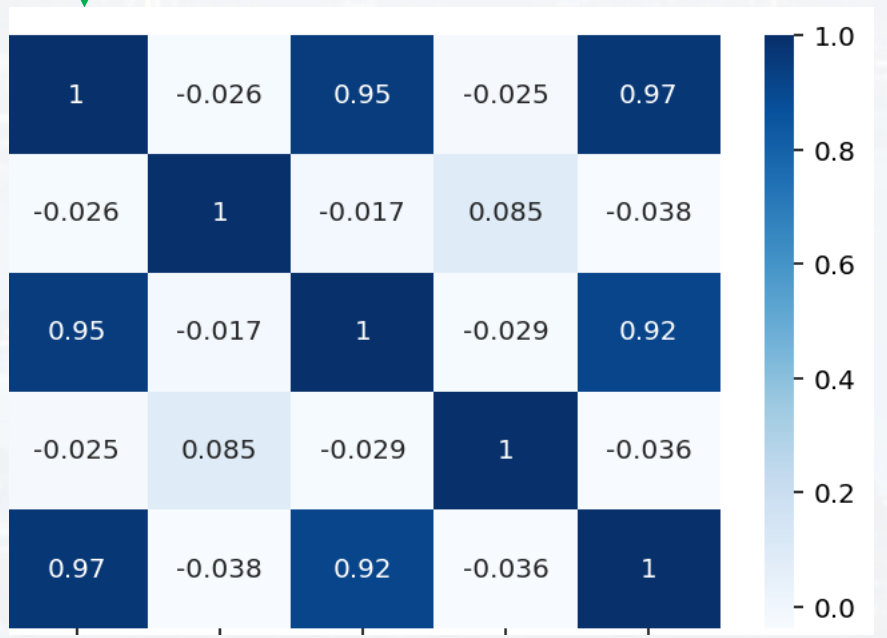
The Problem

some features correlate!

label	molecular_weight	electronegativity	bond_lengths	num_hydrogen_bonds	logP
Toxic	382.602	2.00269	3.61153	3	9.82666
Toxic	408.961	2.93626	3.47904	6	9.85889
Non-Toxic	239.548	2.71413	2.63922	8	6.75962
Non-Toxic	315.58	2.85598	2.86034	9	8.70674
Non-Toxic	282.521	2.83877	2.9664	1	7.8173



$$corr(x,y) = \frac{cov(x,y)}{\sqrt{var(x)var(y)}}$$



Dimension Reduction and PCA: The Problem

some features correlate!

label	molecular_weight	electronegativity	bond_lengths	num_hydrogen_bonds	logP
Toxic	382.602	2.00269	3.61153	3	9.82666
Toxic	408.961	2.93626	3.47904	6	9.85889
Non-Toxic	239.548	2.71413	2.63922	8	6.75962
Non-Toxic	315.58	2.85598	2.86034	9	8.70674
Non-Toxic	282.521	2.83877	2.9664	1	7.8173

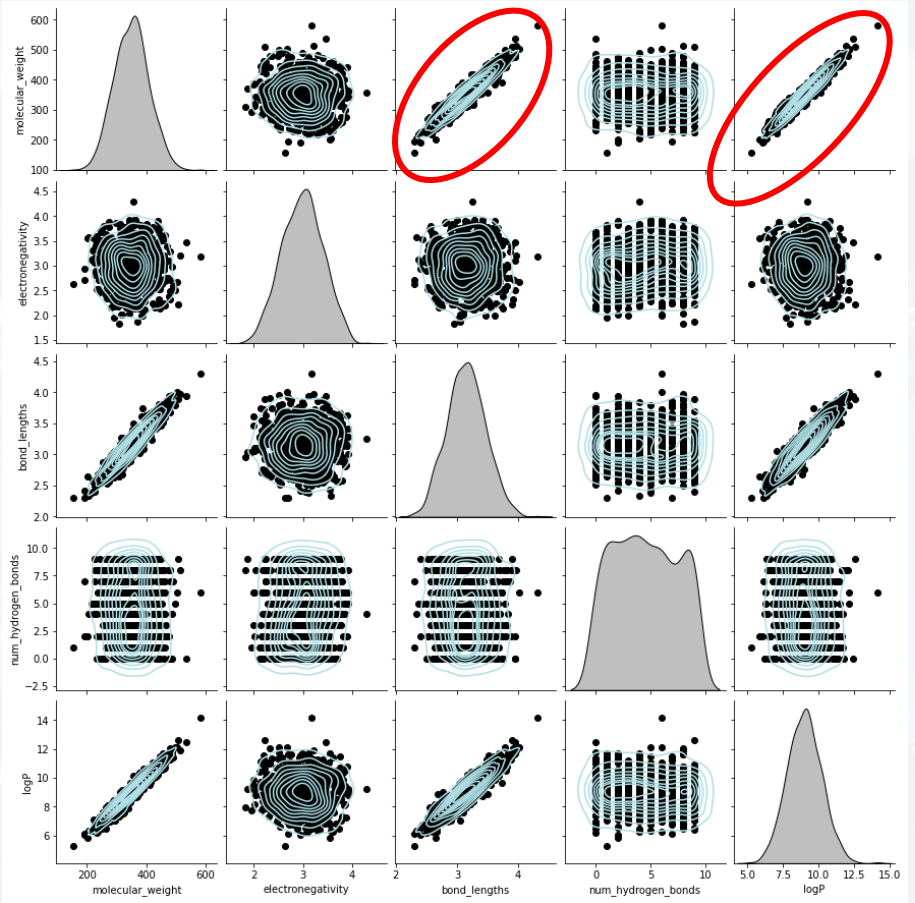
correlation means:

- features are **not mutually independent**

- we can predict feature ***a*** from feature ***b*** to some extend

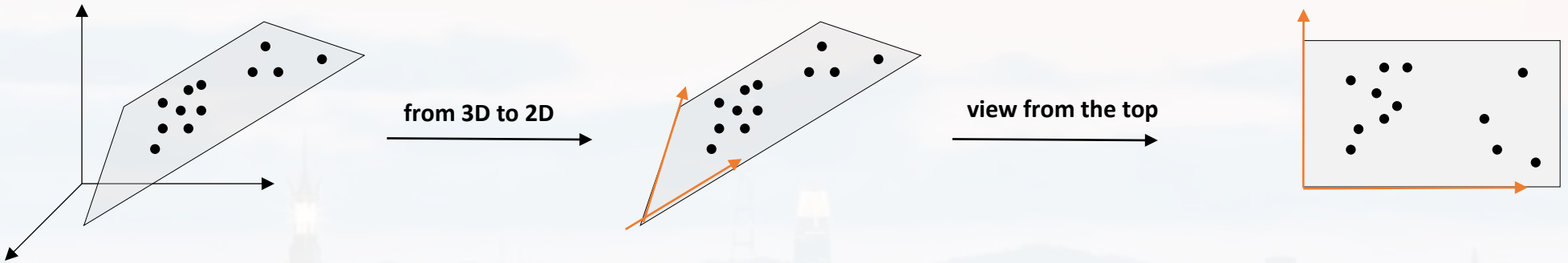
- we don't need all features

→ **reducing number of features** (dimensions) without losing information



Dimension Reduction and PCA: The Problem

some features correlate!



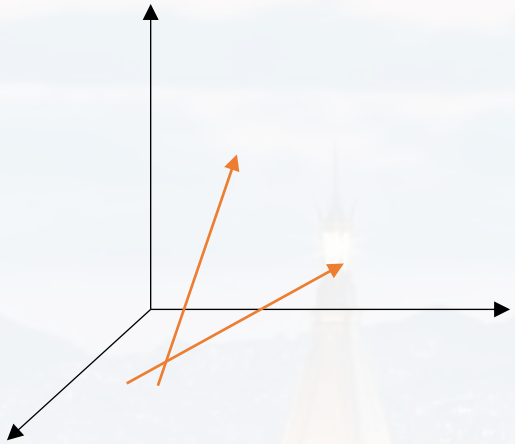
each data point is represented by **three** features...

… but those features correlate (x, y) → z

new coordinate system

Dimension Reduction and PCA: The Problem

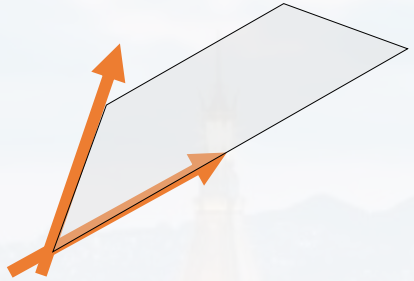
some features correlate!



Dimension Reduction and PCA:

The Problem

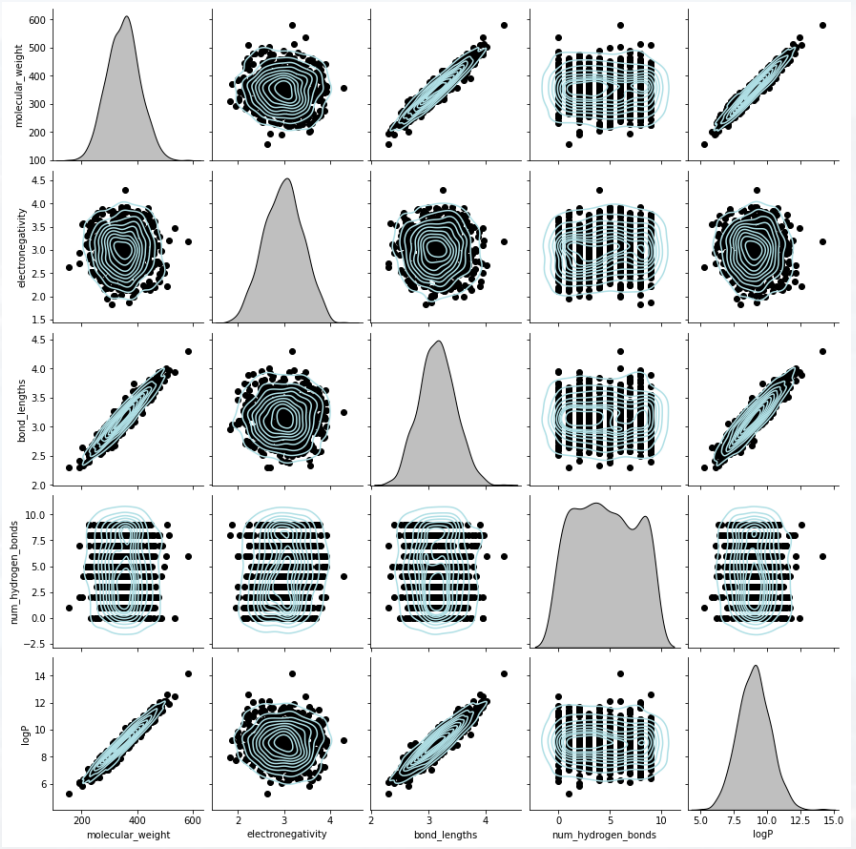
some features correlate!

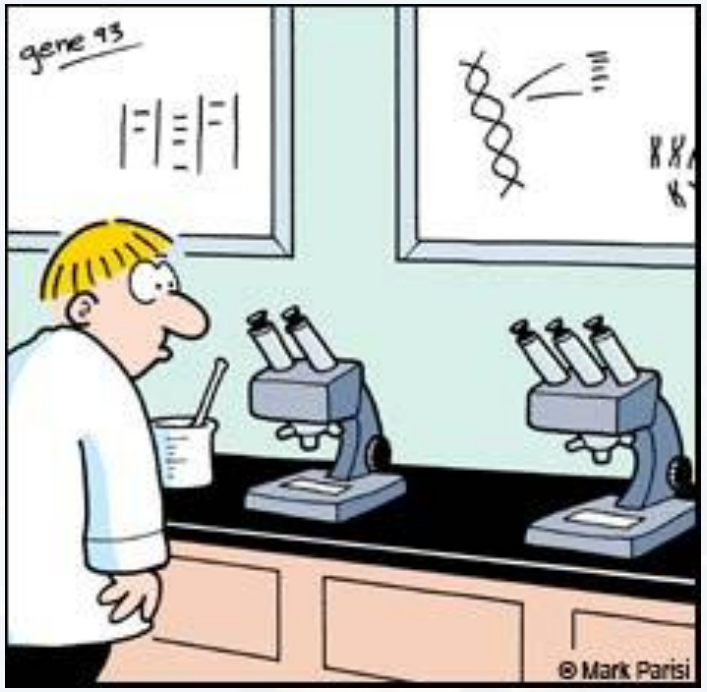


The **axis** of the **new coordinate** system are called **eigenvectors**

eigen: loosely translated from German “proper”

How do we find the eigenvectors based on correlation?





Outline

- The Problem
- Mathematical formulation of the Problem
- Examples

Dimension Reduction and PCA:

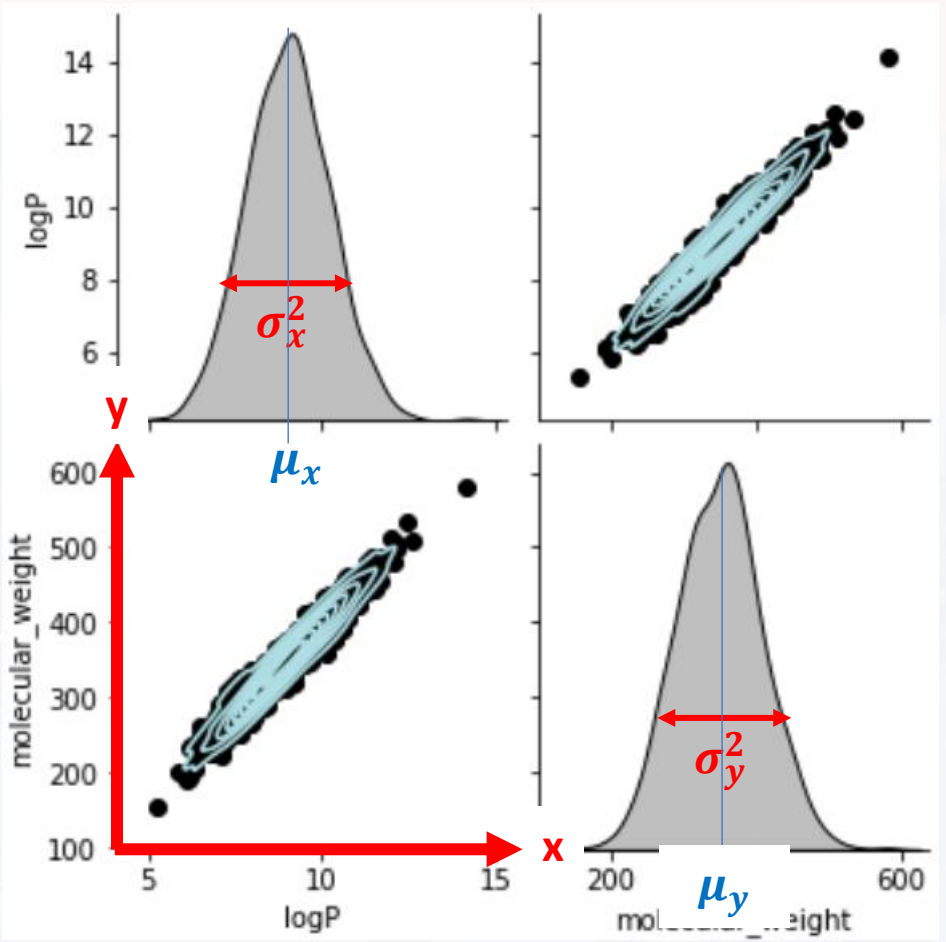
$$corr(x,y) := \frac{cov(x,y)}{\sqrt{var(x)var(y)}}$$

$$var(x) \equiv \sigma_x^2 := \sum_i^N (x_i - \mu_x)^2$$

$$cov(x,y) := \sum_j^M \sum_i^N (x_i - \mu_x)(y_j - \mu_y)$$

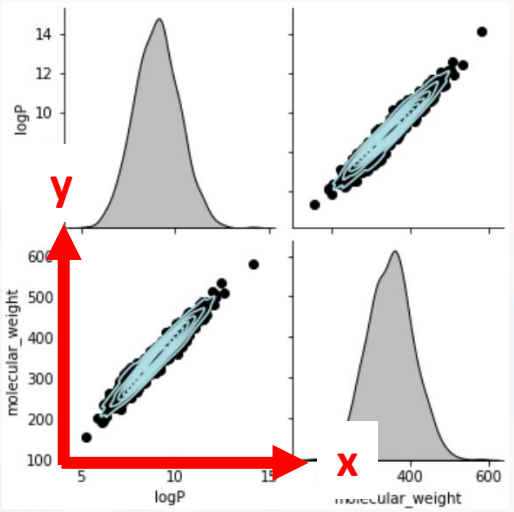
$$\sigma_{tot}^2 = \boxed{\sigma_x^2} + \boxed{\sigma_y^2} + \boxed{2\,cov(x,y)}$$

Let's try to remember this structure!

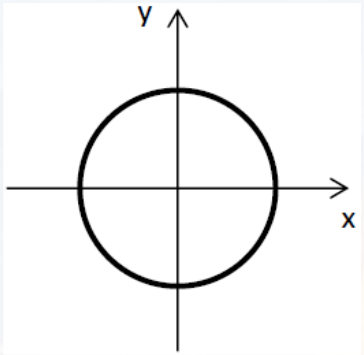


Dimension Reduction and PCA:

$$\sigma_{tot}^2 = \sigma_x^2 + \sigma_y^2 + 2\,cov(x,y)$$

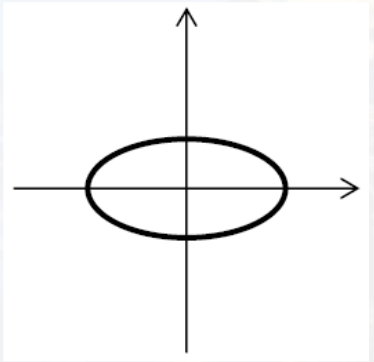


about cone sections:



$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = const$$

$$a = b \;\rightarrow\; x^2 + y^2 = r^2$$

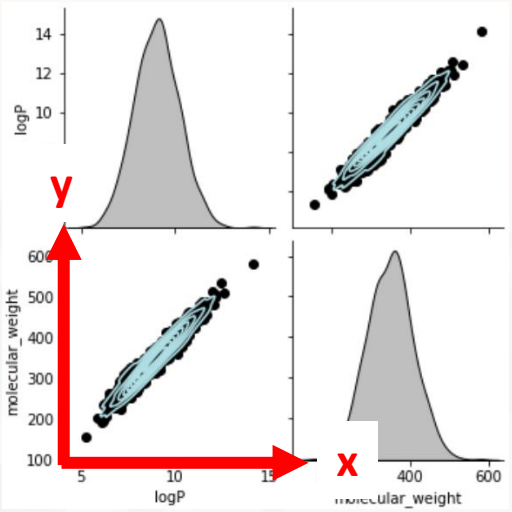


$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = const$$

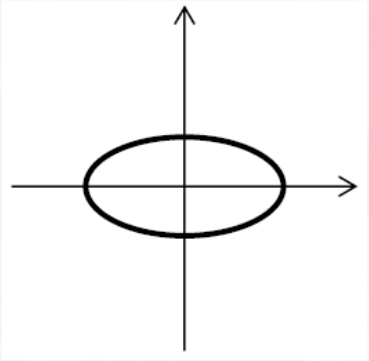
$$a \neq b$$

Dimension Reduction and PCA:

$$\sigma_{tot}^2 = \sigma_x^2 + \sigma_y^2 + 2\,cov(x,y)$$

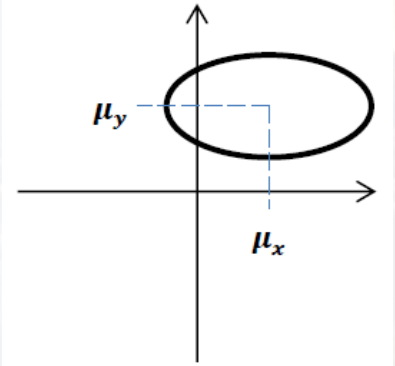


about cone sections:



$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = const$$

$$a \neq b$$

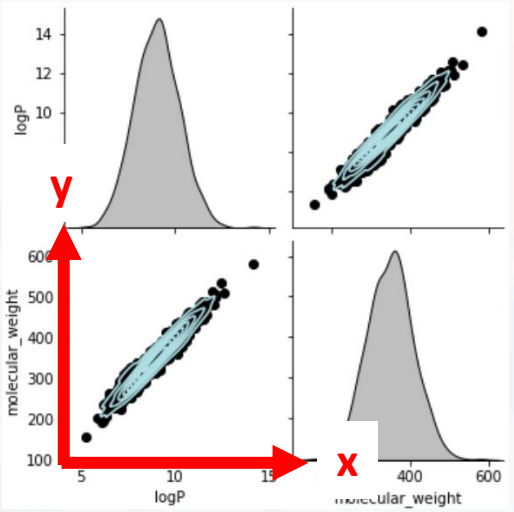


$$\frac{(x-\mu_x)^2}{a^2} + \frac{(y-\mu_y)^2}{b^2} = const$$

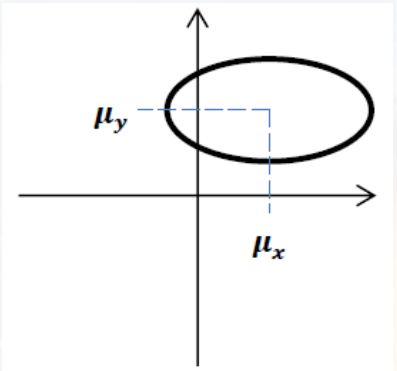
$$a \neq b$$

Dimension Reduction and PCA:

$$\sigma_{tot}^2 = \sigma_x^2 + \sigma_y^2 + 2\,cov(x,y)$$

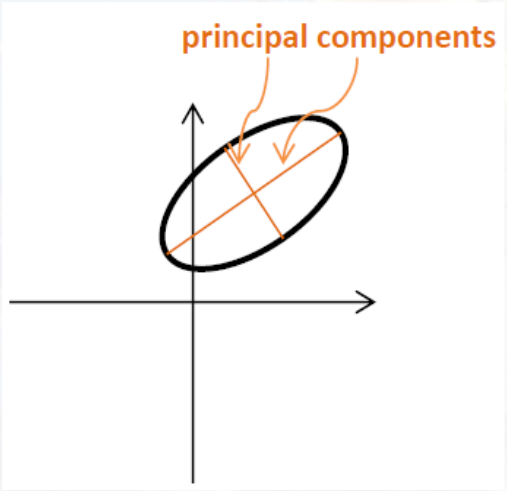


about cone sections:



$$\frac{(x-\mu_x)^2}{a^2} + \frac{(y-\mu_y)^2}{b^2} = const$$

$$a \neq b$$



$$\frac{(x-\mu_x)^2}{a^2} + \frac{(y-\mu_y)^2}{b^2} + 2\,c(x-\mu_x)(y-\mu_y) = const$$

$$a \neq b$$

Dimension Reduction and PCA:

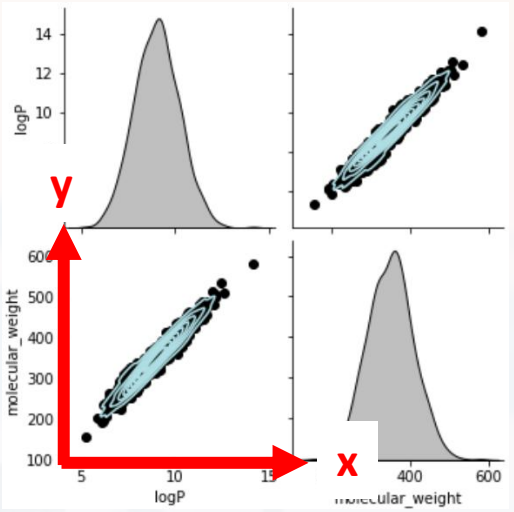
The Math

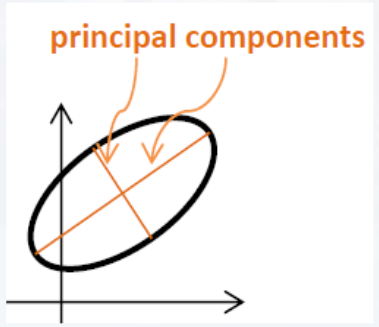
$$\sigma_{tot}^2 = \sigma_x^2 + \sigma_y^2 + 2\,cov(x,y)$$

$$= \sum_i^N (x_i - \mu_x)^2 + \sum_j^M (y_j - \mu_y)^2 + 2\sum_j^M \sum_i^N (x_i - \mu_x)(y_j - \mu_y)$$

$$const = \frac{(x-\mu_x)^2}{a^2} + \frac{(y-\mu_y)^2}{b^2} + 2\,c(x-\mu_x)(y-\mu_y)$$

$$const = \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}^T \begin{pmatrix} 1/a^2 & c \\ c & 1/b^2 \end{pmatrix} \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}$$





more general:

$$const = \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}^T \begin{pmatrix} \alpha & \gamma_{12} \\ \gamma_{21} & \beta \end{pmatrix} \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}$$

covariance matrix

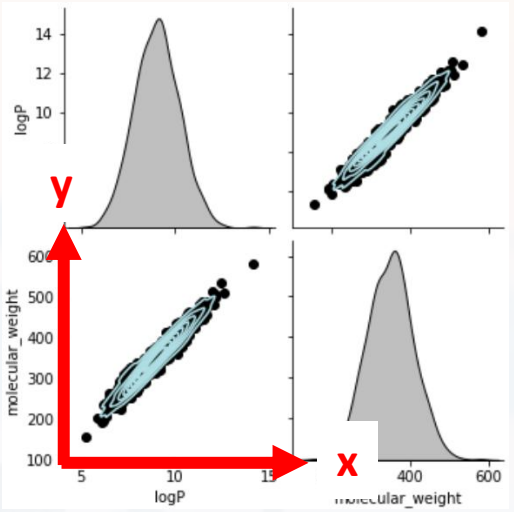
$$= v^T S v$$ ... called **quadratic form** (also in N-D)

Dimension Reduction and PCA:

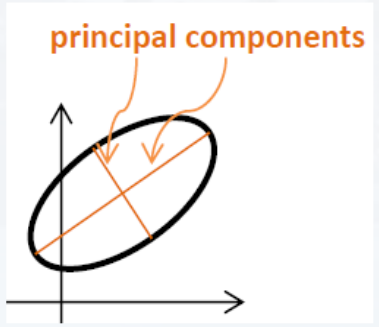
The Math

$$\sigma_{tot}^2 = \sigma_x^2 + \sigma_y^2 + 2\,cov(x,y)$$

$$= \sum_i^N (x_i - \mu_x)^2 + \sum_j^M (y_j - \mu_y)^2 + 2\sum_j^M \sum_i^N (x_i - \mu_x)(y_j - \mu_y)$$



$$const = \frac{(x-\mu_x)^2}{a^2} + \frac{(y-\mu_y)^2}{b^2} + 2\,c(x-\mu_x)(y-\mu_y)$$



$$const = \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}^T \begin{pmatrix} 1/a^2 & c \\ c & 1/b^2 \end{pmatrix} \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}$$

more general:

$$const = \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}^T \begin{pmatrix} \alpha & \gamma_{12} \\ \gamma_{21} & \beta \end{pmatrix} \begin{pmatrix} x-\mu_x \\ y-\mu_y \end{pmatrix}$$

$$= v^T S v$$ ... called **quadric** (also in N-D)

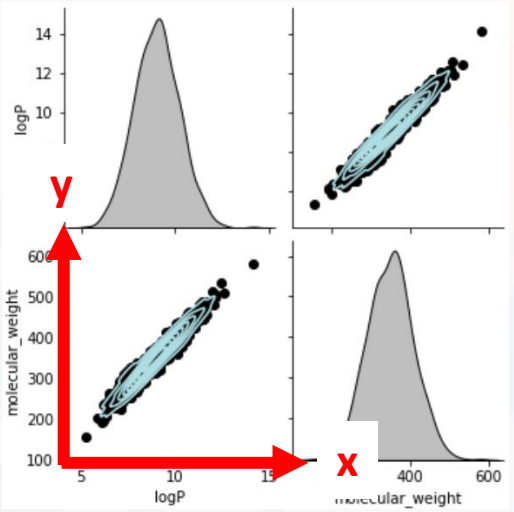
Dimension Reduction and PCA:

The Math

$$\sigma_{tot}^2 = \sigma_x^2 + \sigma_y^2 + 2\,cov(x,y)$$

$$const = \frac{(x-\mu_x)^2}{a^2} + \frac{(y-\mu_y)^2}{b^2} + 2\,c(x-\mu_x)(y-\mu_y)$$

$$const = \begin{pmatrix} \boldsymbol{x}-\boldsymbol{\mu_x} \\ \boldsymbol{y}-\boldsymbol{\mu_y} \end{pmatrix}^T \begin{pmatrix} 1/a^2 & c \\ c & 1/b^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}-\boldsymbol{\mu_x} \\ \boldsymbol{y}-\boldsymbol{\mu_y} \end{pmatrix}$$



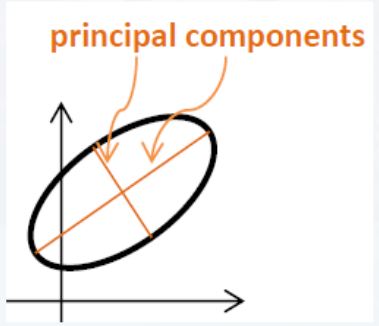
- geometrically, the **covariance matrix** can be interpreted as quadratic form
- the covariances are the **non-diagonal** elements of the **covariance matrix**
- aim: finding a coordinate transformation, where the **covariance matrix** is diagonal

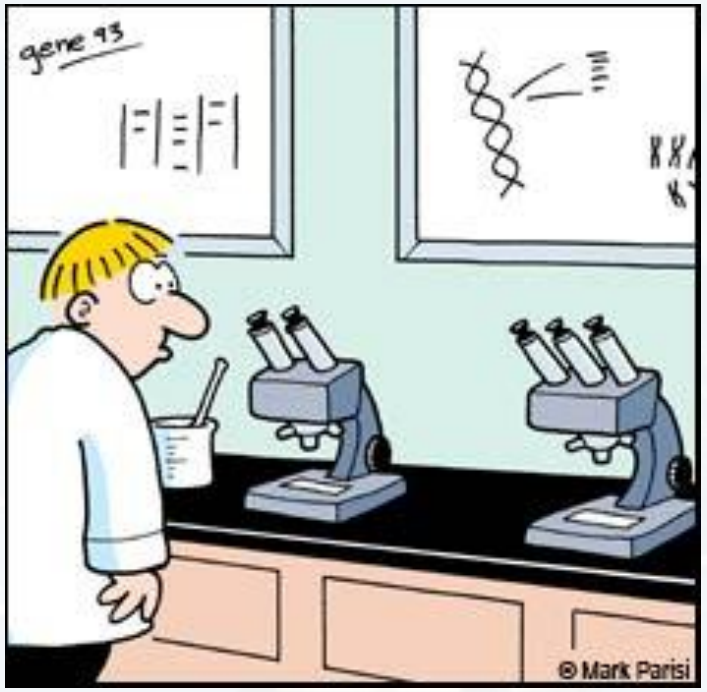
$$\begin{pmatrix} \lambda_1 & \ddots & 0 & \dots & 0 \\ 0 & & \lambda_i & \ddots & 0 \\ 0 & & 0 & & \lambda_N \end{pmatrix}$$

the diagonal entries are called eigenvalues (= variances in new coordinate system)

→ all variables **are independent**

→ principal components of the **covariance matrix** are **parallel** to the **new coordinate axes (= eigenvectors)**





Outline

- The Problem
- Mathematical formulation of the Problem
- Examples

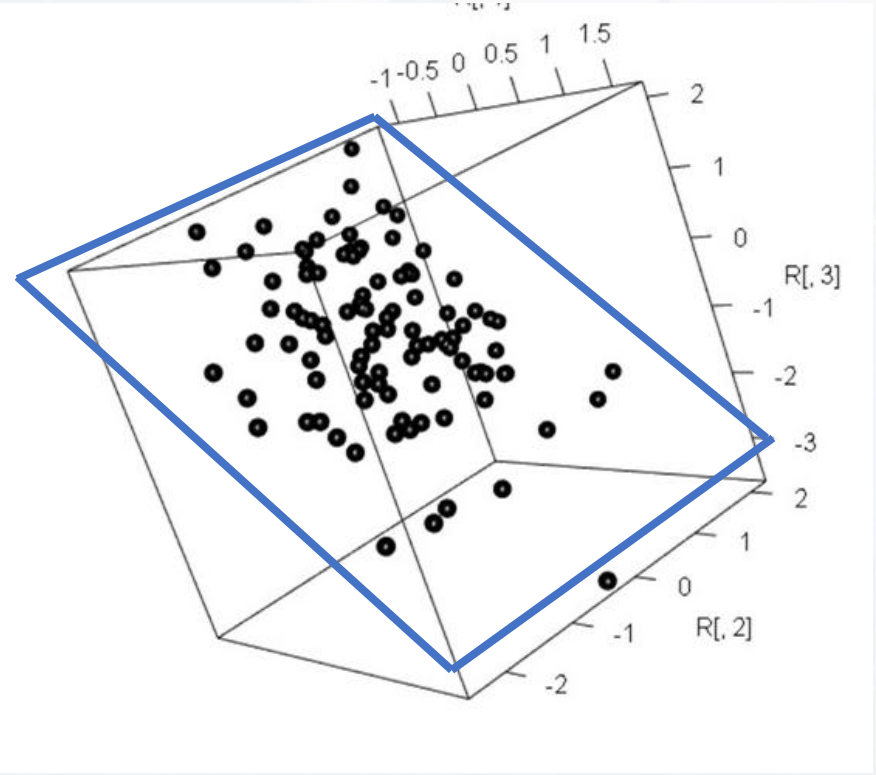
Dimension Reduction and PCA:


```
from sklearn.decomposition import PCA
```

Let us take a look at some artificial data first:

see **PCA_simple.ipynb**

- 3D data cloud
- however, all data points seem to be located on **one plane**
- PCA should be able to **reduce dimensions**



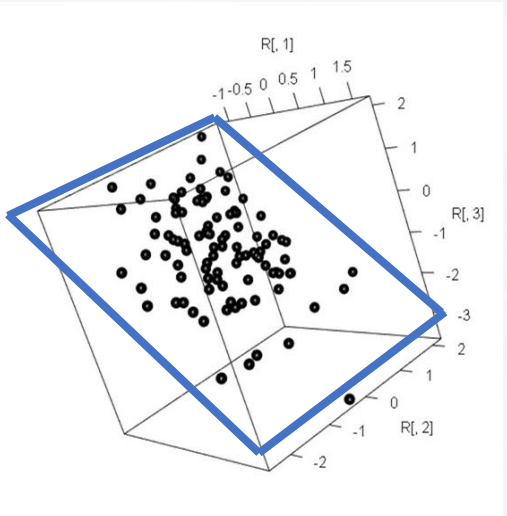
Dimension Reduction and PCA:


```
from sklearn.decomposition import PCA
```

Let us take a look at some artificial data first:

- 3D data cloud
- however, all data points seem to be located on **one plane**
- PCA should be able to **reduce dimensions**

```
XYZ = pd.read_csv('Rot.txt', delim_whitespace = True,\
                  header = None)
XYZ = np.array(XYZ)

fig = plt.figure(figsize = (12, 12))
ax = fig.add_subplot(projection = '3d')
ax.scatter(XYZ[:,0], XYZ[:,1], XYZ[:,2], c = 'black',\
           marker = 'o', s = 40)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.tick_params(axis = 'both', which = 'major', labelsize = 30)
plt.show()
```



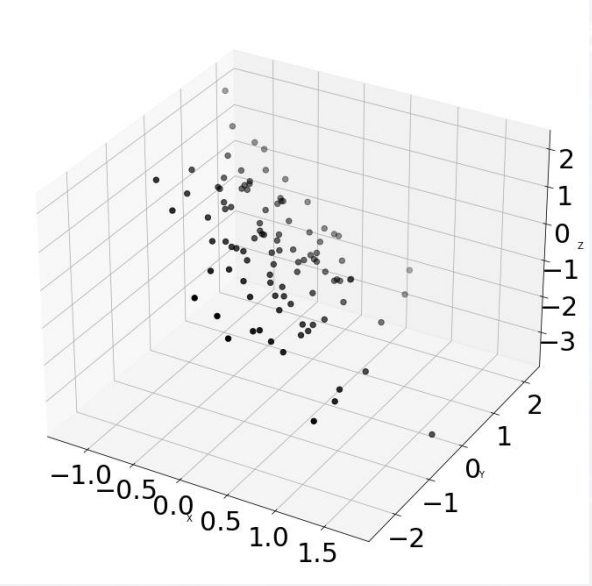
Dimension Reduction and PCA:


```
from sklearn.decomposition import PCA
```

Let us take a look at some artificial data first:

- 3D data cloud
- however, all data points seem to be located on **one plane**
- PCA should be able to **reduce dimensions**

```
XYZ = pd.read_csv('Rot.txt', delim_whitespace = True,\
                  header = None)
XYZ = np.array(XYZ)

fig = plt.figure(figsize = (12, 12))
ax = fig.add_subplot(projection = '3d')
ax.scatter(XYZ[:,0], XYZ[:,1], XYZ[:,2], c = 'black',\
           marker = 'o', s = 40)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.tick_params(axis = 'both', which = 'major', labelsize = 30)
plt.show()
```



Dimension Reduction and PCA:

performing the actual PCA:

```
out = PCA(n_components = 3).fit(XYZ)

eigenVec = out.components_
eigenVal = out.explained_variance_
eigenXYZ = out.transform(XYZ)
```

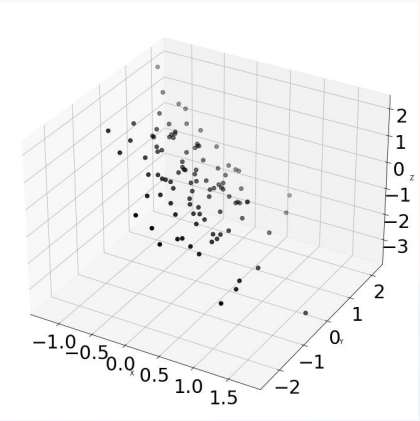
- 3D data cloud
- however, all data points seem to be located on **one plane**
- PCA should be able to **reduce dimensions**

plotting the eigenvalue spectrum:

```
xplot = np.arange(1,4)

plt.bar(xplot, eigenVal, color = (0.8, 0.8, 0.8), edgecolor = 'black')
plt.xlabel('dimension')
plt.ylabel('eigenvalue')
plt.yscale('log')
plt.xticks(xplot)
plt.show()
```

Dimension Reduction and PCA:

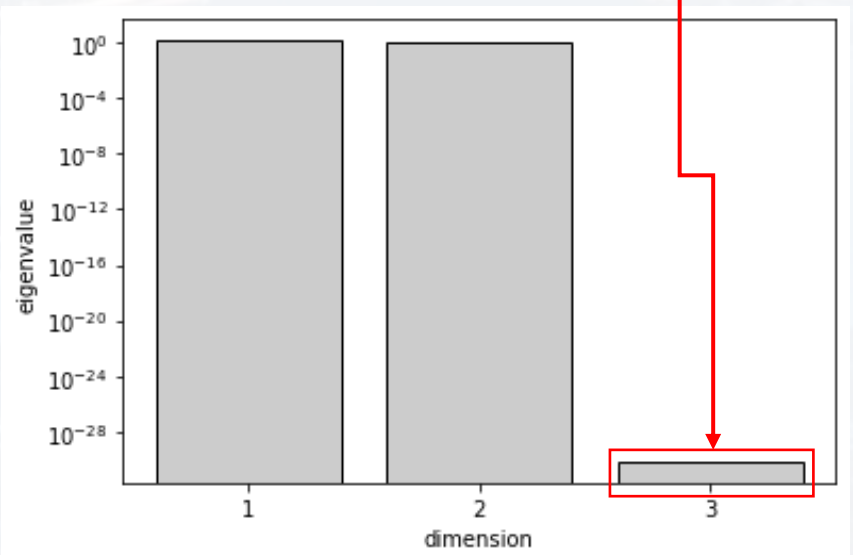

```
out = PCA(n_components = 3).fit(XYZ)

eigenVec = out.components_
eigenVal = out.explained_variance_
eigenXYZ = out.transform(XYZ)
```

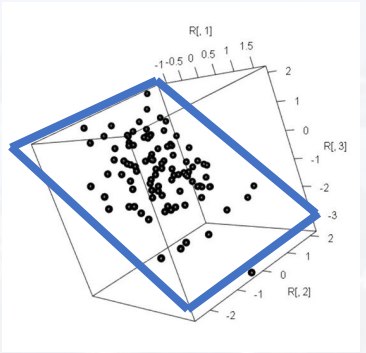


plotting the eigenvalue spectrum:

```
xplot = np.arange(1,4)

plt.bar(xplot, eigenVal, color = (0.8, 0.8, 0.8), edgecolor = 'black')
plt.xlabel('dimension')
plt.ylabel('eigenvalue')
plt.yscale('log')
plt.xticks(xplot)
plt.show()
```

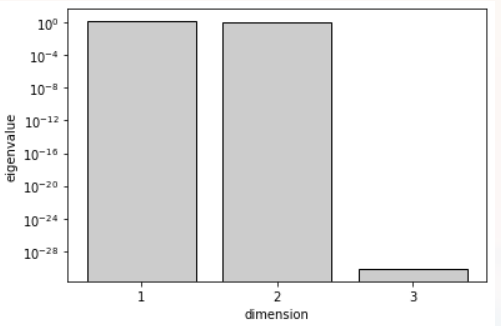
one eigenvalue is zero





Dimension Reduction and PCA:

plotting the eigenvalue spectrum:



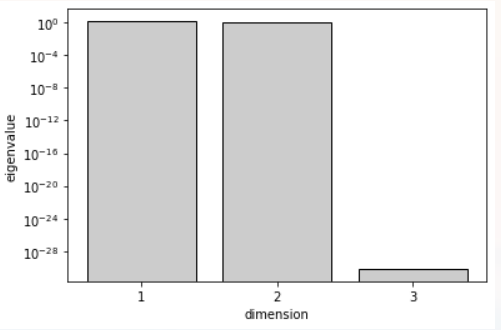

```
xplot = np.arange(1,4)

plt.bar(xplot, eigenVal, color = (0.8, 0.8, 0.8), edgecolor = 'black')
plt.xlabel('dimension')
plt.ylabel('eigenvalue')
plt.yscale('log')
plt.xticks(xplot)
plt.show()
```

```
fig = plt.figure(figsize = (12, 12))
ax = fig.add_subplot(projection = '3d')
ax.scatter(eigenXYZ[:,0], eigenXYZ[:,1], eigenXYZ[:,2], c = 'black', \
           marker = 'o', s = 40)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.tick_params(axis = 'both', which = 'major', labelsize = 30)
plt.show()
```

Dimension Reduction and PCA:

Examples

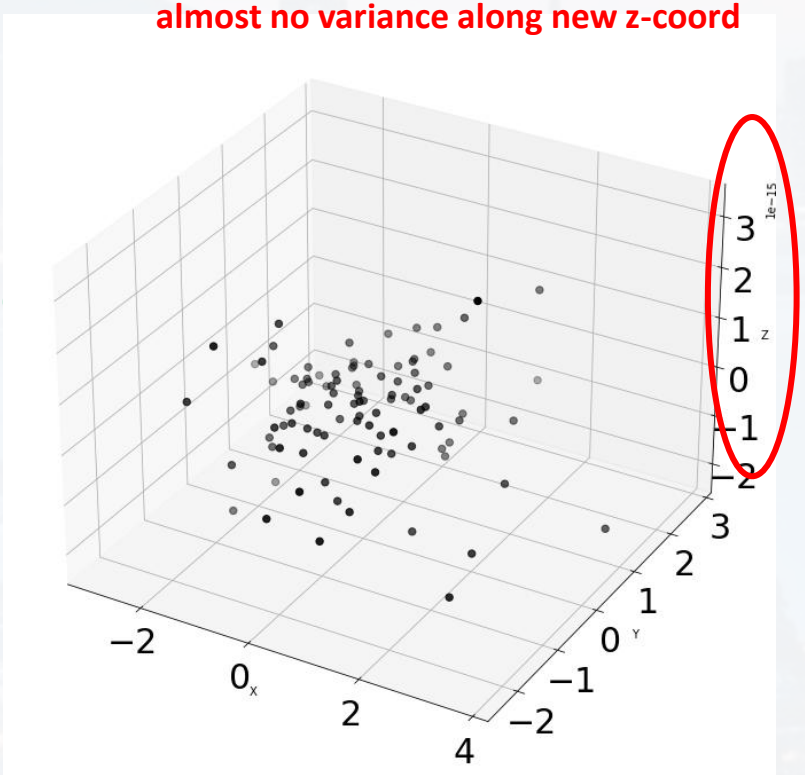
plotting the eigenvalue spectrum:




```
xplot = np.arange(1,4)

plt.bar(xplot, eigenVal, color = (0.8, 0.8, 0.8), edgecolor = 'black')
plt.xlabel('dimension')
plt.ylabel('eigenvalue')
plt.yscale('log')
plt.xticks(xplot)
plt.show()
```

```
fig = plt.figure(figsize = (12, 12))
ax = fig.add_subplot(projection = '3d')
ax.scatter(eigenXYZ[:,0], eigenXYZ[:,1], eigenXYZ[:,2], c = 'bl
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.tick_params(axis = 'both', which = 'major', labelsize = 30)
plt.show()
```

almost no variance along new z-coord



check also eg:

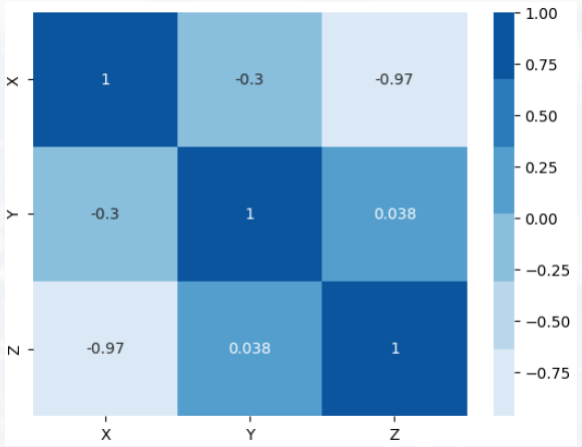
```
np.dot(eigenVec[:,0],eigenVec[:,1])
```

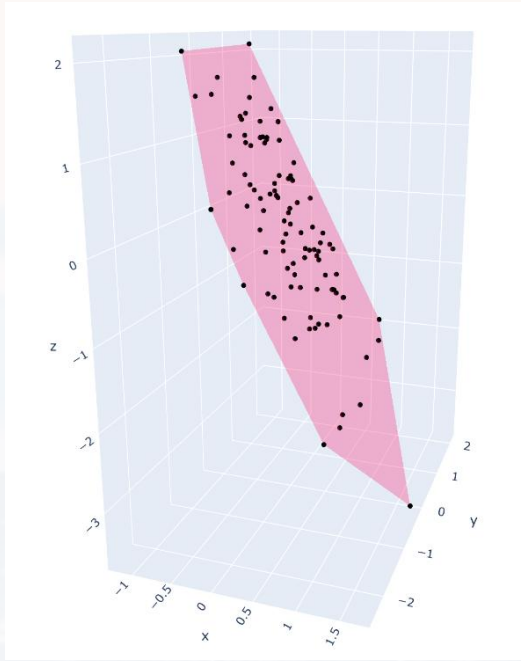
Dimension Reduction and PCA:

Examples

Summary:

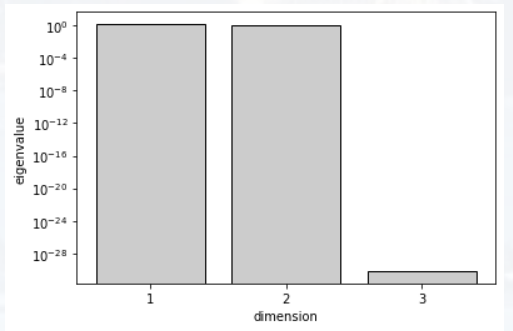
- We don’t need **three** coordinates in order to describe the data points
  → some of the directions (features) correlate



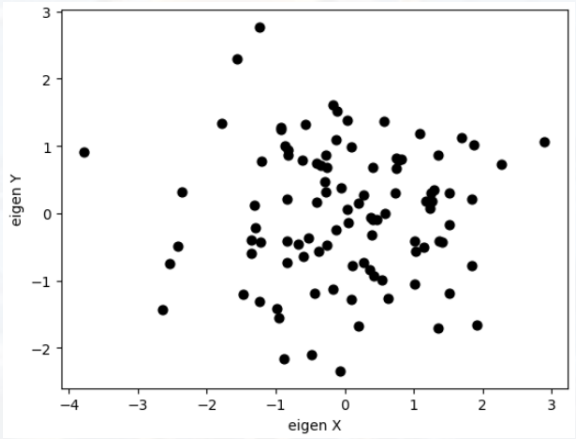


- running a PCA in order to find the proper coordinate system

- **one** of **three** eigenvalues is a lot smaller than the other **two**



→ We only need **two** coordinates for the data set



We can reduce the complexity
of the data set without loosing information

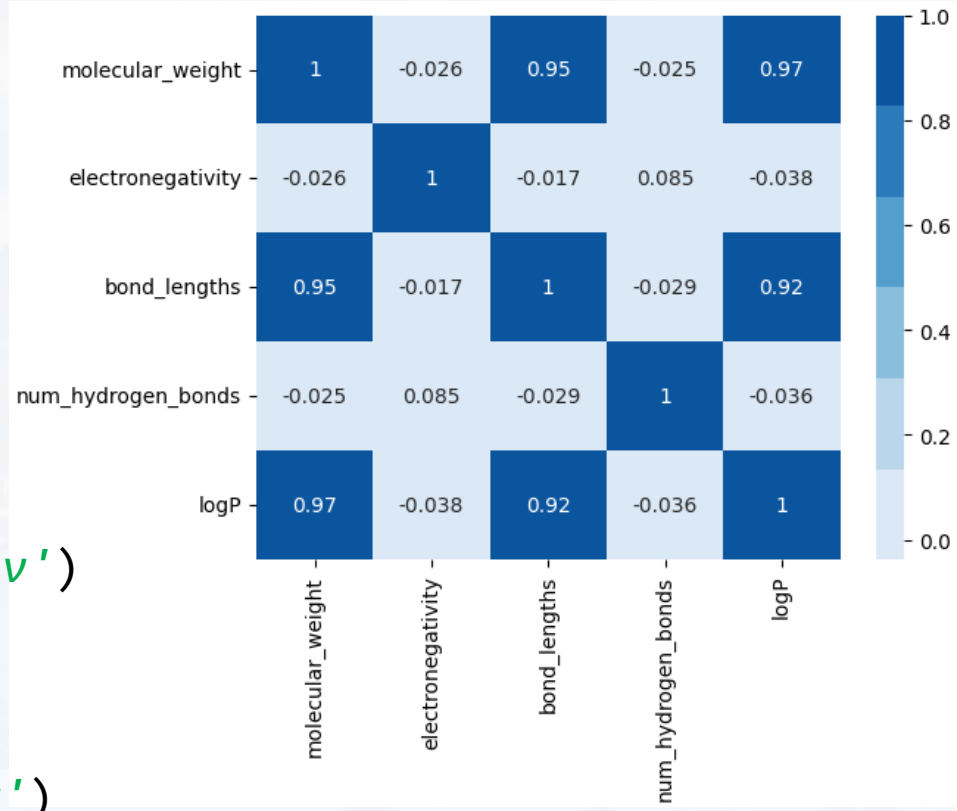
Dimension Reduction and PCA: Examples

let us return to the molecule data set now:

$$corr(x,y) = \frac{cov(x,y)}{\sqrt{var(x)var(y)}}$$

see **NaiveBayes_PCA.ipynb**

label	molecular_weight	electronegativity	bond_lengths	num_hydrogen_bonds	logP
Toxic	382.602	2.00269	3.61153	3	9.82666
Toxic	408.961	2.93626	3.47904	6	9.85889
Non-Toxic	239.548	2.71413	2.63922	8	6.75962
Non-Toxic	315.58	2.85598	2.86034	9	8.70674
Non-Toxic	282.521	2.83877	2.9664	1	7.8173




```
Train  = pd.read_csv('molecular_train_gbc.csv')
TrainY = Train['label']
TrainX = Train.drop('label', axis = 1)

Test   = pd.read_csv('molecular_test_gbc.csv')
TestY  = Test['label']
TestX  = Test.drop('label', axis = 1)
```

Dimension Reduction and PCA:

let us return to the molecule data set now:

```
from sklearn.preprocessing import MinMaxScaler

scaler   = MinMaxScaler(feature_range = (0, 1))

TrainXS = scaler.fit_transform(TrainX)
TestXS  = scaler.transform(TestX)

out = PCA(n_components = 5).fit(TrainXS)

eigenVec    = out.components_
eigenVal    = out.explained_variance_
eigenTrainX = out.transform(TrainXS)
```

important: scaling and normalization

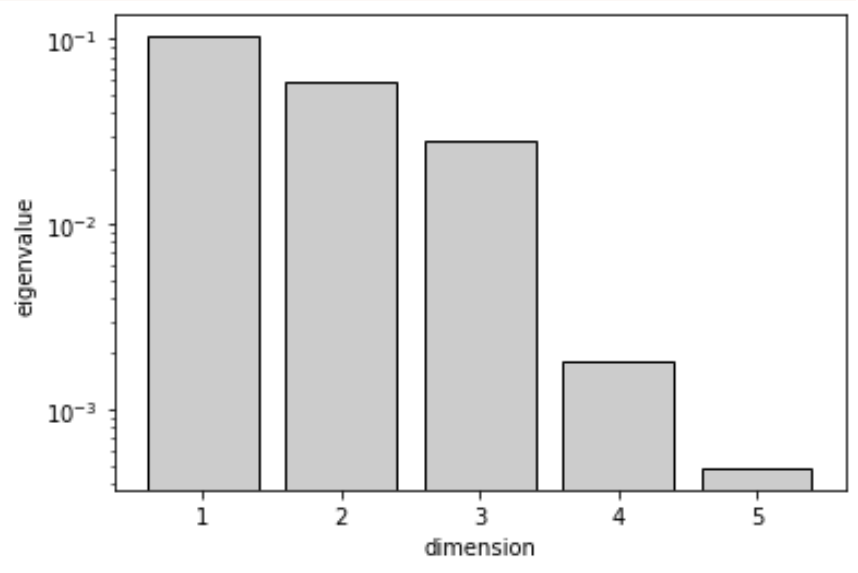
creating our model with .fit

Finally transforming the data into eigencoordinates

Dimension Reduction and PCA:

let us return to the molecule data set now:

```
out = PCA(n_components = 5).fit(TrainXS)

eigenVec    = out.components_
eigenVal    = out.explained_variance_
eigenTrainX = out.transform(TrainXS)

xplot = np.arange(1,6)
```




```
plt.bar(xplot, eigenVal, color = (0.8, 0.8, 0.8), edgecolor = 'black')
plt.xlabel('dimension')
plt.ylabel('eigenvalue')
plt.yscale('log')
plt.xticks(xplot)
plt.show()
```

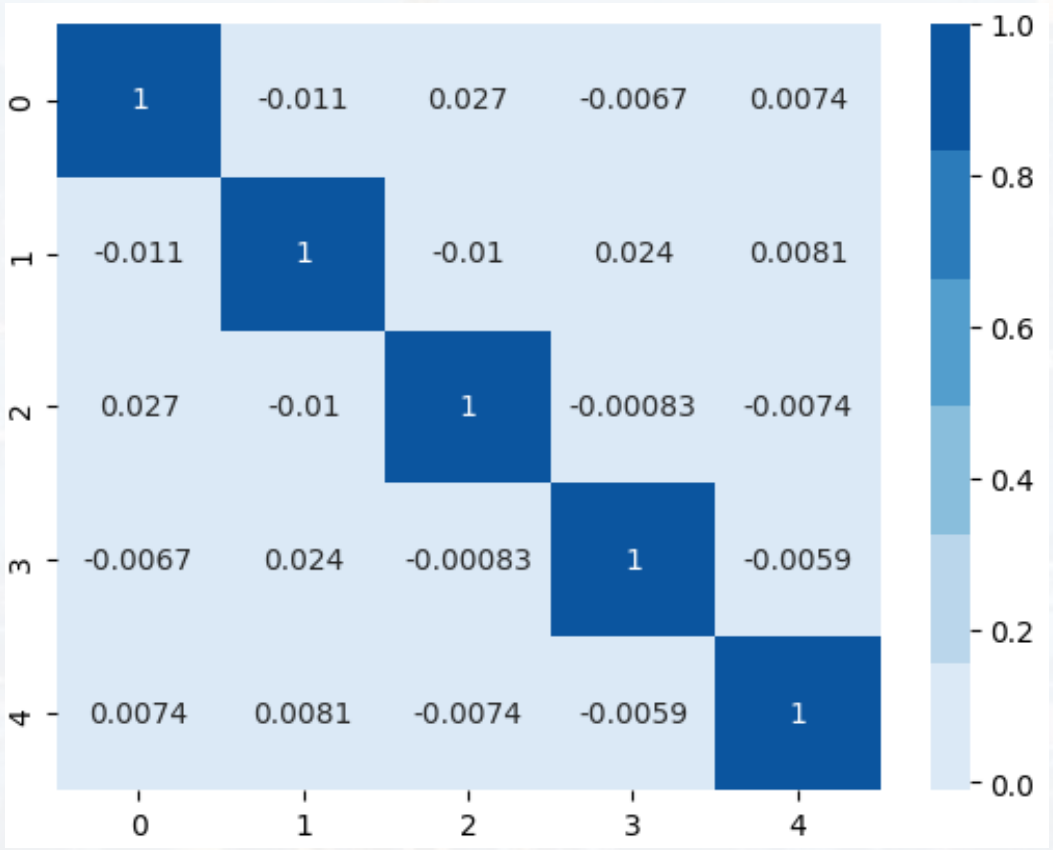
Dimension Reduction and PCA:

Examples

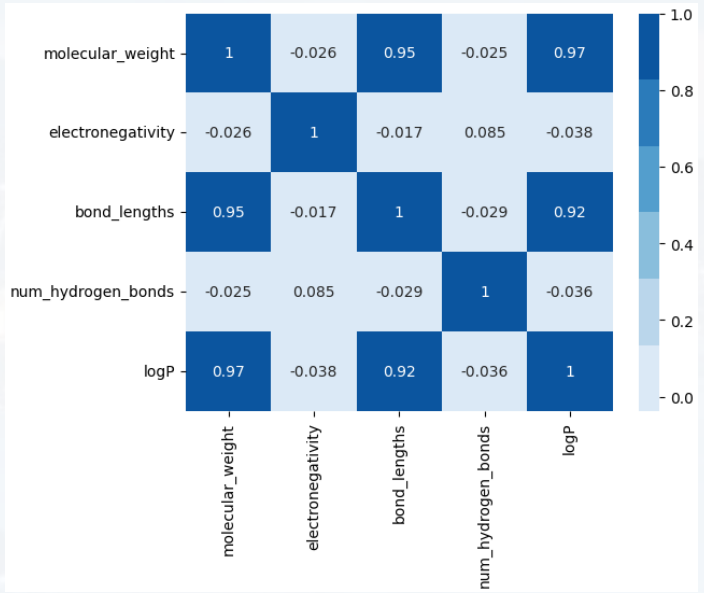
let us return to the molecule data set now:

```
sns.heatmap(pd.DataFrame(eigenTrainX).corr(), annot = True,
                                    cmap = color_palette("Blues"))
plt.show()
```

after PCA:



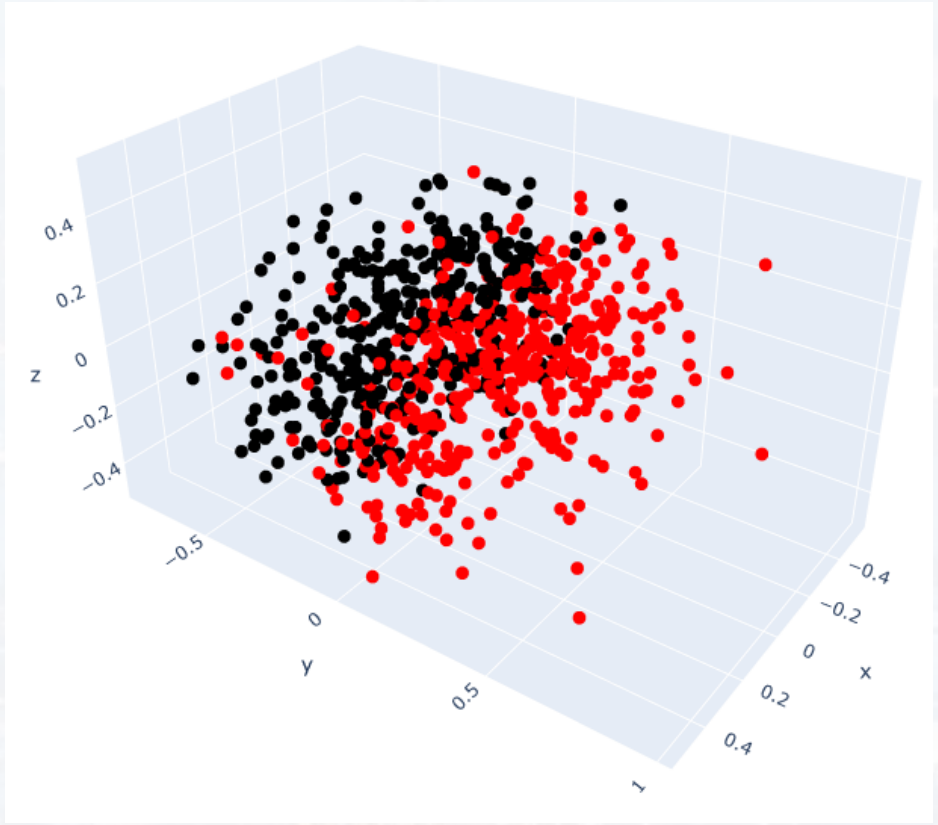
no PCA:

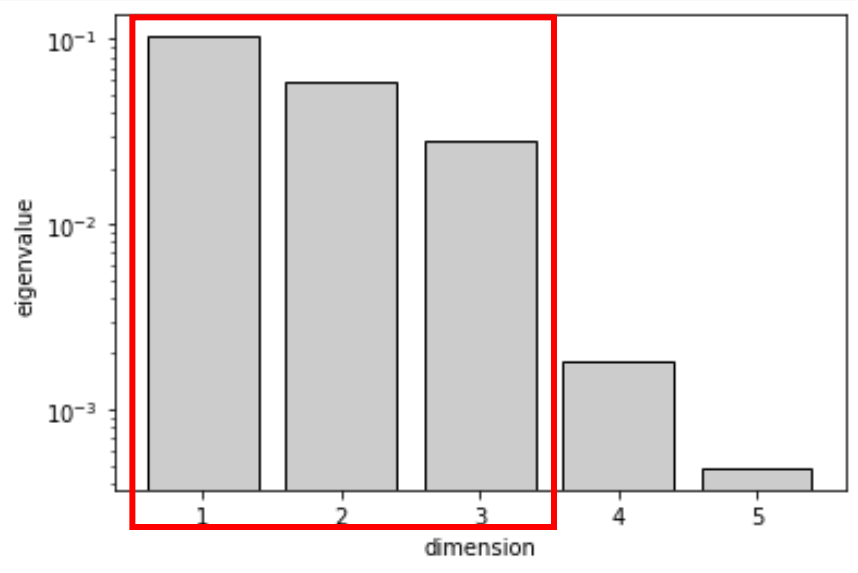


Dimension Reduction and PCA:

let us return to the molecule data set now:

We also need only **three** directions now and therefore can create a normal scatter plot:

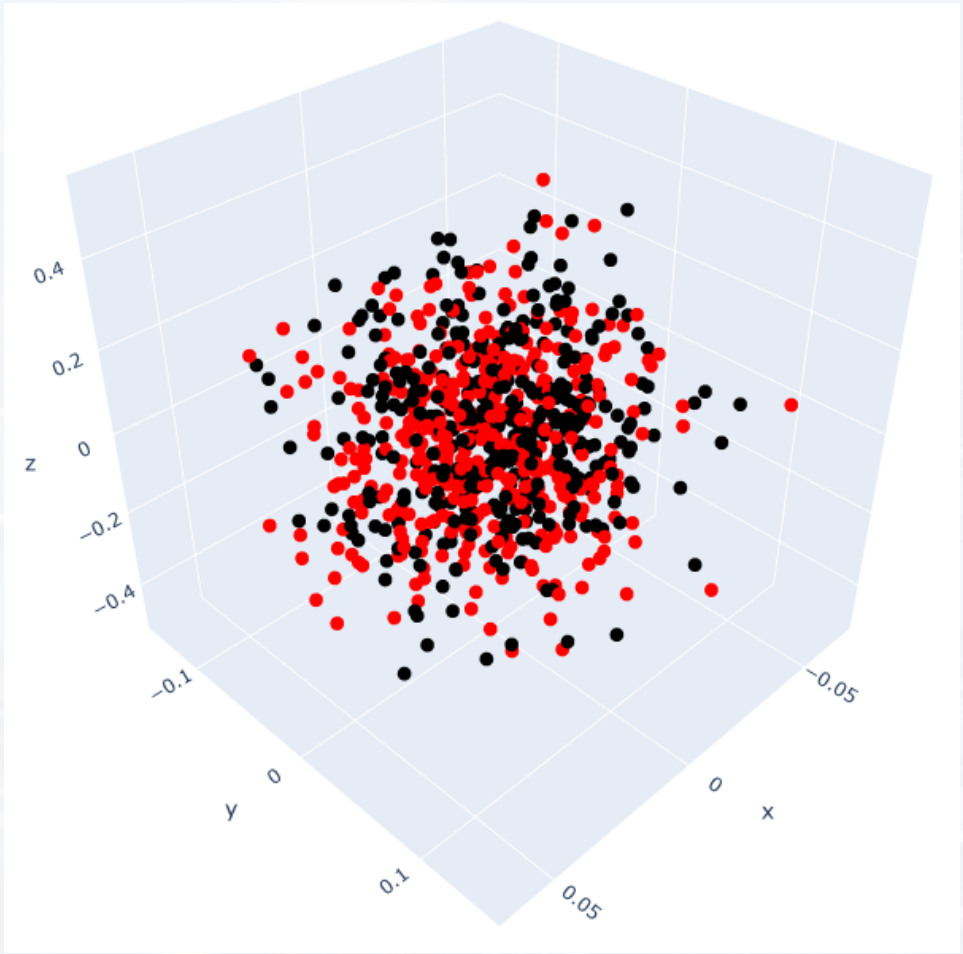


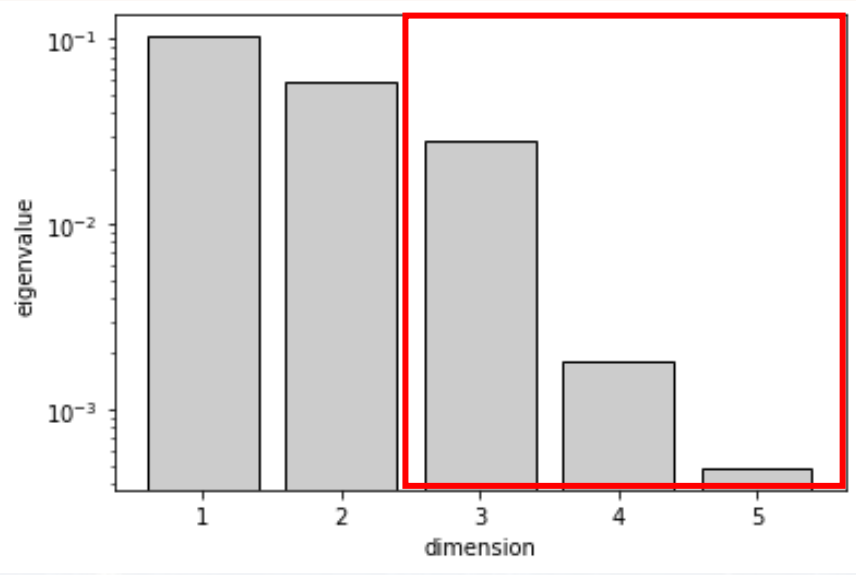


Dimension Reduction and PCA:

let us return to the molecule data set now:

as a consistency check: plotting the **three** directions with ***lowest*** eigenvalues





Dimension Reduction and PCA: Examples

Classification Naïve Bayes

$$k_{new} = \underset{k}{argmax} \left\{ P(C_k) \prod_{i=1}^{I} P(x_i|C_k) \right\}$$

no PCA (five features):

```
GaussianNB:    accuracy = 81%
```

after PCA (three features):

```
GaussianNB:    accuracy = 83%
```

Thank you very much for your attention!