Dimensionality Reduction

B063012054 林祐安

CONTENTS

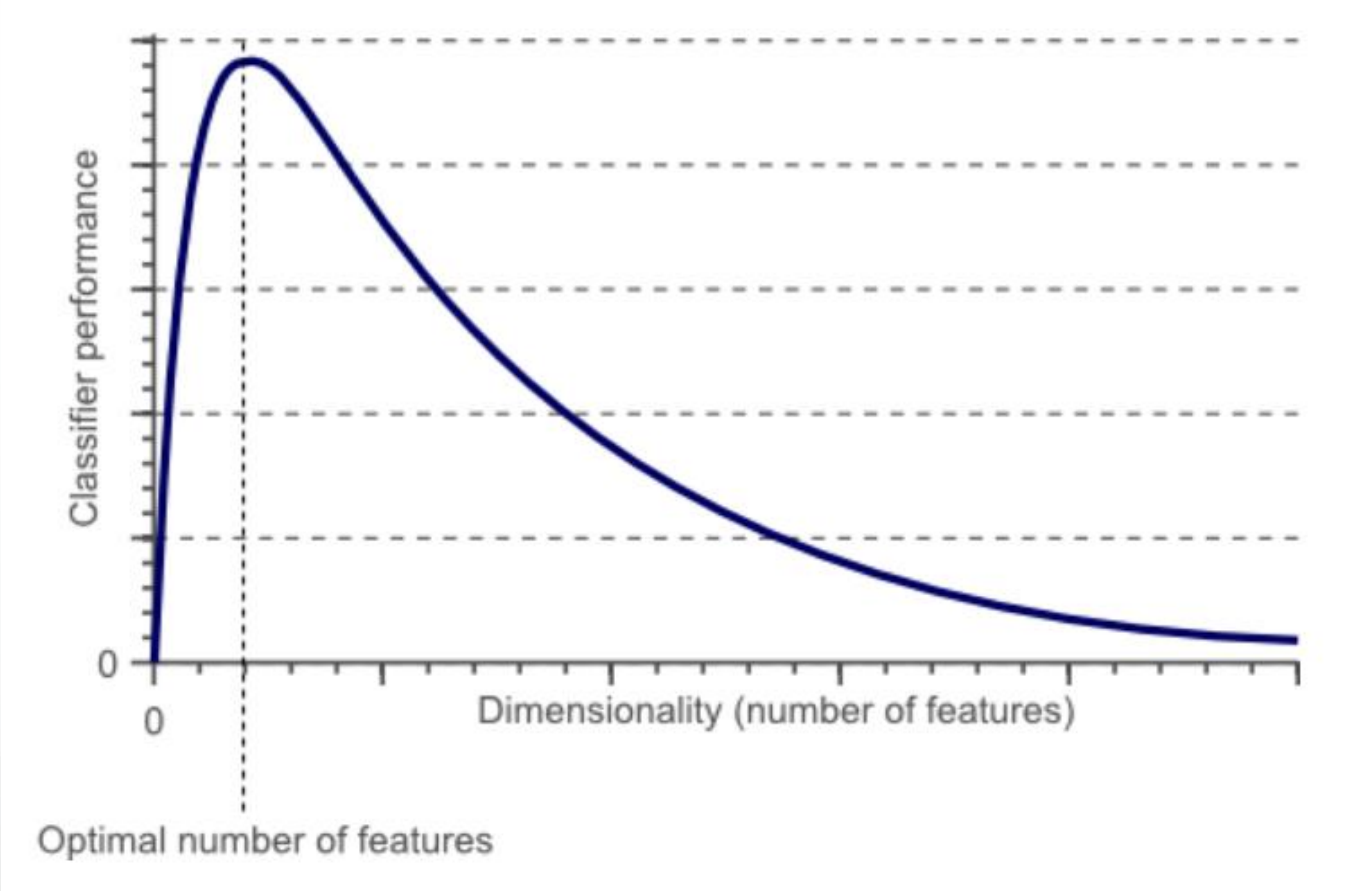
Purpose

To be a data scientist or machine learning engineer, data analysis is important for businesses. Since most of engineers do NOT need to spend much time to build a model, there has been lots of complete modules and algorithms in python. People are much more concerned about how to adapt their dataset to modules.

Garbage in, Garbage out!
Data's quality will affect your model to make decisions.
In practice, data is always extremely complicated, at least dozens of variables (dimensionality) or more.
How can we decide which features are necessary?
That is a trade-off between efficient training and the risk of losing relevant information.
Lets' firstly take a look at the following problem what we may encounter in high dimensions.

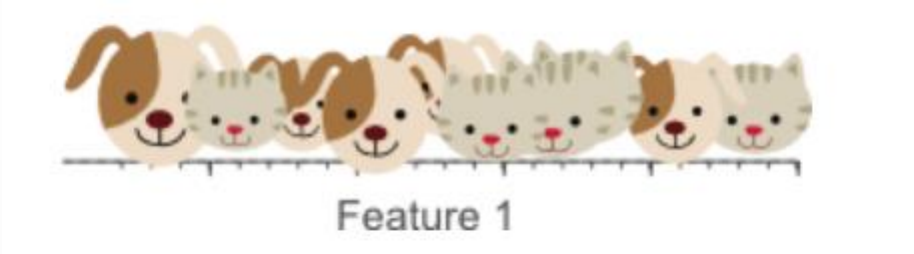
CONTENTS

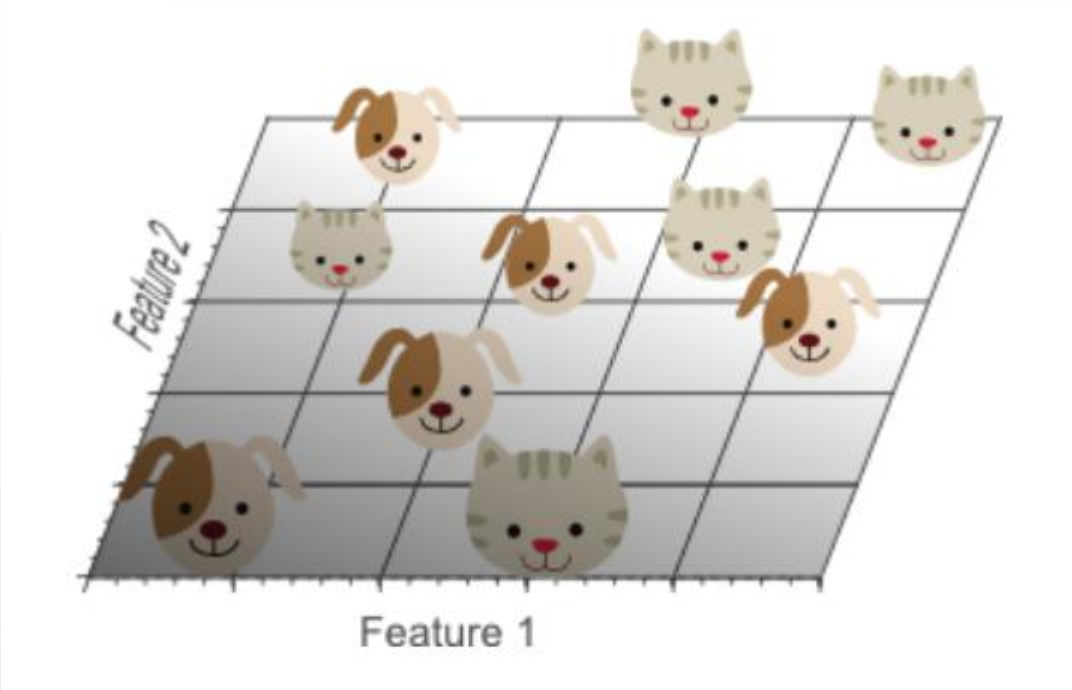
The Curse of Dimensionality (維度災難、維度詛咒)

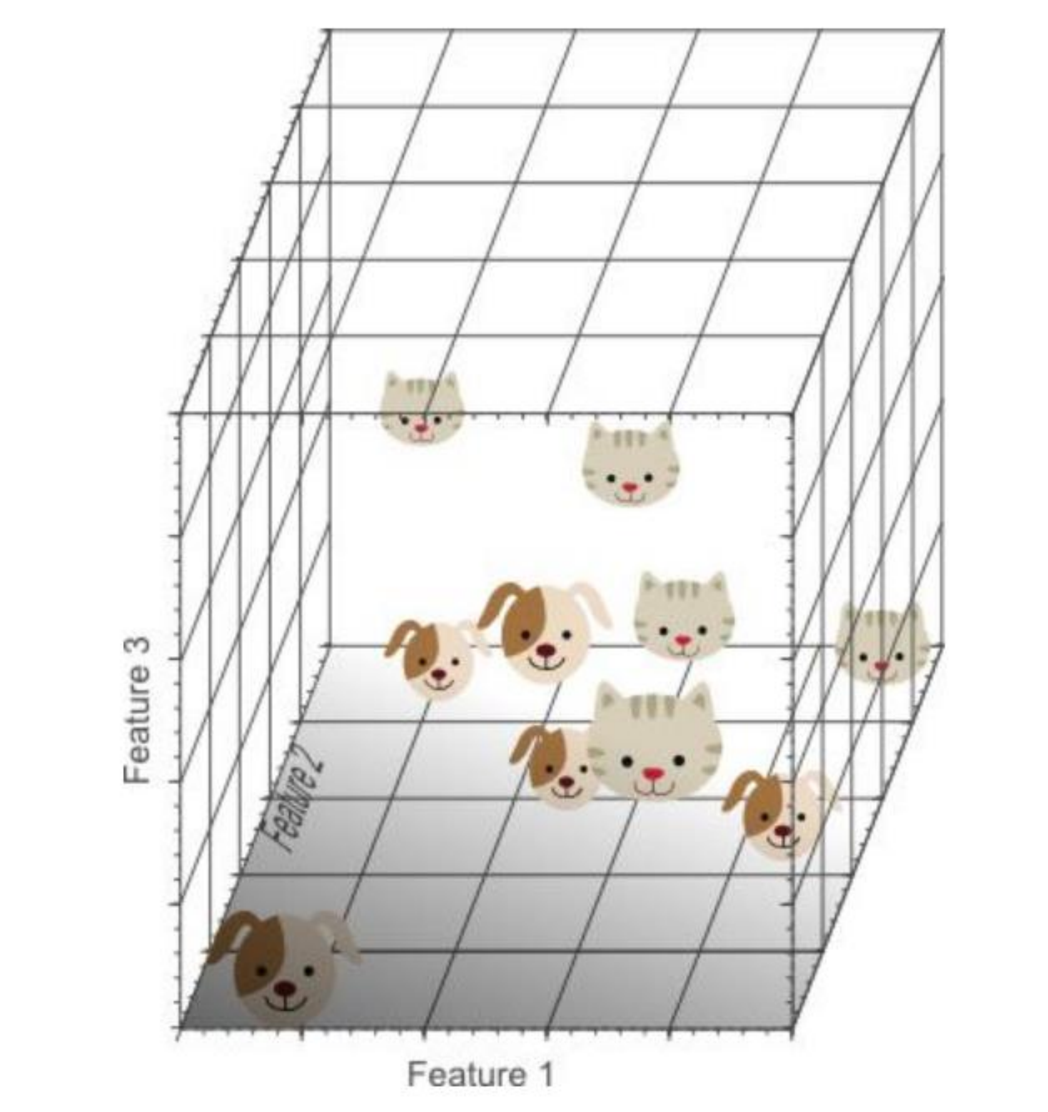


Features -> Dimensions -> location
(red, green, blue, weight ….)
Classifications
(cat, dog …..)

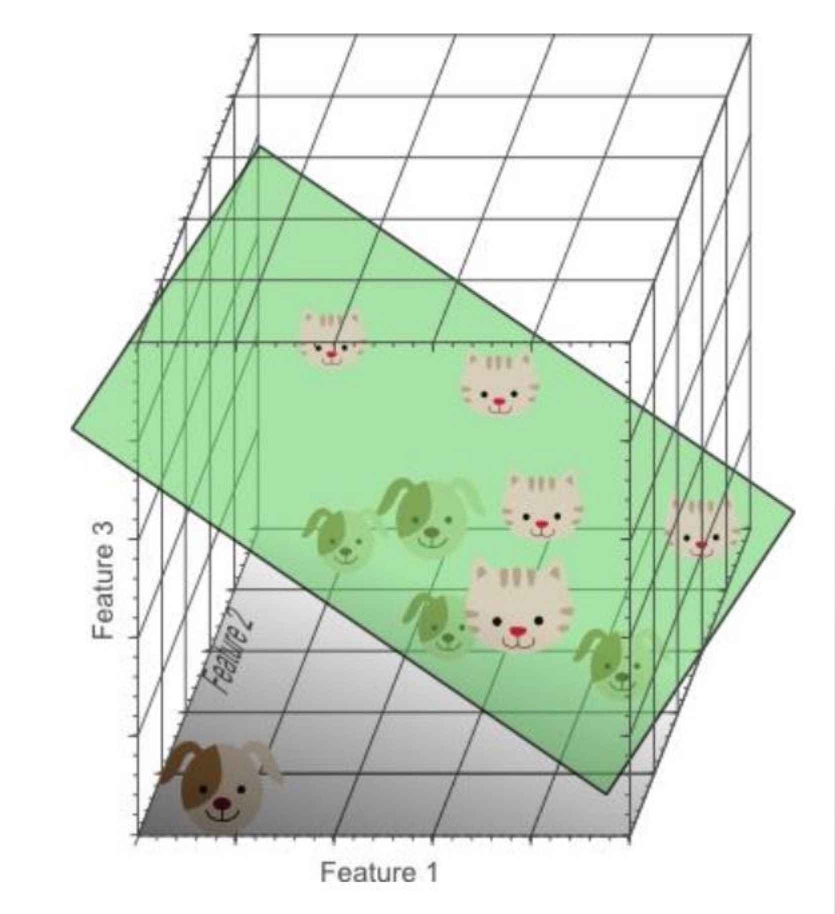
The Curse of Dimensionality

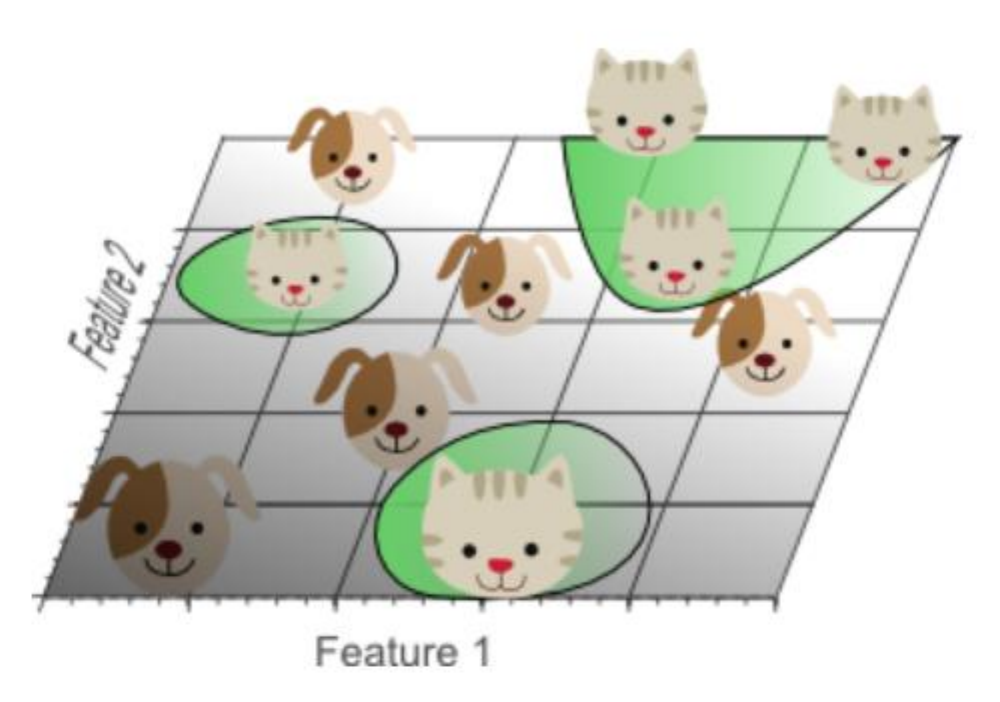


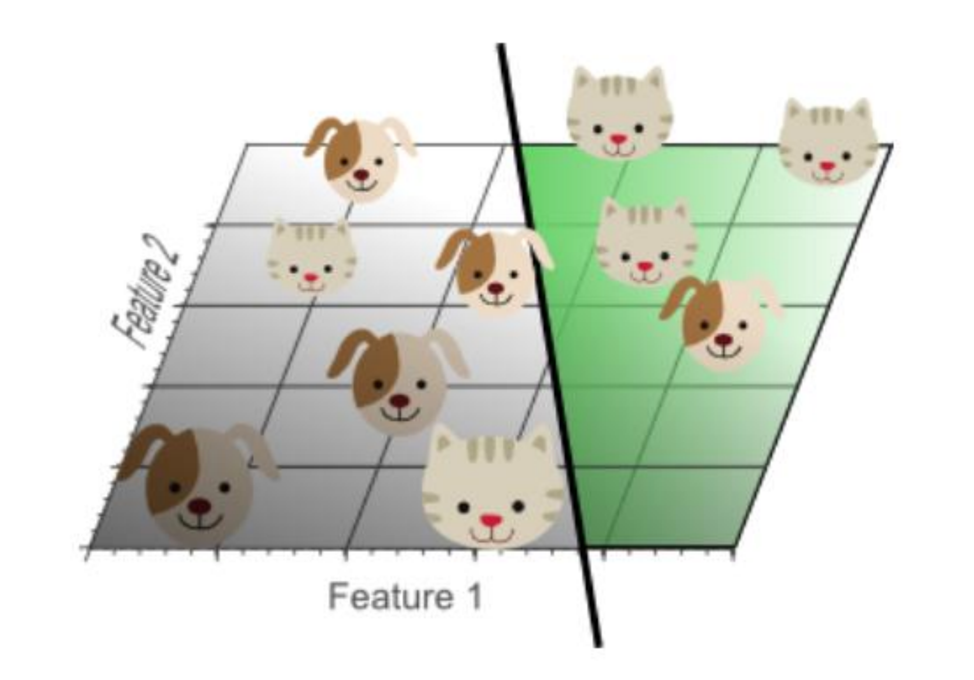




Overfitting





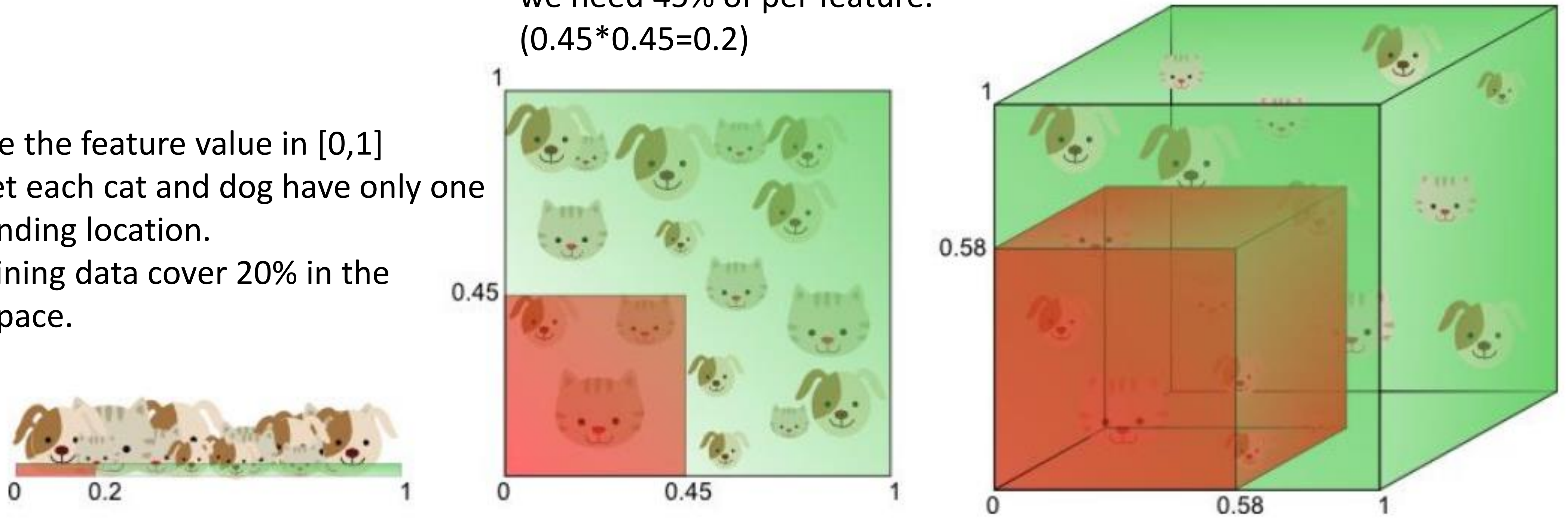


Combinational explosion

Then, we add the second feature.
In order to maintain the proportion as per conversation, we need 45% of per feature.
(0.45*0.45=0.2)

For 3D, we need 58% of per feature.(0.58*0.58*0.58=2)

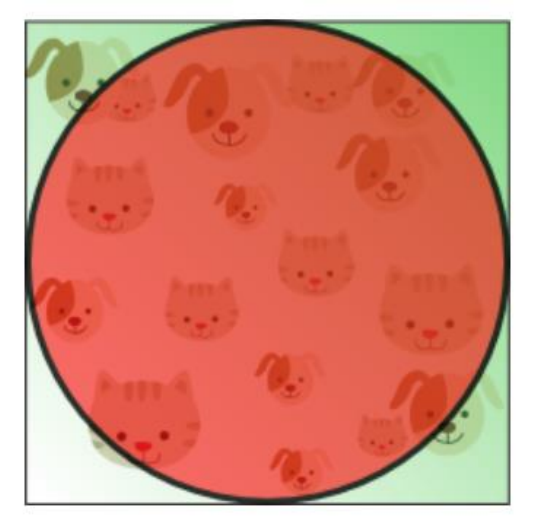
Normalize the feature value in [0,1]
For 1D, let each cat and dog have only one corresponding location.
If our training data cover 20% in the feature space.

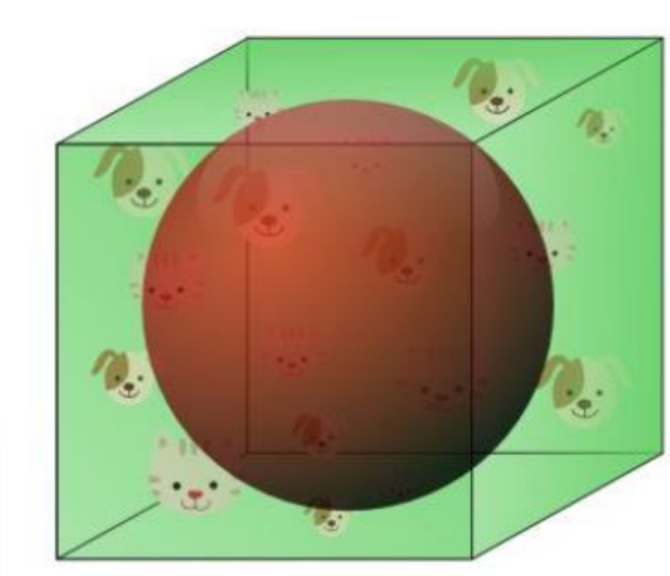


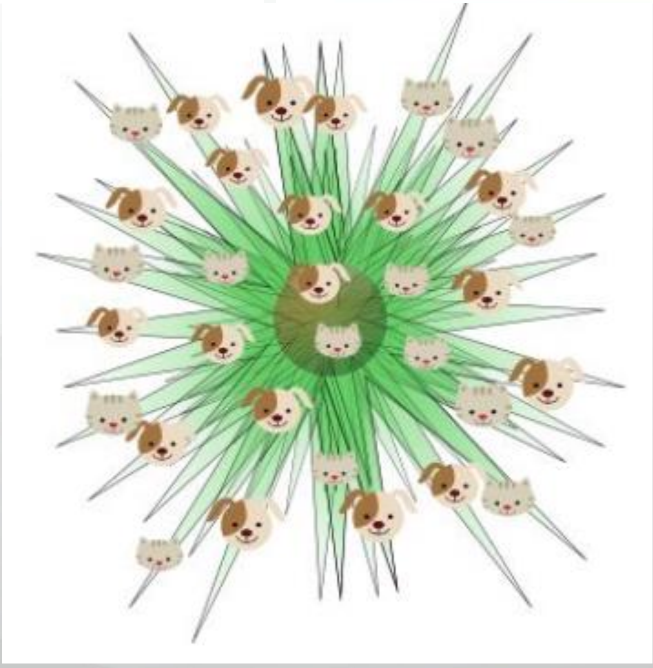
The requirement of # training data is exponential growth

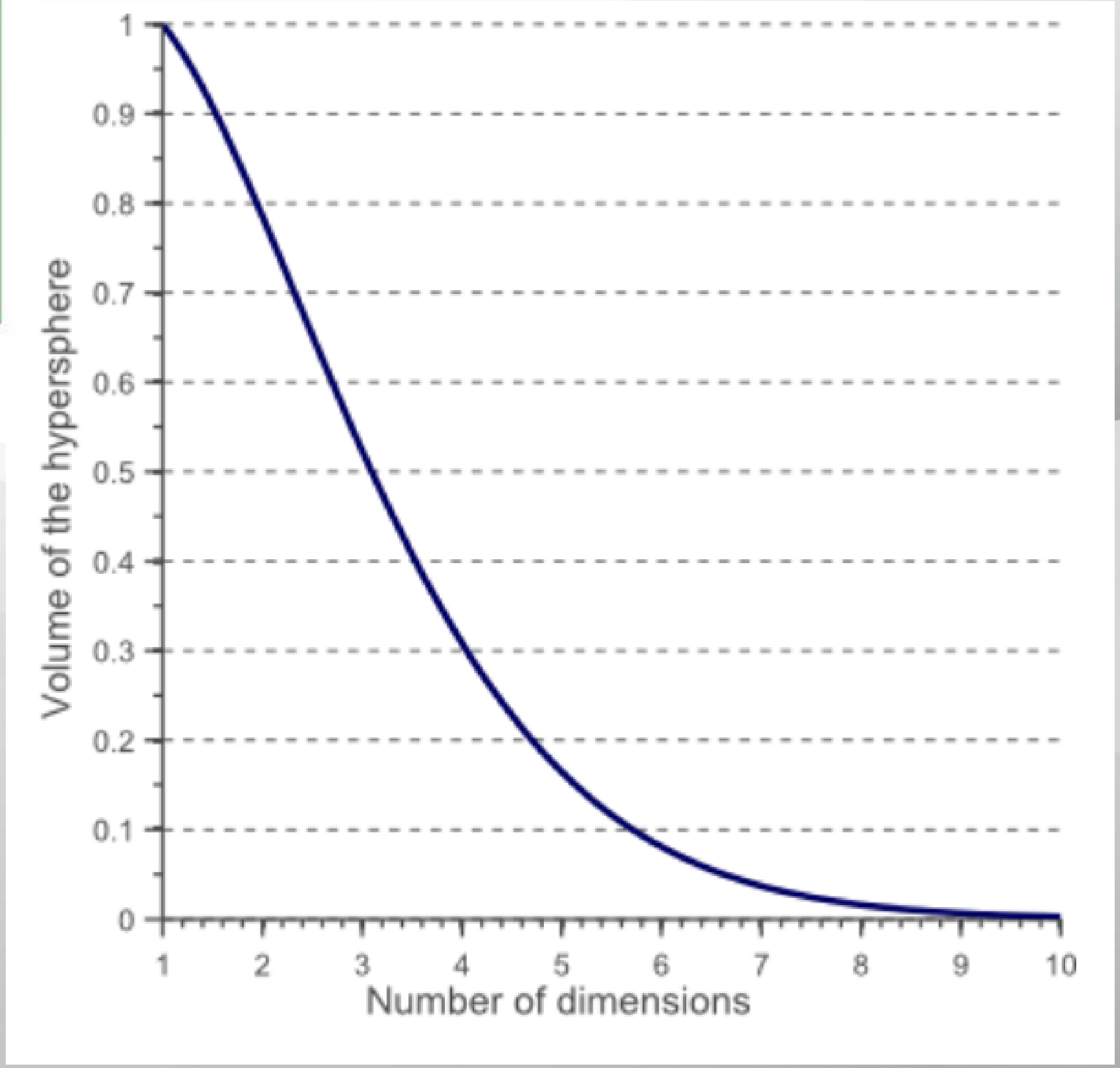
Invalid measurements

$$V(d) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} 0.5^d$$





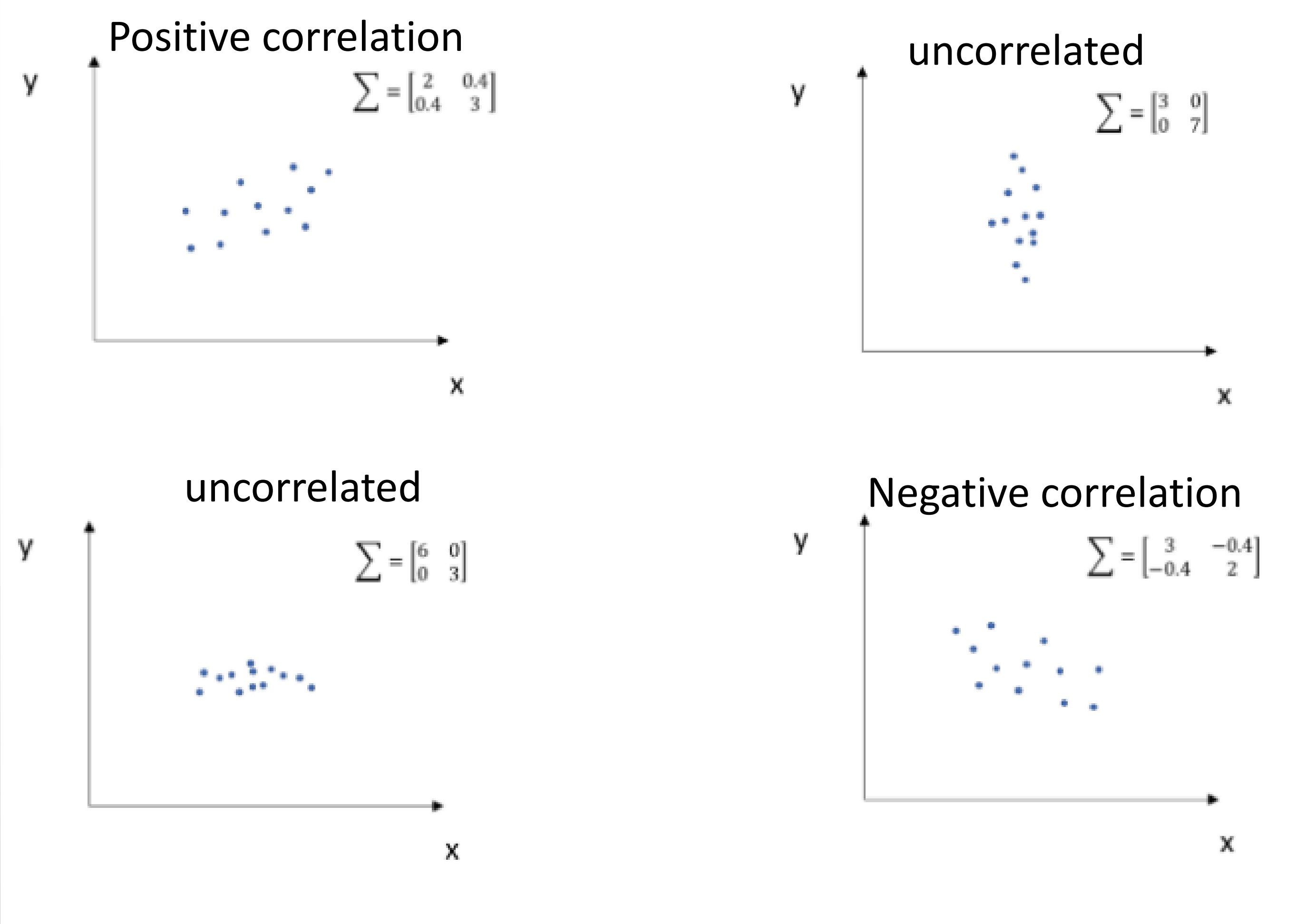




CONTENTS

Linear algebra and probability review

$$\text{Covariance matrix } \Sigma = \begin{bmatrix} V_x & Cov(x,y) \\ Cov(y,x) & V_y \end{bmatrix}$$



Linear algebra and probability review

Singular value decomposition

$$A \epsilon R^{m\times n}, then\ A = USV^T$$

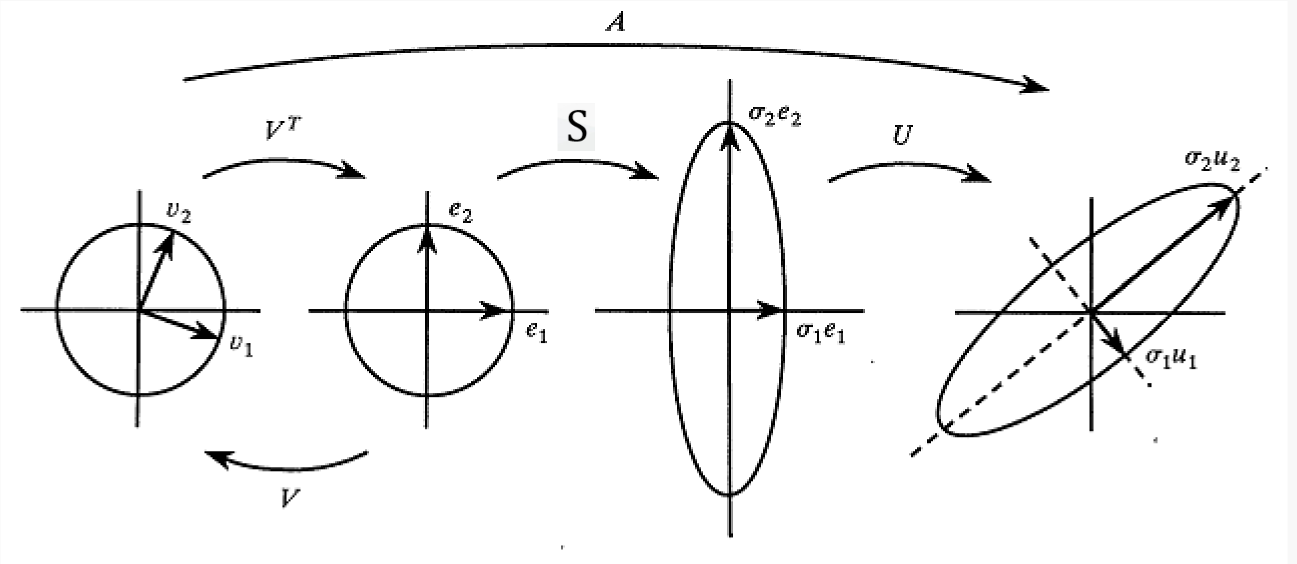
$$U\epsilon R^{m\times m}\ \ is\ an\ orthogonal\ matrix\ (AA^T = UDU^T)$$

$$S \epsilon R^{m\times n}, S = \begin{pmatrix} \sigma_1 & & & & & 0 \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ 0 & & & & & 0 \end{pmatrix}$$

$$V\epsilon R^{n\times n}\quad is\ an\ orthogonal\ matrix\ \ (A^T A = VDV^T)$$

spectral decomposition

$$A \epsilon R^{m\times m}, then\ A = Q\Lambda Q^{-1}$$



A
V^T
S
$\sigma_2 e_2$
U
$\sigma_2 u_2$
v_2
e_2
v_1
e_1
$\sigma_1 e_1$
$\sigma_1 u_1$
V

CONTENTS

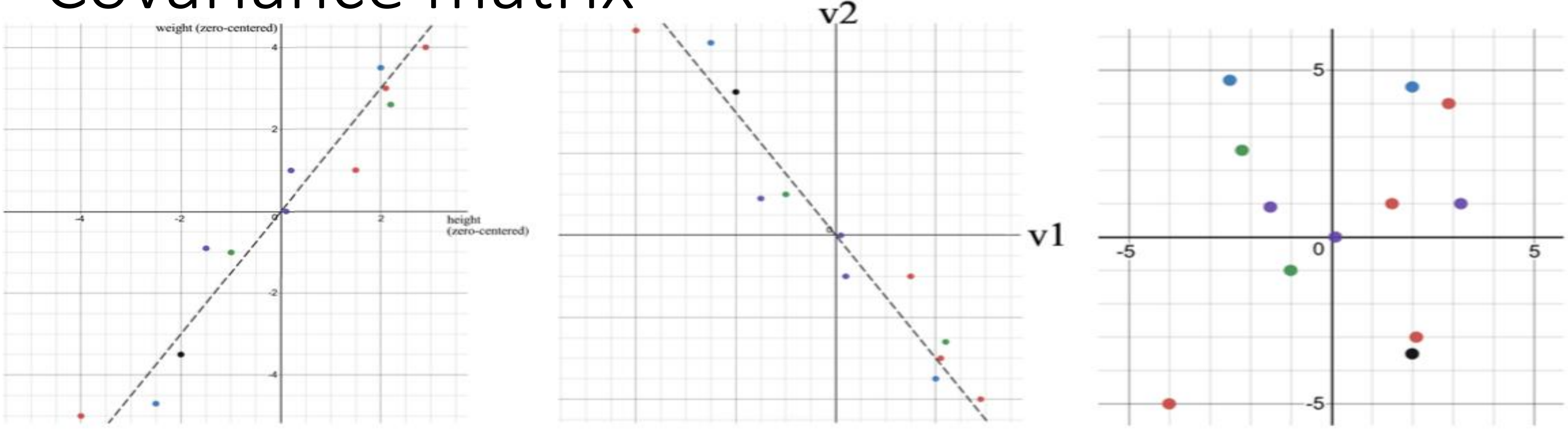
Principal Component Analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components.

1. Maximum variance: PCA aims to make the new data transformed have the larger variance. The reason is that entangled data usually makes model training harder and difficult to be classified. Sparseness in the lower dimensionality is worthwhile to find a decision boundary.
2. Minimum error: Minimize the reconstructed errors caused by using the transformed points to reconstruct original N-dimensional dataset. Furthermore, the error is calculated as the sum of the perpendicular distance from each projected point and its corresponding original point.

$$\Sigma = \begin{bmatrix} Var(x) & Cov(x,y) \\ Cov(y,x) & Var(y) \end{bmatrix}$$

Covariance matrix



$$\frac{1}{11}\begin{bmatrix} 53.46 & 73.42 \\ 73.42 & 107.16 \end{bmatrix} \qquad \frac{1}{11}\begin{bmatrix} 53.46 & -73.42 \\ -73.42 & 107.16 \end{bmatrix} \qquad \frac{1}{11}\begin{bmatrix} 43.5 & 0 \\ 0 & 78.3 \end{bmatrix}$$

weight and height are positively correlated

variable 1 & 2 are negatively correlated

variable 1 & 2 are not correlated

$\sigma_1 = 0.19$

$\sigma_2 = 14.4$

$$V = \begin{bmatrix} -0.8196 & 0.5729 \\ 0.5729 & 0.8196 \end{bmatrix}$$

$\sigma_1 = 0.19$

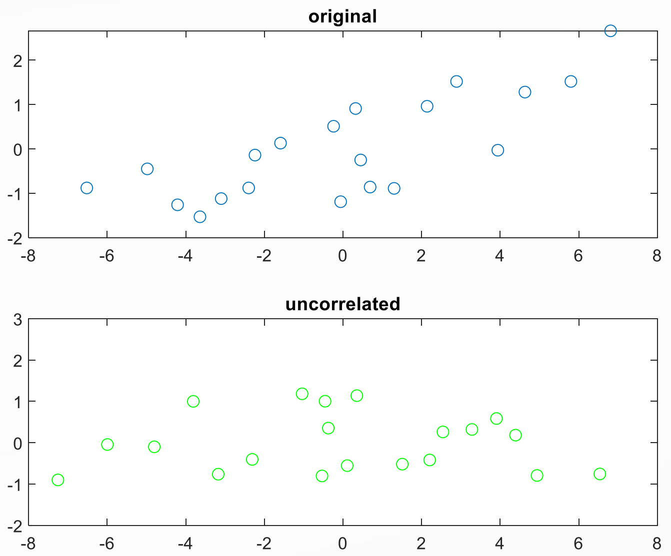
$\sigma_2 = 14.4$

$$V = \begin{bmatrix} -0.8196 & -0.5729 \\ -0.5729 & 0.8196 \end{bmatrix}$$

$\sigma_1 = 3.95$

$\sigma_2 = 7.12$

$$V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\Sigma = Q\Lambda Q^{-1}$$

Go one step further, we can reformulate ΣA as $Q\Lambda Q^T A$:

1. Q^T is a rotation operation such that A is transformed to a new space spanned by eigenspace V.

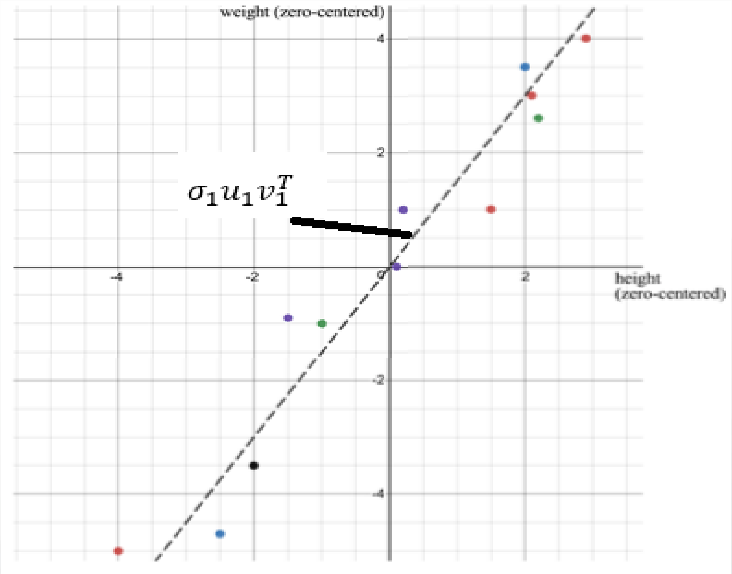
2. Λ scales the new space along the principal axes.

3. finally, our dataset comes full circle back to the Cartesian coordinates.

CONTENTS

SVD

$A \in R^{m\times n}$, $then\ A = USV^T\ = \sigma_1 u_1 v_1^T\ + \cdots + \sigma_r u_r v_r^T\ \ (\ let\ \sigma_1 > \cdots > \sigma_r)$.
Since u_i and v_i are unit vectors, we can even ignore these small terms ($\sigma_i u_i v_i^T$ if σ_i is small).

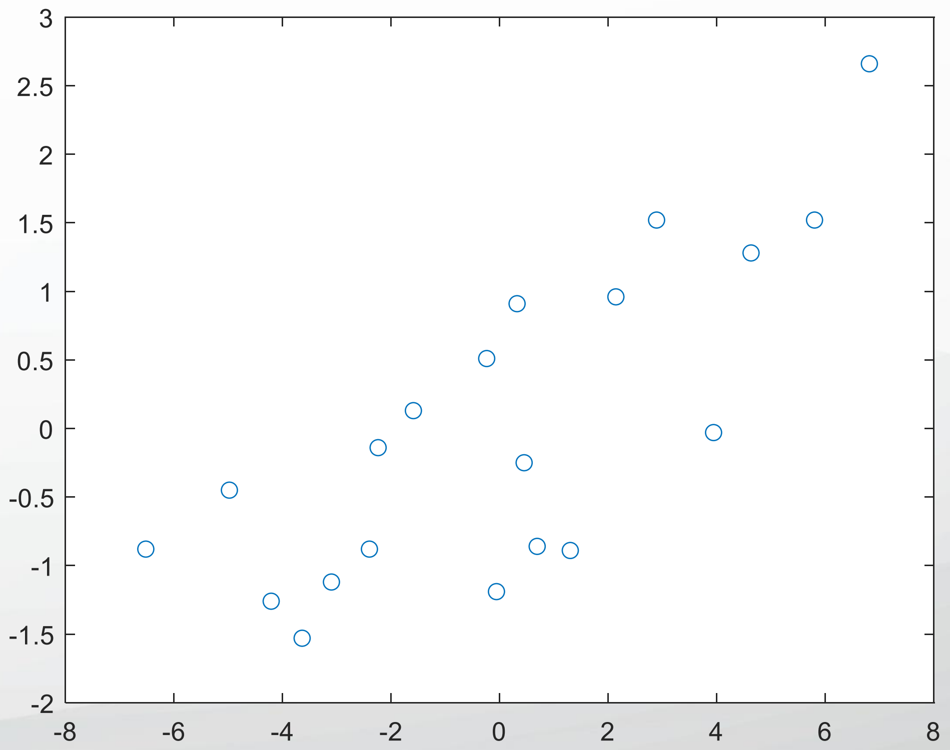


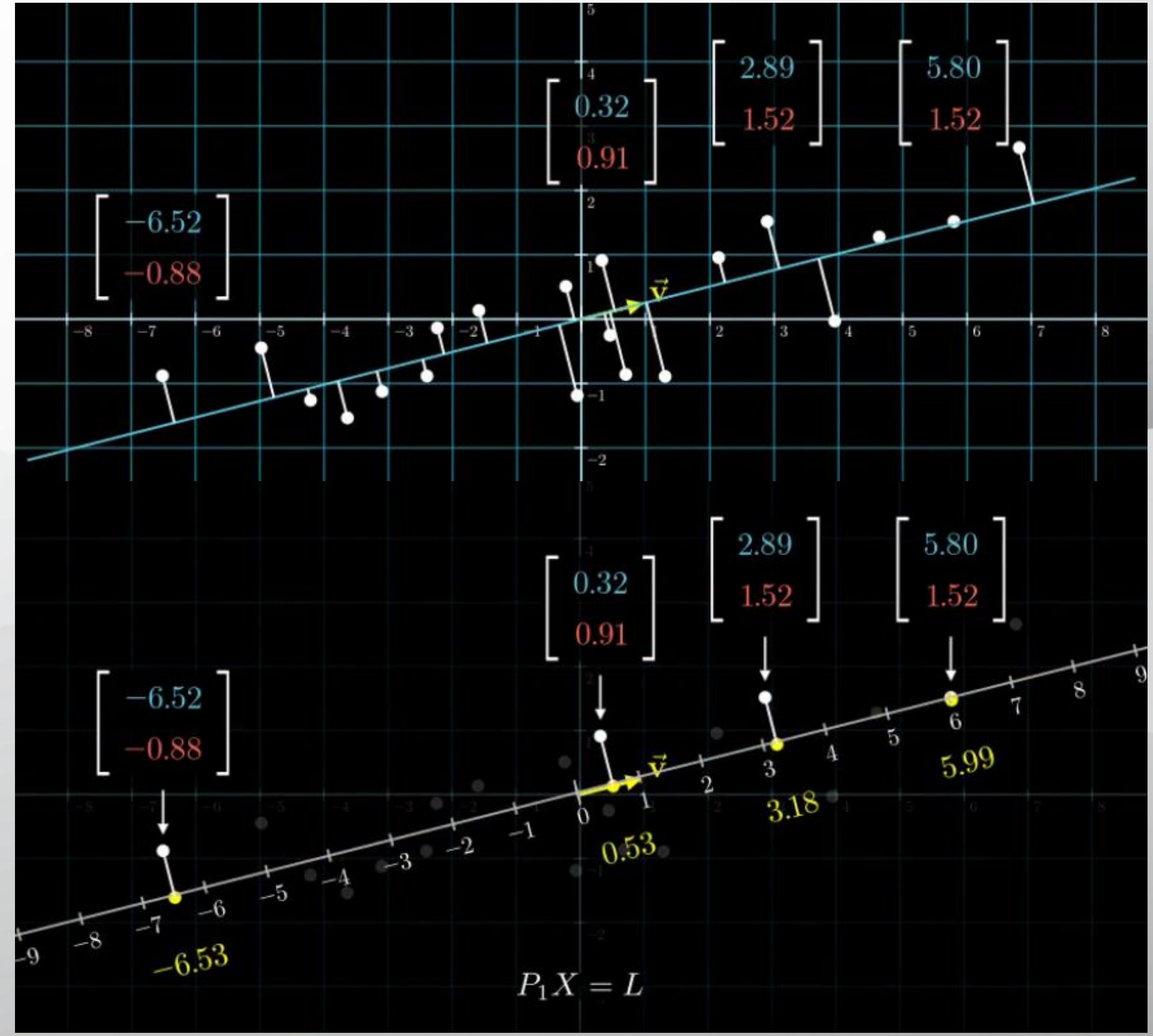
$$A = \sigma_1 u_1 v_1^T + \cancel{\sigma_2 u_2 v_2^T}$$

CONTENTS

Dimensionality reduction

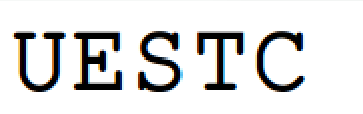
$$\Sigma = \frac{AA^T}{n-1}$$

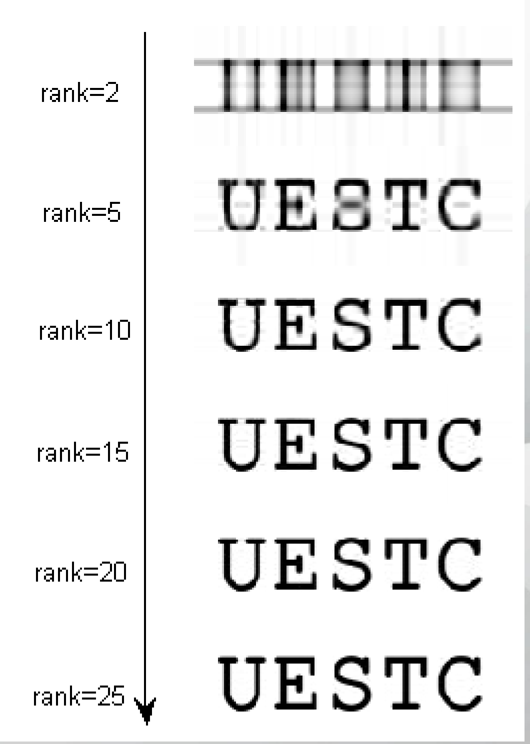




CONTENTS

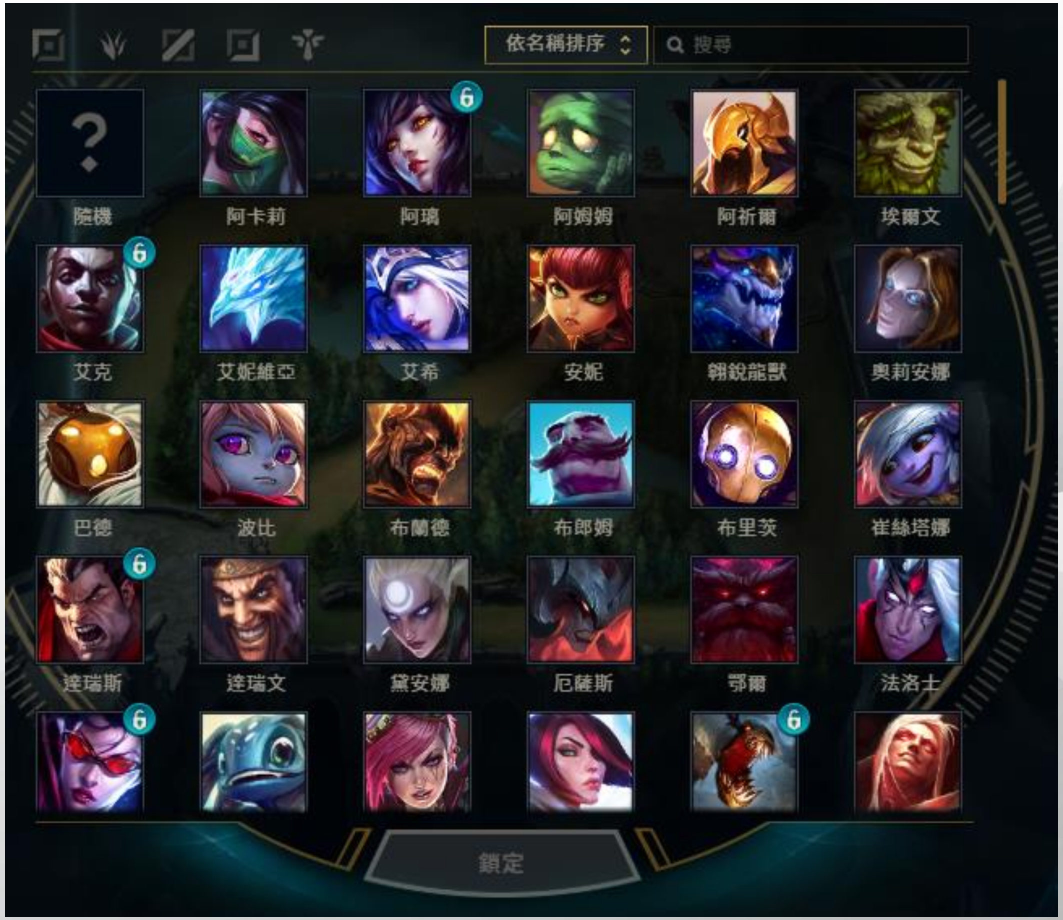
Applications





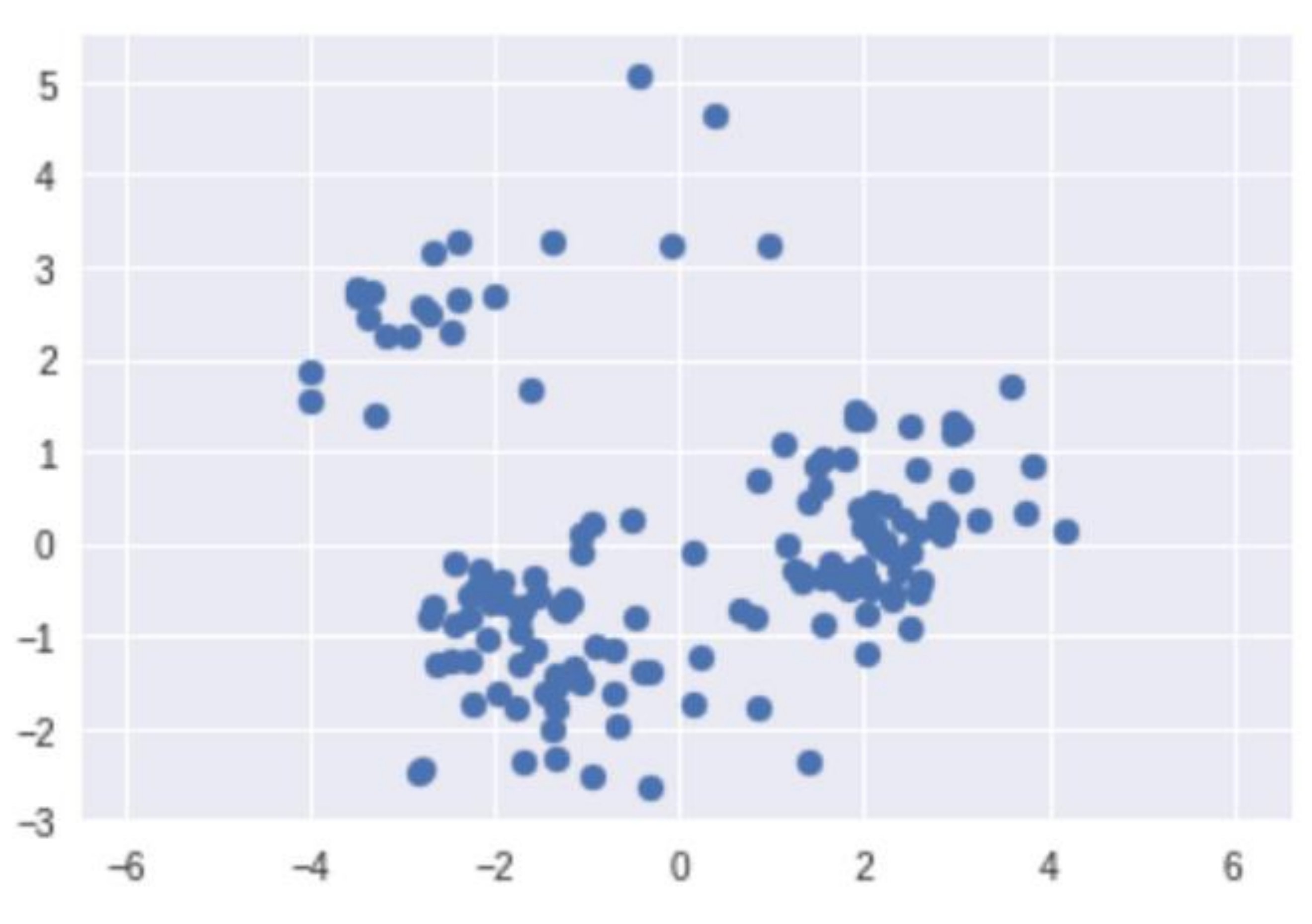
372*675*3

Applications

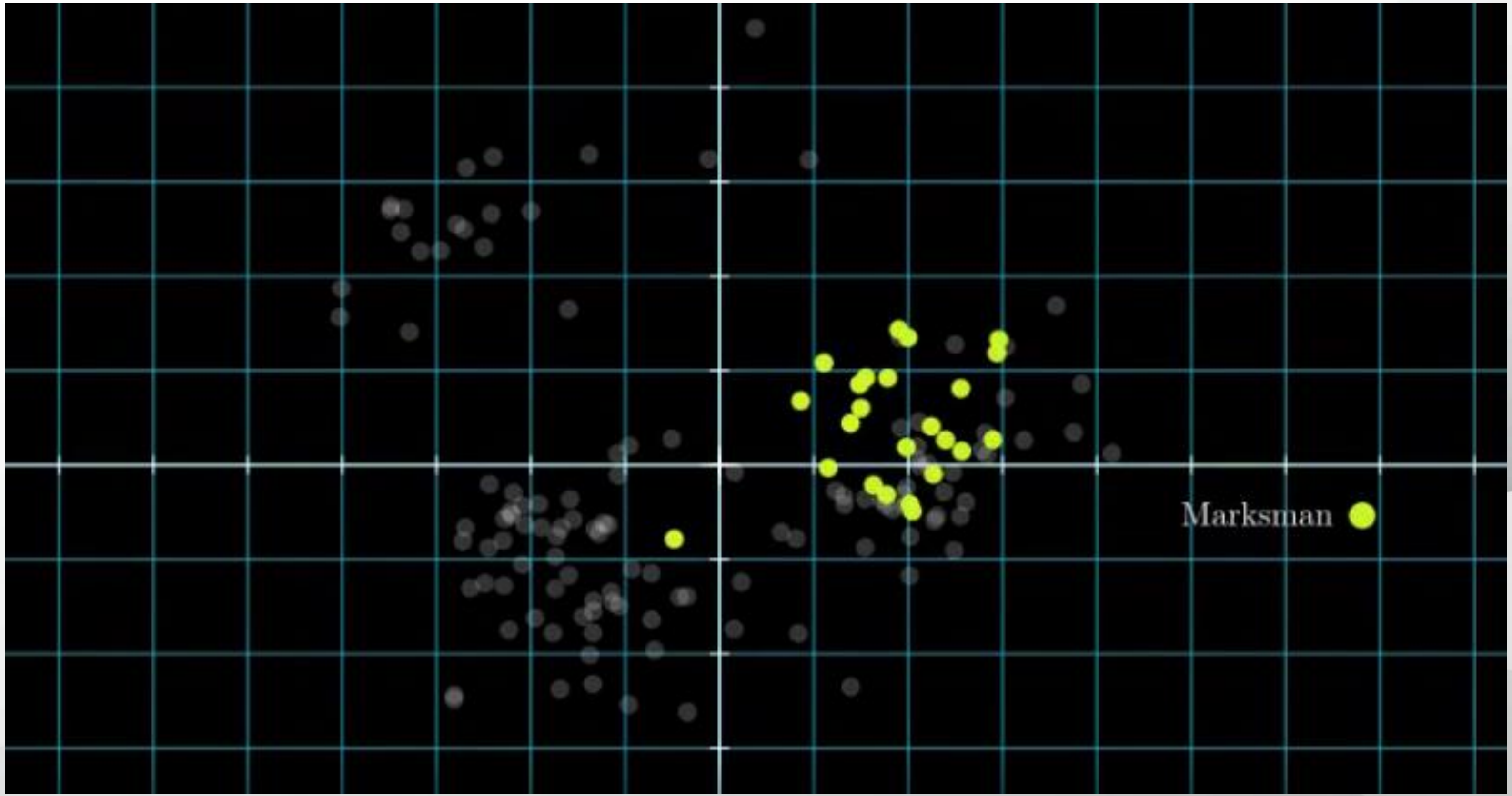


	類型	攻擊距離	魔力	魔力回復	魔力提升	生命提升	生命	生命回復	移動速度	物理攻擊	物理防禦	魔法防禦
名稱												
厄薩斯	鬥士	175	0	0.0	0	90	580.0	3.0	345	60.0	33.0	32.1
阿璃	法師	550	418	0.8	25	92	526.0	6.5	330	53.0	20.9	30.0
阿卡莉	刺客	125	200	0.0	0	95	575.0	8.0	345	62.4	23.0	37.0
亞歷斯塔	坦克	125	350	0.8	40	106	573.4	8.5	330	61.1	44.0	32.1
阿姆姆	坦克	125	287	0.5	40	84	613.1	9.0	335	53.4	33.0	32.1

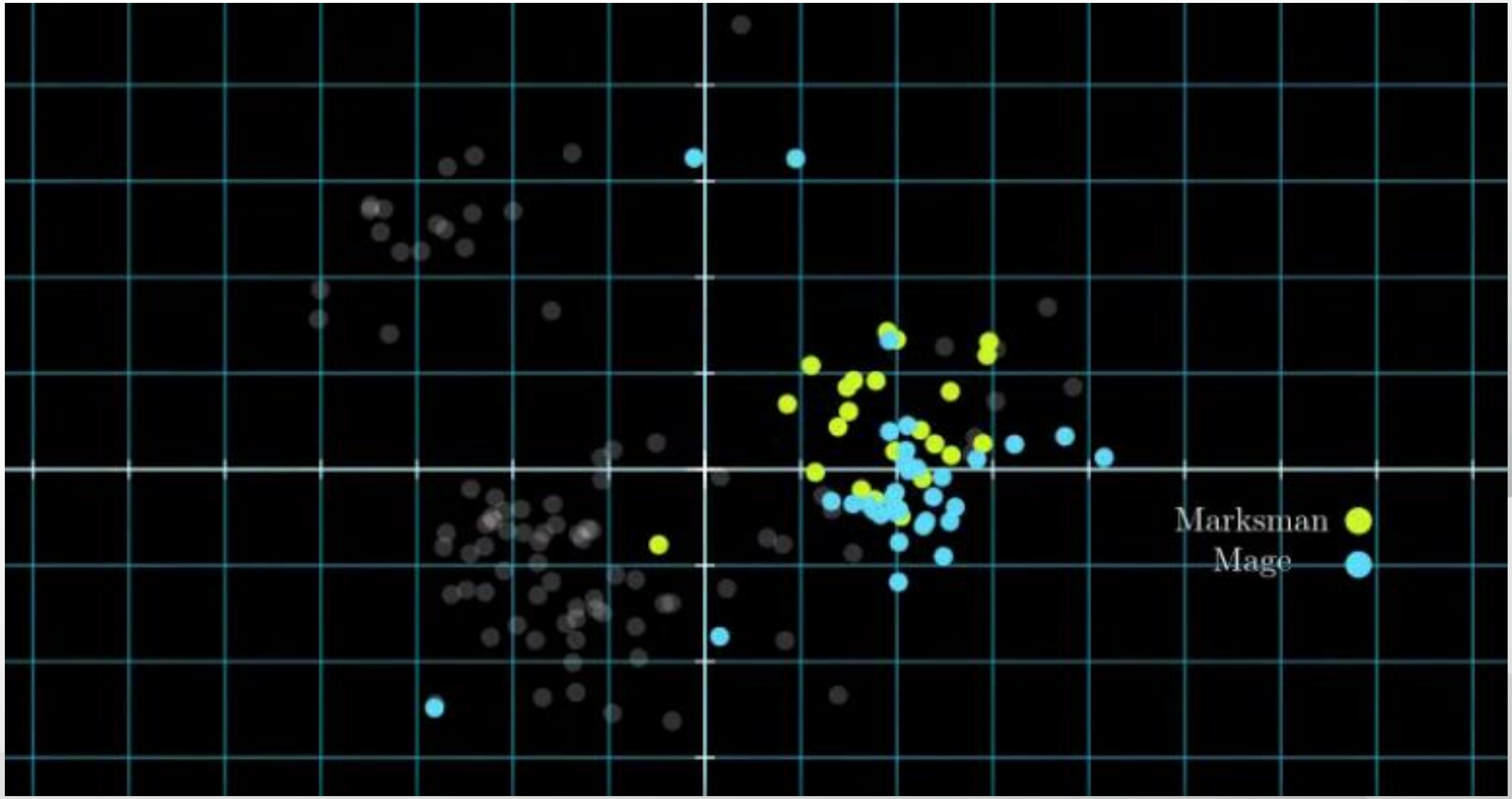
	類型	攻擊距離	魔力	魔力回復	魔力提升	生命提升	生命	生命回復	移動速度	物理攻擊	物理防禦	魔法防禦
名稱												
厄薩斯	鬥士	-0.77	-2.69	-2.02	-1.99	0.22	0.65	-2.03	1.15	0.10	1.19	0.57
阿璃	法師	1.14	0.94	0.84	-0.48	0.52	-0.30	-0.06	-0.84	-1.04	-1.37	-0.66
阿卡莉	刺客	-1.03	-0.95	-2.02	-1.99	0.98	0.52	0.79	1.15	0.49	-1.05	3.45
亞歷斯塔	坦克	-1.03	0.35	0.84	0.42	2.64	0.47	1.07	-0.84	0.28	2.08	0.57
阿姆姆	坦克	-1.03	-0.20	-0.23	0.42	-0.68	1.54	1.35	-0.17	-0.97	0.44	0.57



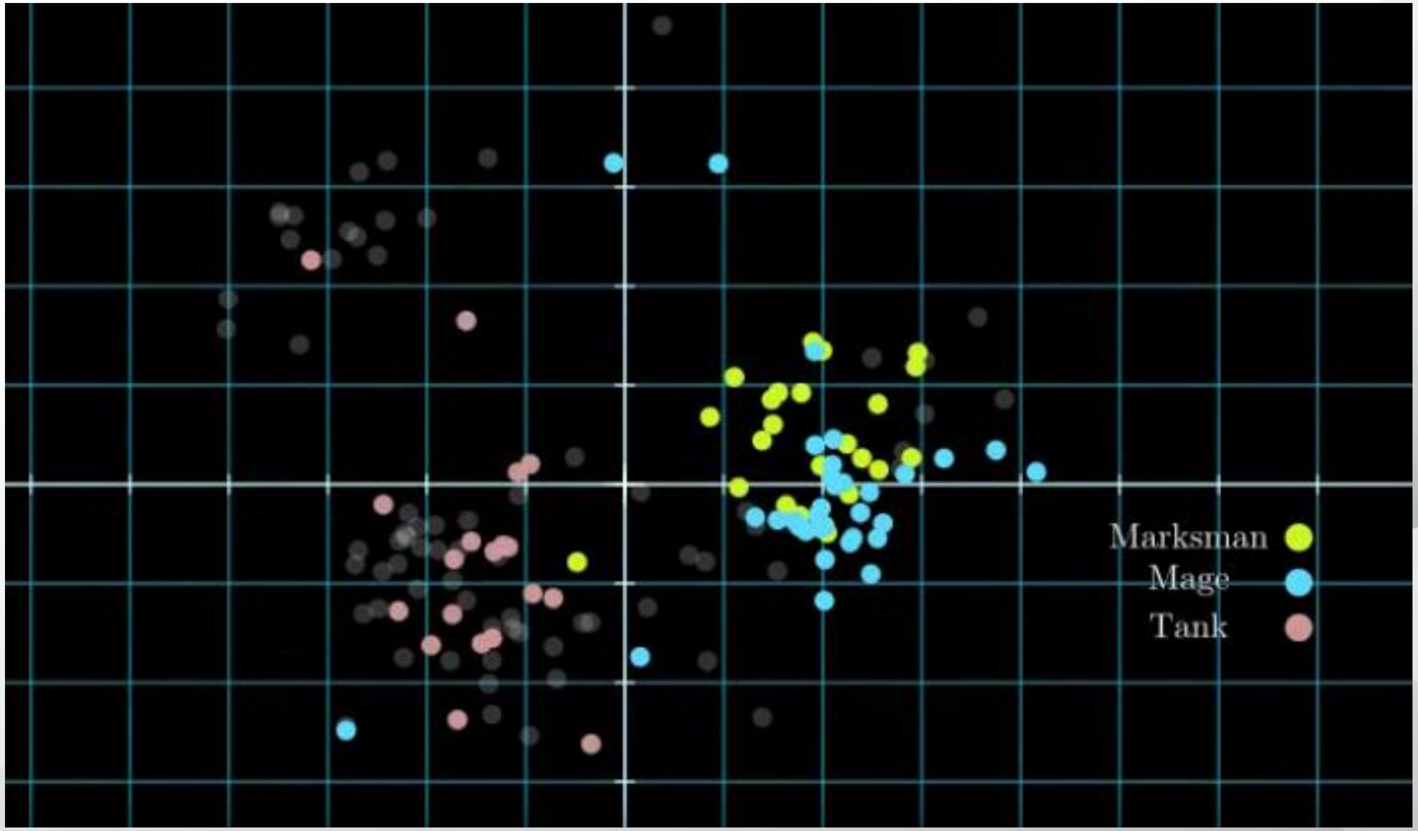
	攻擊距離	魔力	魔力回復	魔力提升	生命提升	生命	生命回復	移動速度	物理攻擊	物理防禦	魔法防禦
第一主成分	0.43	0.28	0.19	0.14	-0.14	-0.32	-0.31	-0.33	-0.35	-0.34	-0.34
第二主成分	0.072	-0.43	-0.56	-0.5	-0.32	-0.24	-0.21	0.014	-0.021	-0.08	-0.16

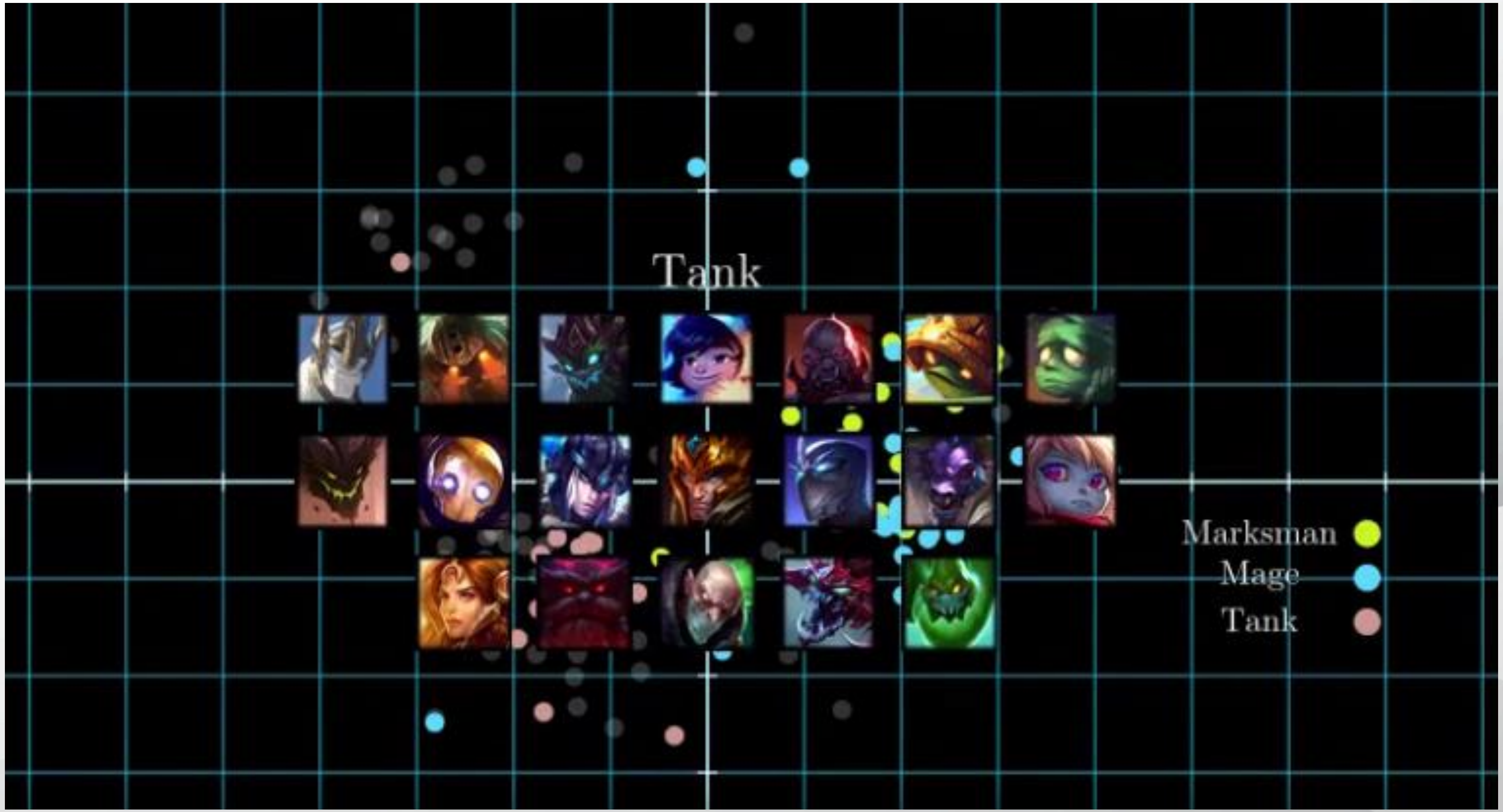

Marksman

$l_1 = x$ decrease
Marksman
Marksman


Marksman
Mage


Mage
Marksman
Mage


Marksman
Mage
Tank


Tank
Marksman
Mage
Tank

0. Original author: https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/
Chinese version: https://zhuanlan.zhihu.com/p/26945814
1. A survey on unsupervised outlier detection in high-dimensional numerical data, Arthur Zimek , Erich Schubert and Hans-Peter Kriegel, 2012;
2. Spatial ordering and encoding for geographic data mining and visualization, Diansheng Guo, Mark Gahegan, 2006, page10;
3. 世界上最生動的PCA:直觀理解並應用主成分分析,LeeMeng,2020;
https://leemeng.tw/essence-of-principal-component-analysis.html
4. Machine Learning — Singular Value Decomposition (SVD) & Principal Component Analysis (PCA), Jonathan Hui, 2019;
https://jonathan-hui.medium.com/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45e885e491
5. PCA: Eigenvectors and Eigenvalues, Valentina Alto, 2019;
https://towardsdatascience.com/pca-eigenvectors-and-eigenvalues-1f968bc6777a
6. You Don’t Know SVD (Singular Value Decomposition), Hussein Abdullatif,2019;
https://towardsdatascience.com/svd-8c2f72e264f
7. Relationship between SVD and PCA. How to use SVD to perform PCA?
https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca
8. how to do SVD and PCA with big data?
https://datascience.stackexchange.com/questions/1159/how-to-do-svd-and-pca-with-big-data
9.SVD在图像处理中的基本应用
https://heleifz.github.io/1508462629253.html