Fundamentos de Ciencias de Datos

Semana 11 - Reducción de Dimensionalidad

Dimensión de un Dataset

En el contexto de *Machine Learning* hemos hablado de la dimensión de un *dataset*

Por ejemplo, considera la siguiente versión del *dataset Iris*

X =

Largo del pétalo	Ancho del pétalo
1.4	0.2
1.4	0.2
1.3	0.2
...	...

y =

Tipo de flor
Iris Setosa
Iris Setosa
Iris Setosa
...

Dimensión de un Dataset

Aquí nuestros datos (X) incluyen el largo y ancho del pétalo

De hecho al ser 2 dimensiones, incluso lo podemos visualizar

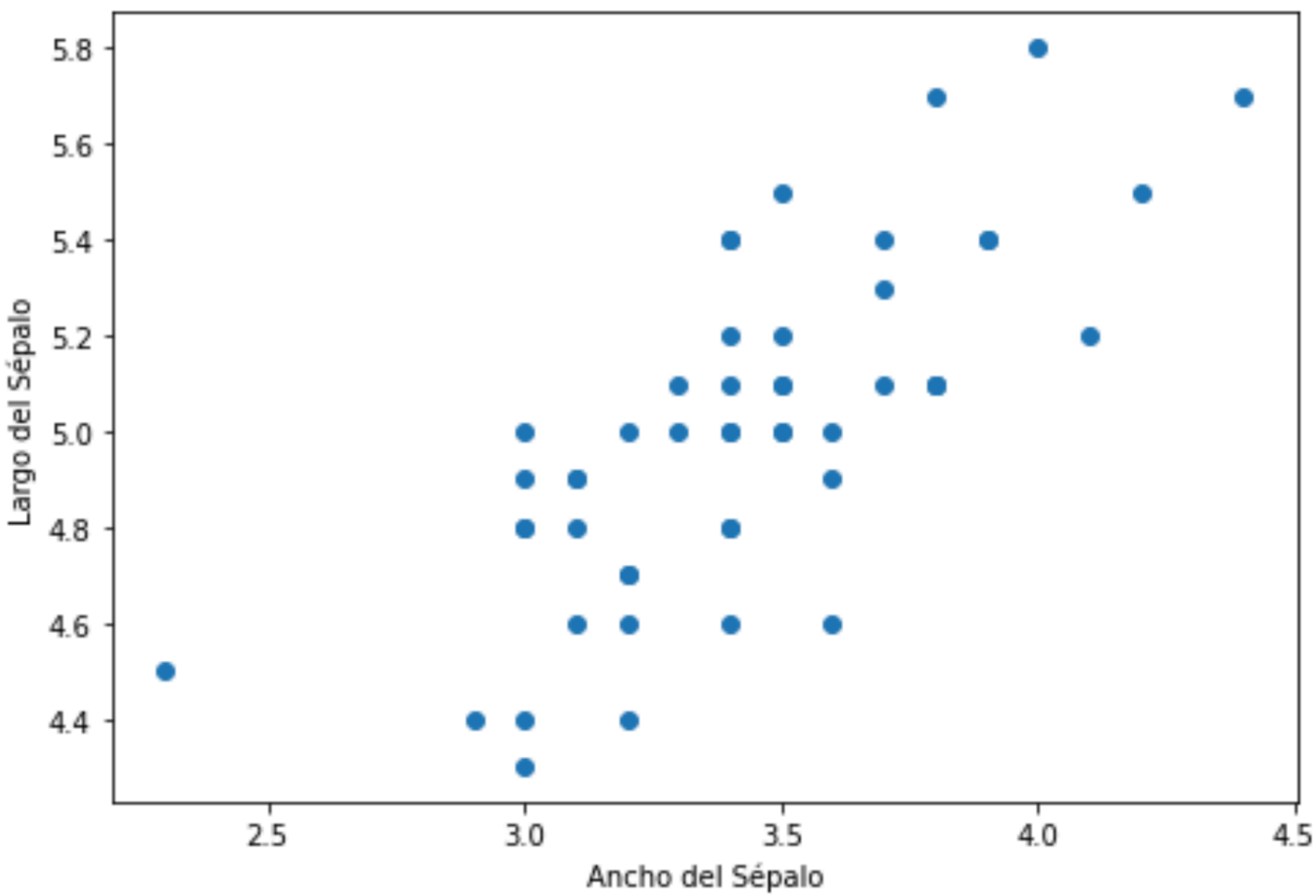
X =

Largo del pétalo	Ancho del pétalo
1.4	0.2
1.4	0.2
1.3	0.2
...	...

y =

Tipo de flor
Iris Setosa
Iris Setosa
Iris Setosa
...

Dimensión de un Dataset



Dimensión de un Dataset

¿Pero qué pasa cuando tenemos 3 dimensiones?

¿Y si tenemos 4 dimensiones?

¿Y si tenemos 784 dimensiones (como el *dataset* MNIST)?

Dimensión de un Dataset

Piensa en una imagen de 1080x720 pixeles

¿Cuál es la dimensión de un *dataset* que tiene imágenes de esa resolución?

¿Qué problemas traen los *dataset* que tienen grandes dimensiones?

Reducción de Dimensionalidad

Muchas veces hemos escuchado el término "compresión"

Reducción de Dimensionalidad

Esta misma idea la podemos usar en el contexto de *Machine Learning*

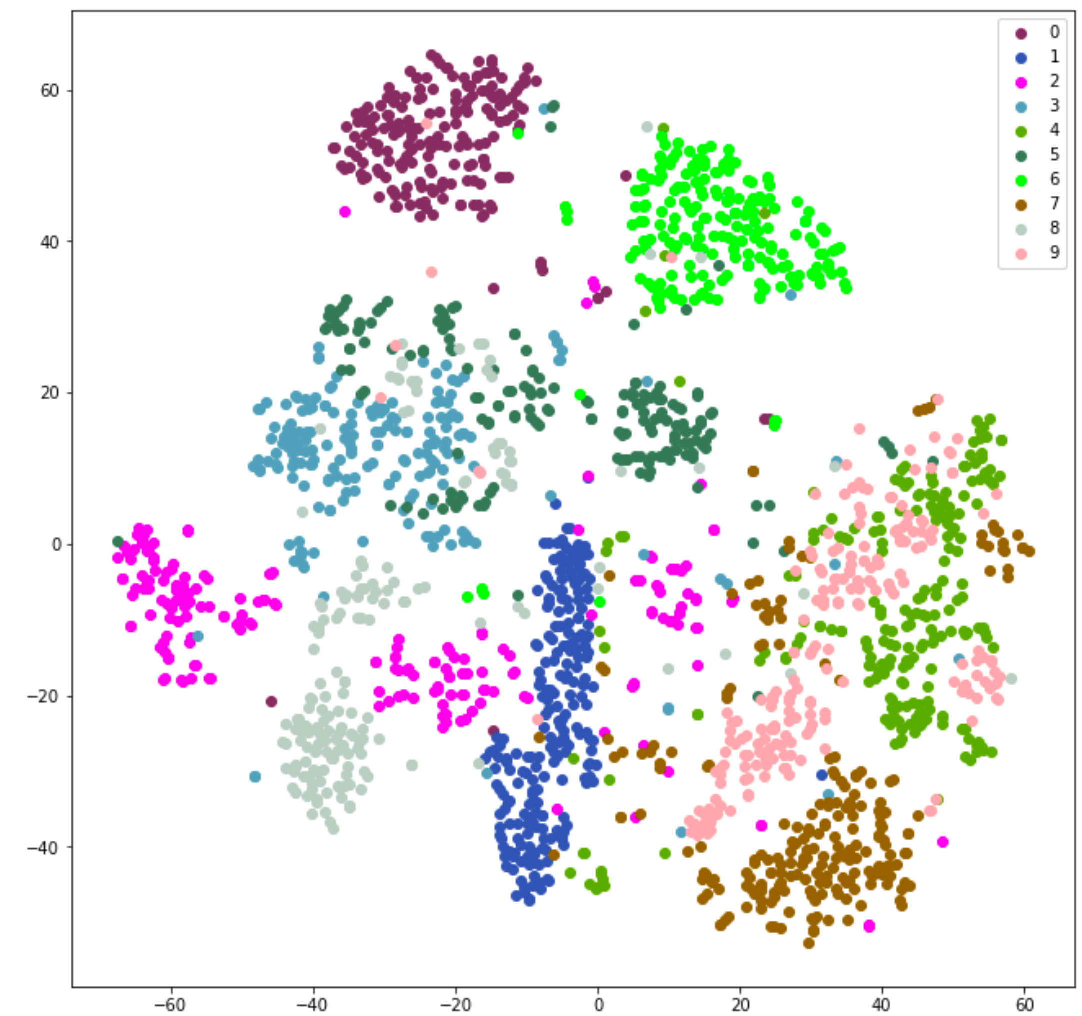
Al **reducir dimensionalidad** de un *dataset*, queremos "comprimir" el *dataset* a uno de menos dimensiones que pierda la menor cantidad de información posible

Reducción de Dimensionalidad

Ejemplo - Reducción de Dimensionalidad con TSNE

Con el algoritmo TSNE podemos transformar cada imagen de 28x28 en vectores de dos dimensiones

Aquí vemos que a pesar de perder muchas dimensiones, varios números se distinguen muy bien de los otros



Por qué reducir la dimensión

Nos gusta reducir la dimensión de los *dataset* porque:

- Podemos hacer visualizaciones
- Podemos entrenar modelos que generalizan mejor
- El tamaño de los *datasets* se vuelven manejables
- Podemos reducir la "maldición de la dimensión"

Curse of Dimensionality

La "maldición de la dimensión" hace referencia a los fenómenos que genera trabajar con datos de varias dimensiones

En general, se debe a que los espacios de más dimensiones tienden a ser **poco densos**

Curse of Dimensionality

Ejemplo

Supongamos una esfera en d dimensiones de radio r dentro de un hipercubo de lado $2r$

La razón entre el "volumen" de la esfera y el hipercubo es:

$$\frac{\pi^{d/2}}{2^{d-1} d \Gamma(d/2)}$$

¡Esta razón tiende a 0 cuando d tiende a infinito! (debido a que hay "más espacio" entre el cubo y la esfera mientras más dimensiones tengamos)

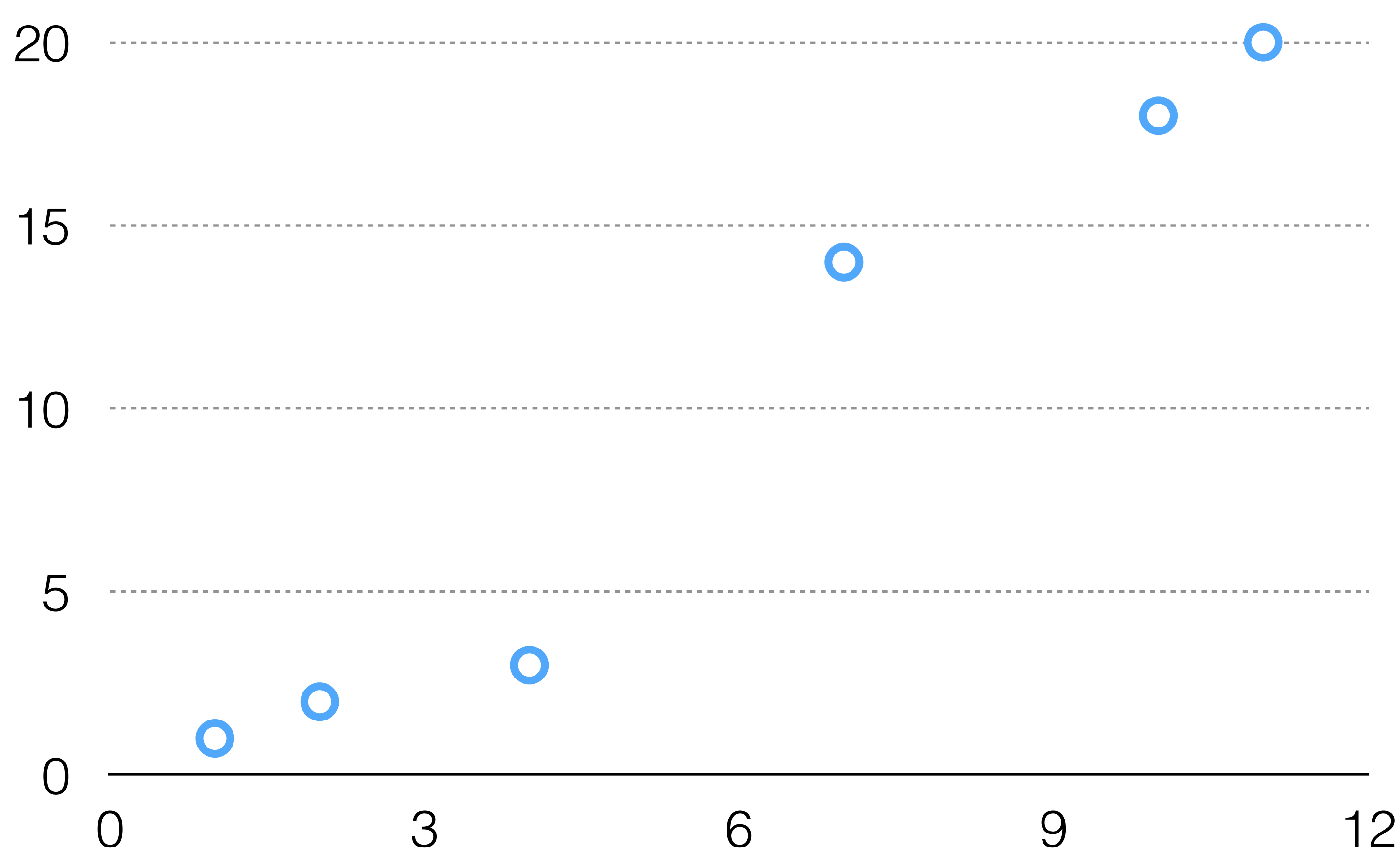
¿Y cómo reducimos la dimensionalidad?

Reducción de Dimensionalidad

Existen varios algoritmos para reducir reducción de dimensionalidad, pero en este curso vamos a ver **PCA**: Principal Component Analysis

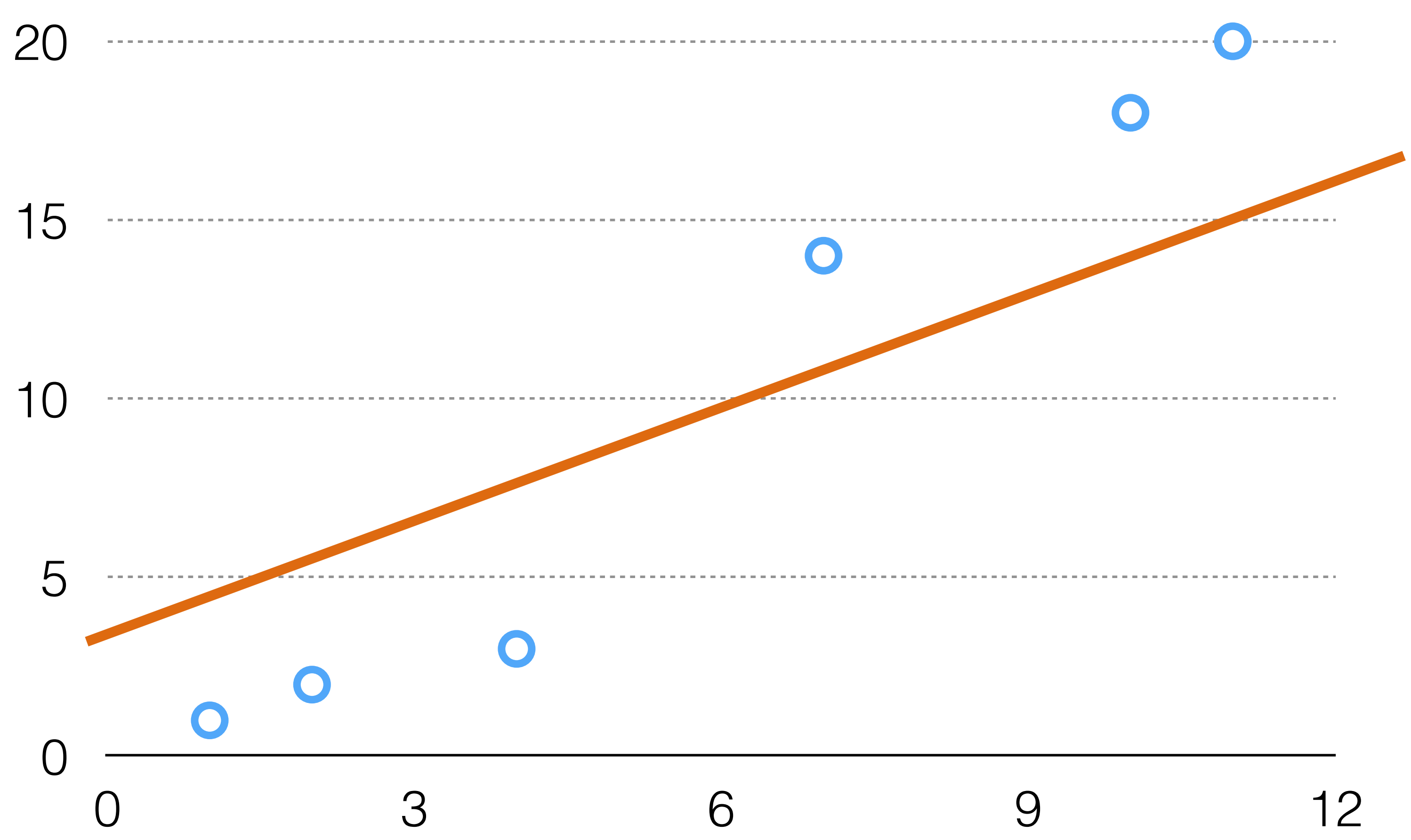
PCA

La idea principal de PCA es proyectar los datos en varias dimensiones a hiperplanos de menor dimensión



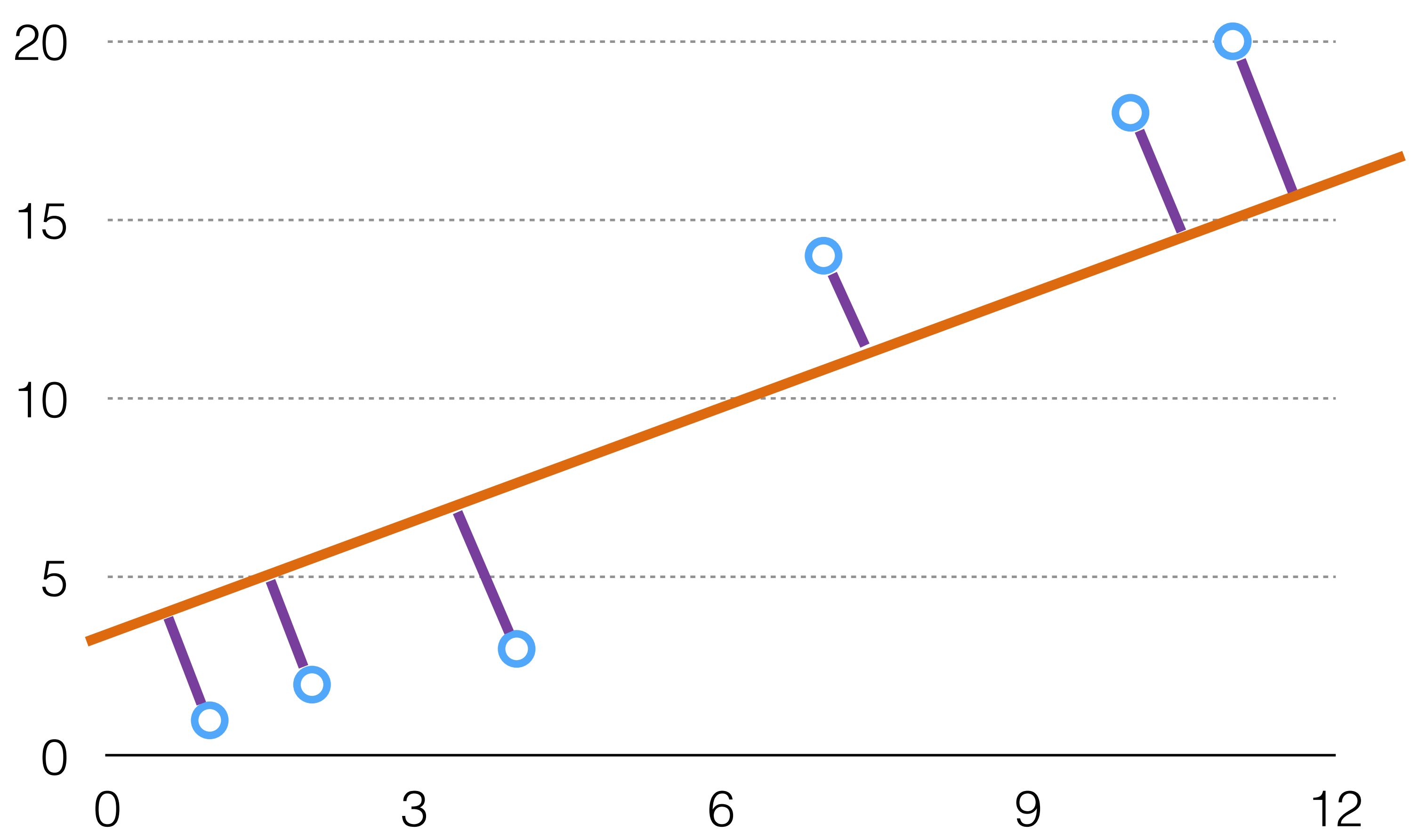
PCA

La idea principal de PCA es proyectar los datos en varias dimensiones a hiperplanos de menor dimensión



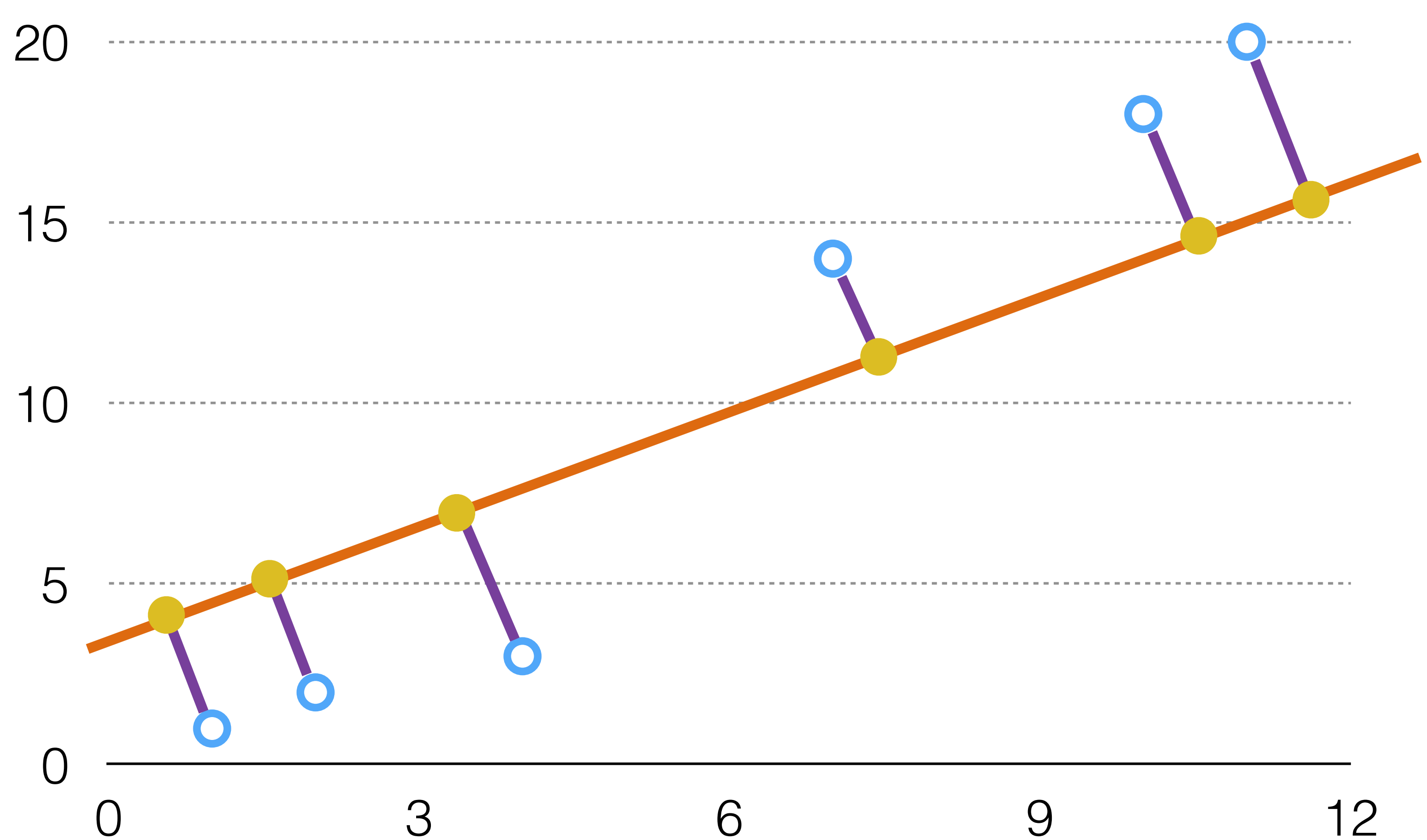
PCA

La idea principal de PCA es proyectar los datos en varias dimensiones a hiperplanos de menor dimensión



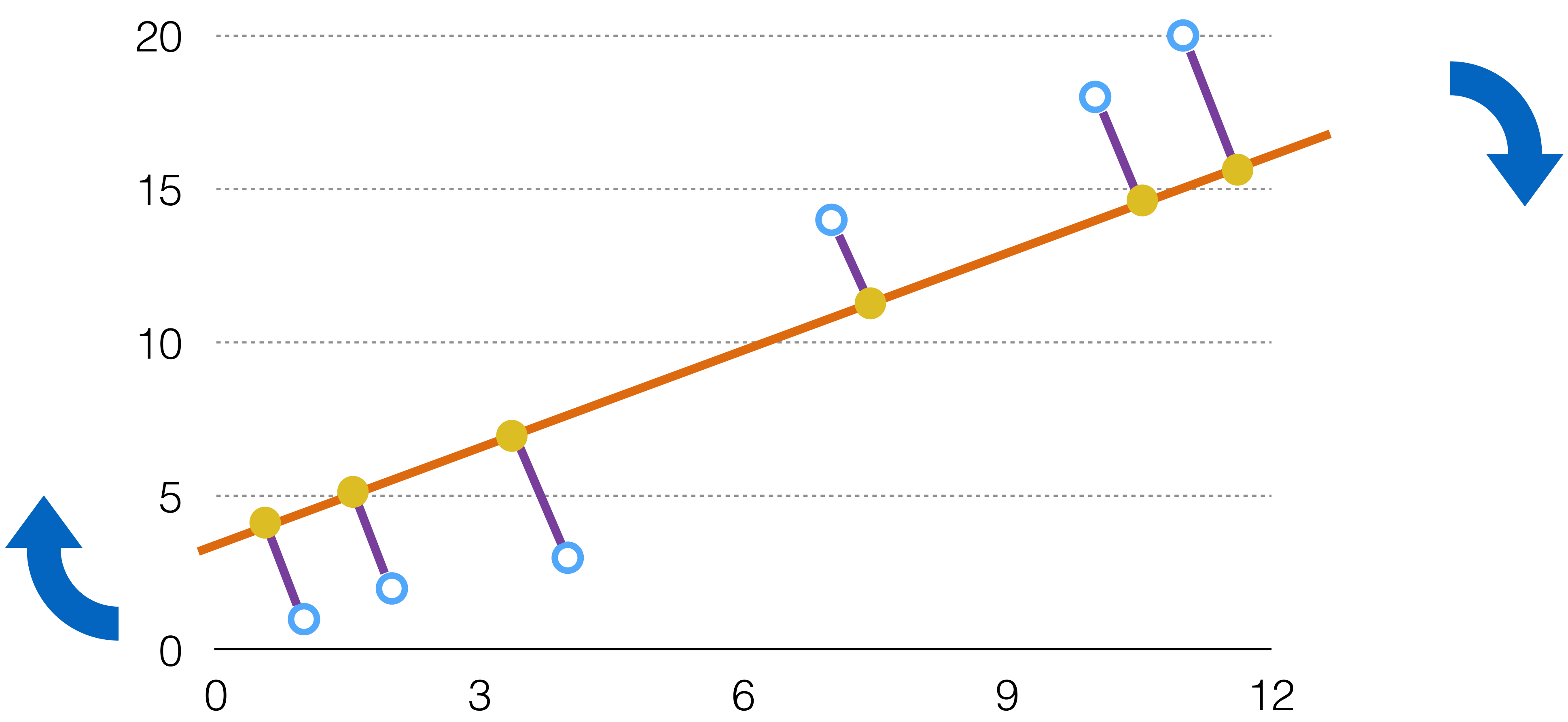
PCA

La idea principal de PCA es proyectar los datos en varias dimensiones a hiperplanos de menor dimensión



PCA

La idea principal de PCA es proyectar los datos en varias dimensiones a hiperplanos de menor dimensión



PCA

La idea principal de PCA es proyectar los datos en varias dimensiones a hiperplanos de menor dimensión

PCA

¿Cómo encontramos el mejor hiperplano sobre el que proyectar nuestros datos?

PCA

Vamos a revisar esto mediante un ejemplo, consideremos el siguiente *dataset*:

País	Renta	Venta
A	189	402
B	190	404
C	208	412
D	293	458
E	308	469
F	316	469

País	Renta	Venta
A	189	402
B	190	404
C	208	412
D	293	458
E	308	469
F	316	469

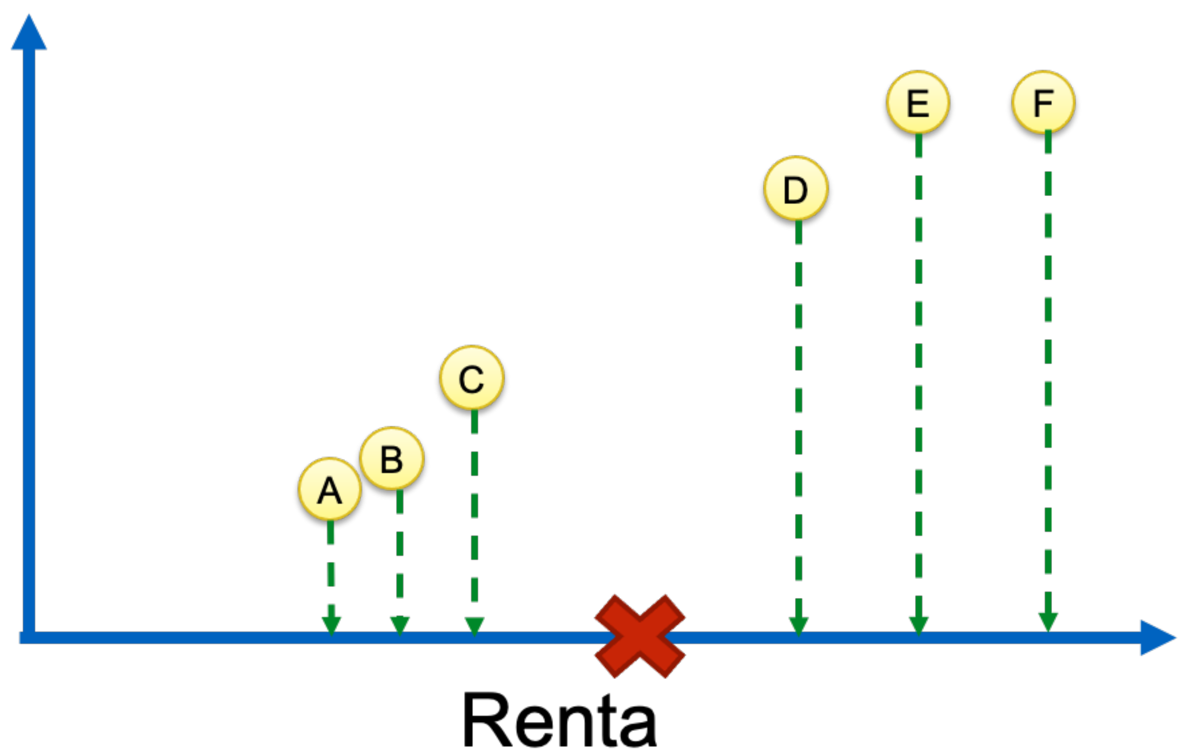
PCA

PCA requiere que los datos estén centrados en el origen

Luego de eso vamos a buscar un hiperplano sobre el que proyectar los datos

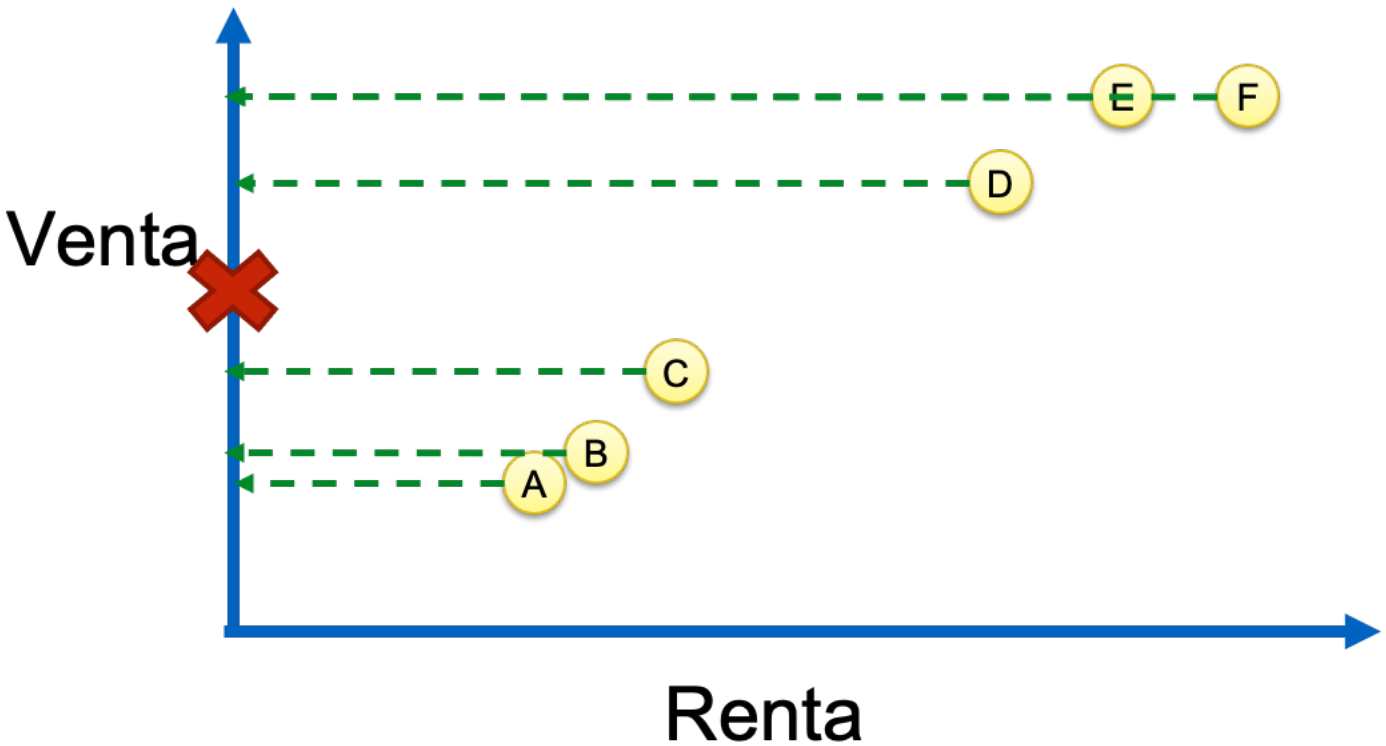
Sacamos el promedio de cada columna

País	Renta	Venta
A	189	402
B	190	404
C	208	412
D	293	458
E	308	469
F	316	469
AVG	**251**	**436**

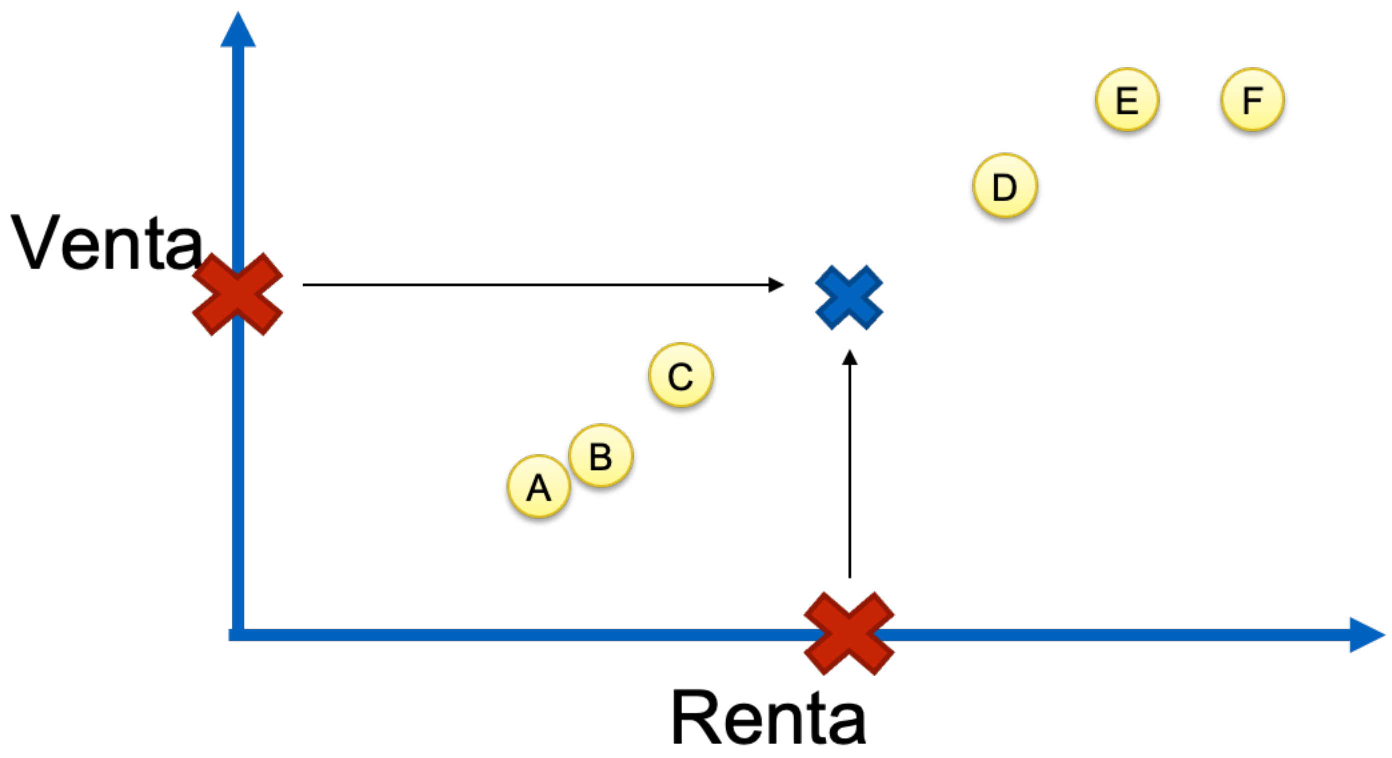


Sacamos el promedio de cada columna

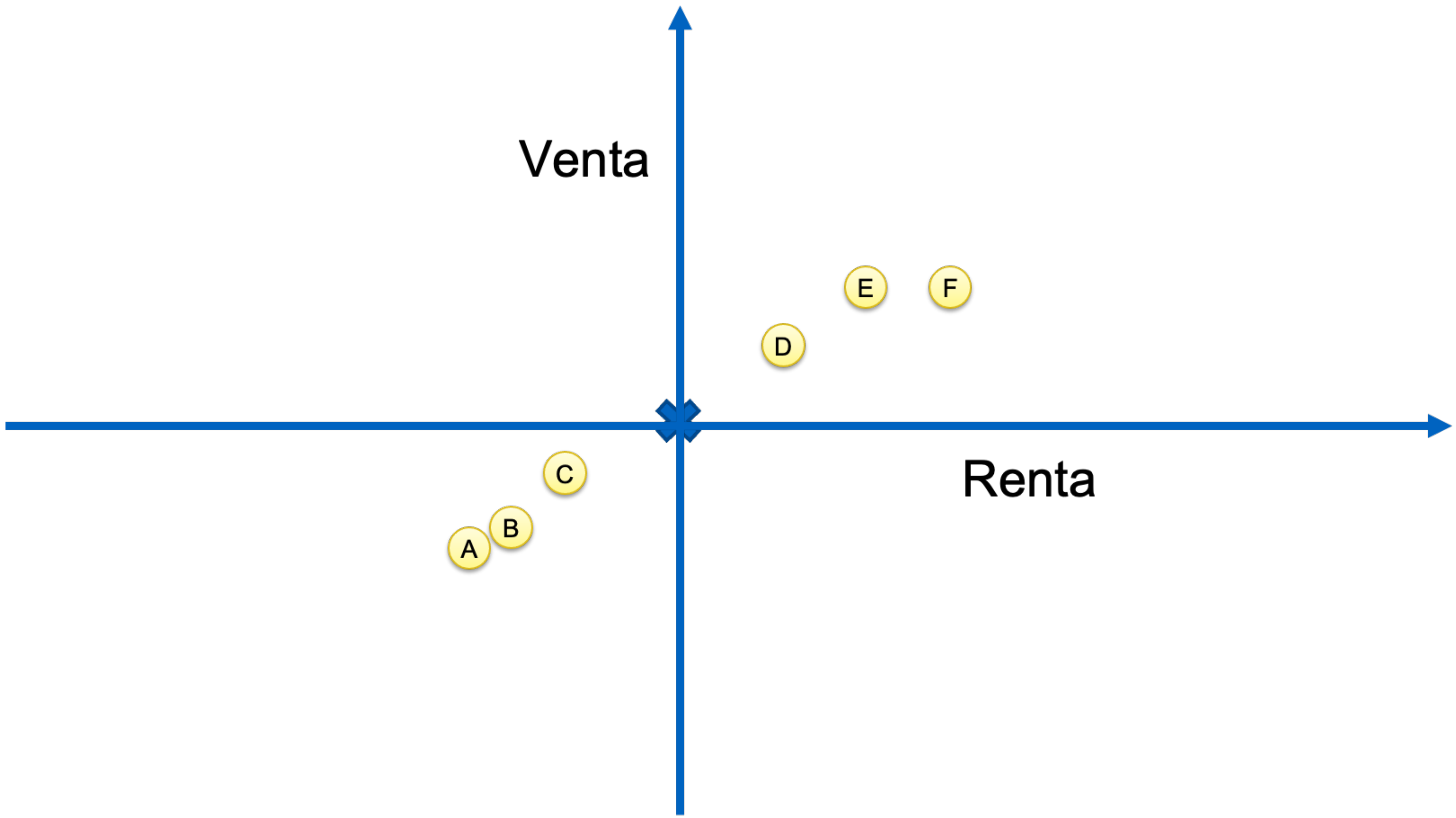
País	Renta	Venta
A	189	402
B	190	404
C	208	412
D	293	458
E	308	469
F	316	469
AVG	**251**	**436**



Con esto podemos trasladar nuestros datos para centrarlos en el origen



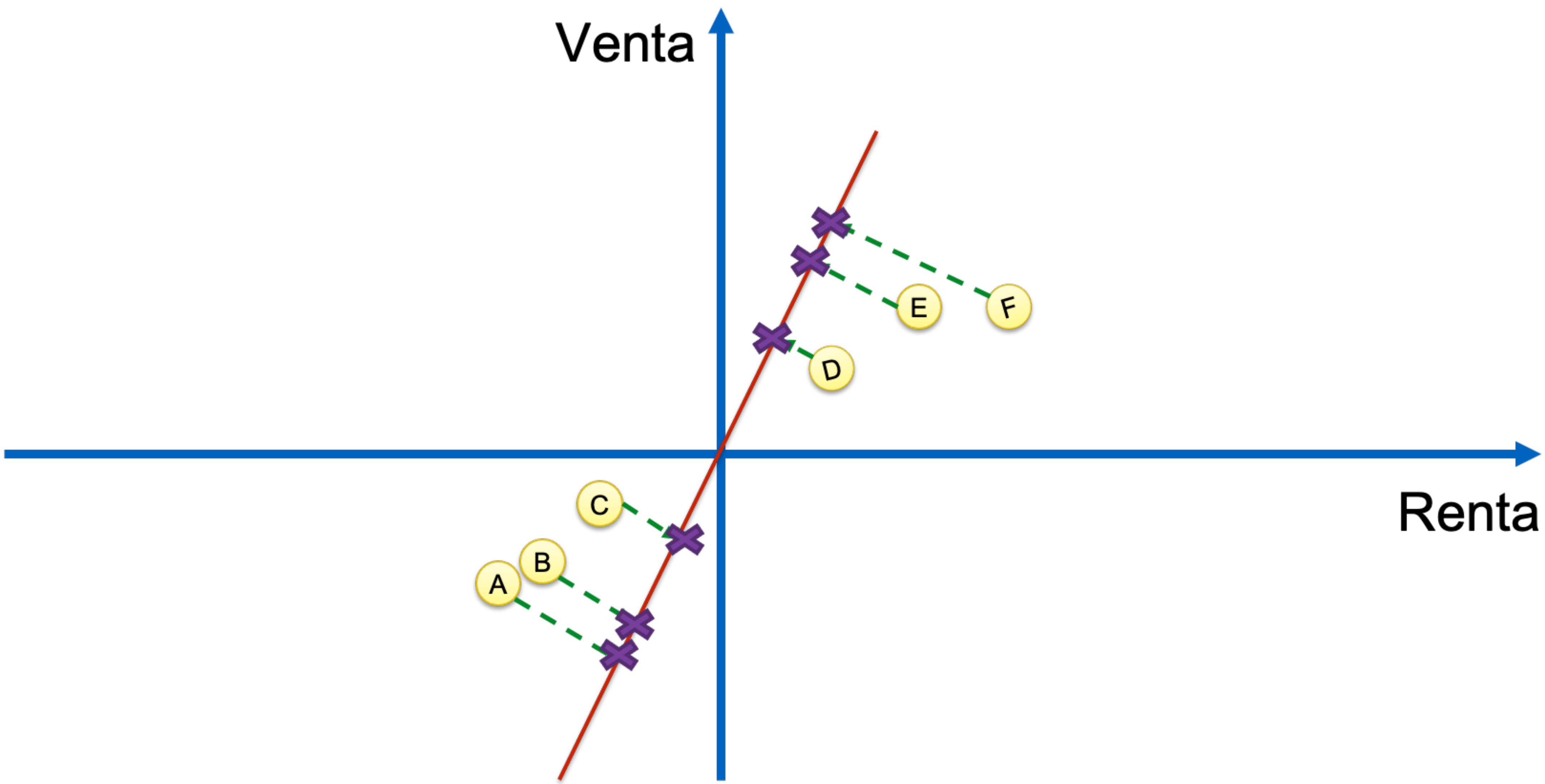
Con esto podemos trasladar nuestros datos para centrarlos en el origen



PCA

Ahora vamos a escoger una recta (hiperplano) al azar y vamos a entender si nos sirve para proyectar nuestros datos

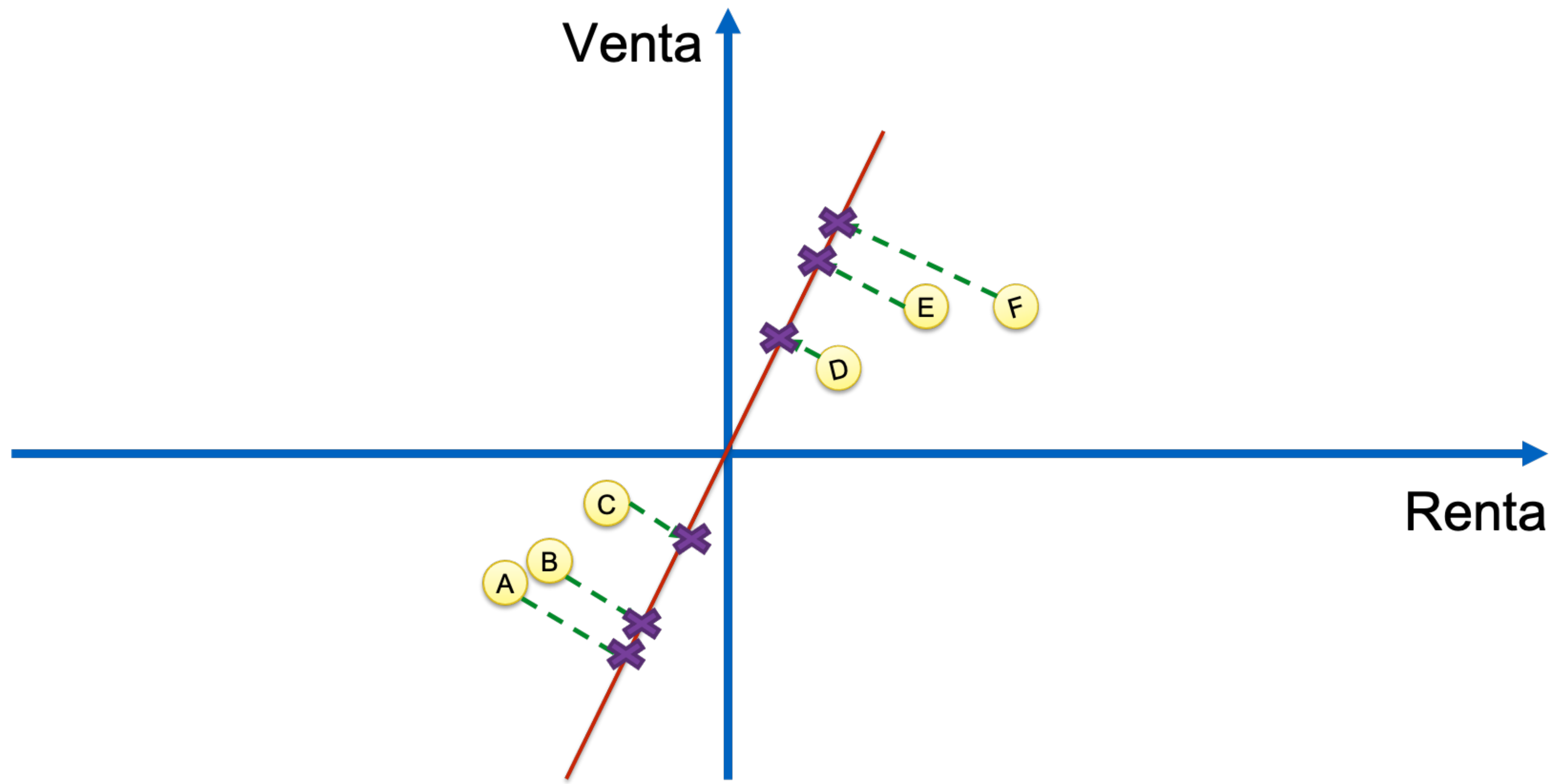
Podemos medir la suma de las distancias de las proyecciones al origen



PCA

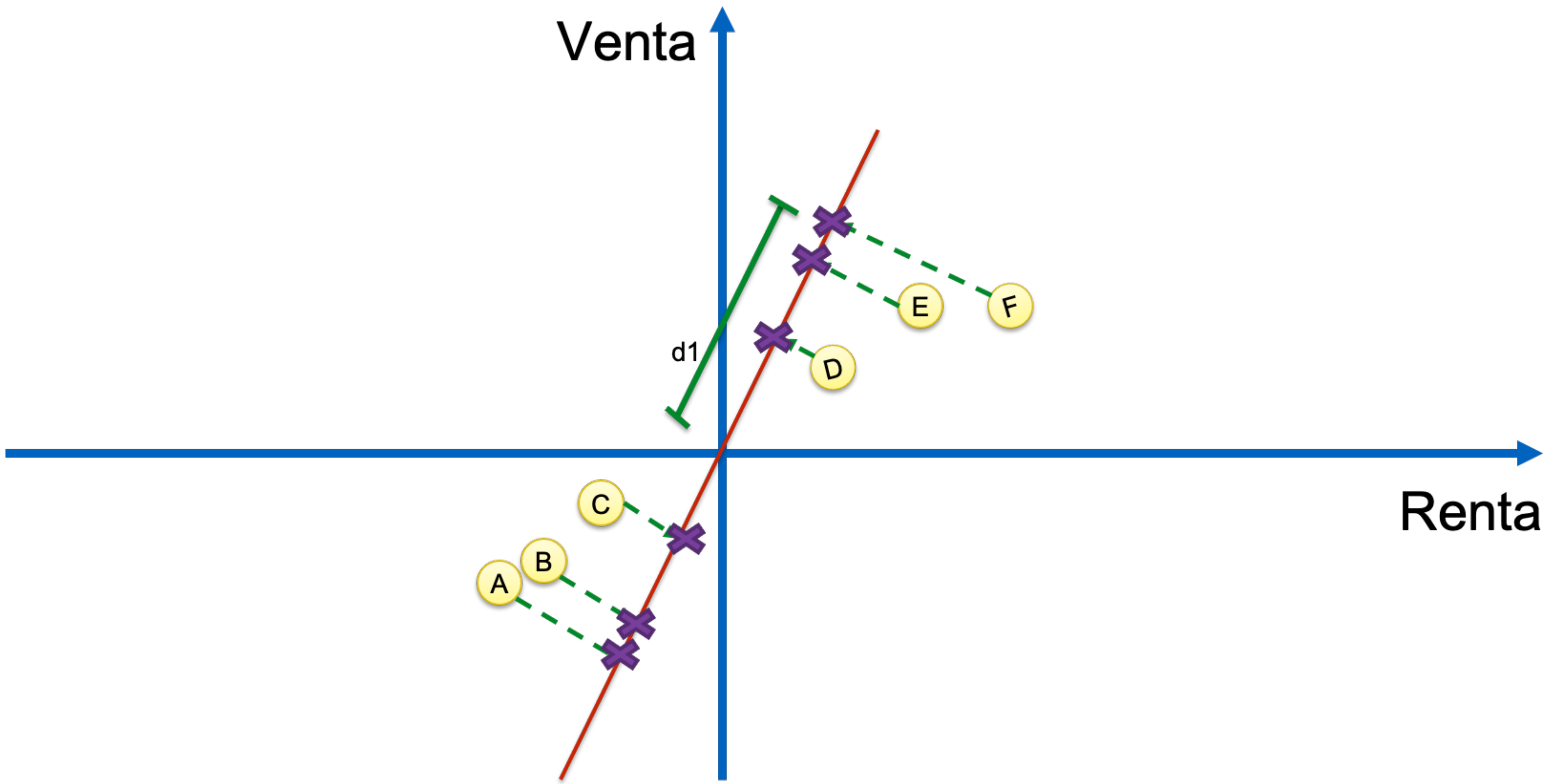
Antes de seguir, una pregunta: ¿prefieres proyectar los datos sobre una recta donde están todos los puntos juntos o donde están dispersos entre ellos?

Idea: calculamos las distancias de las proyecciones al origen e intentamos maximizarlas



Nos gustaría tener una recta que **maximice** estas distancias

En realidad lo que maximizamos es
$$SSD = d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$$



(donde SSD significa *Sum of Squares Distance*)

PCA

¿Qué logramos hasta ahora?

Teníamos un *dataset* en dos dimensiones y encontramos una recta sobre la que proyectar estos puntos

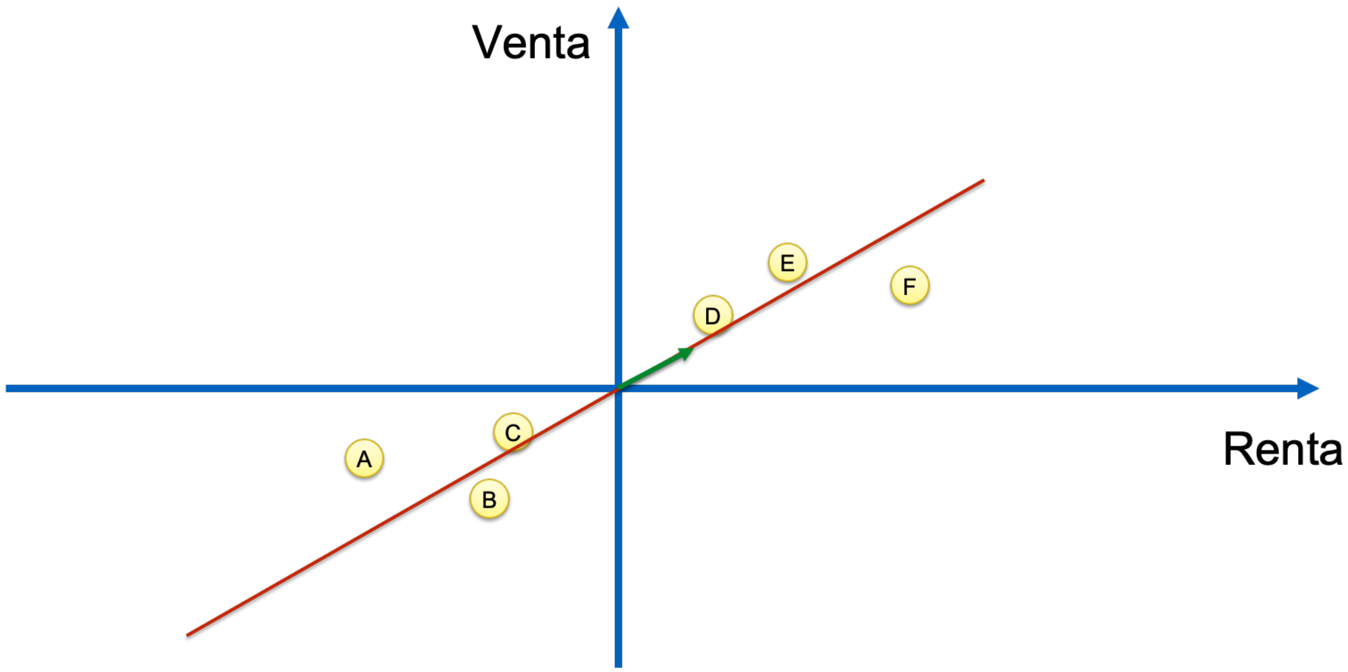
Así, **pasamos de dos dimensiones a una dimensión**

Recordemos que la recta que buscamos maximiza la distancias de las proyecciones al origen

PCA

Componentes Principales

Este "eje" que maximiza la varianza la conocemos como **Componente Principal 1** (PC1) por las siglas en inglés



Ojo. Asociado a este eje hay un vector unitario que estudiaremos más adelante

PCA

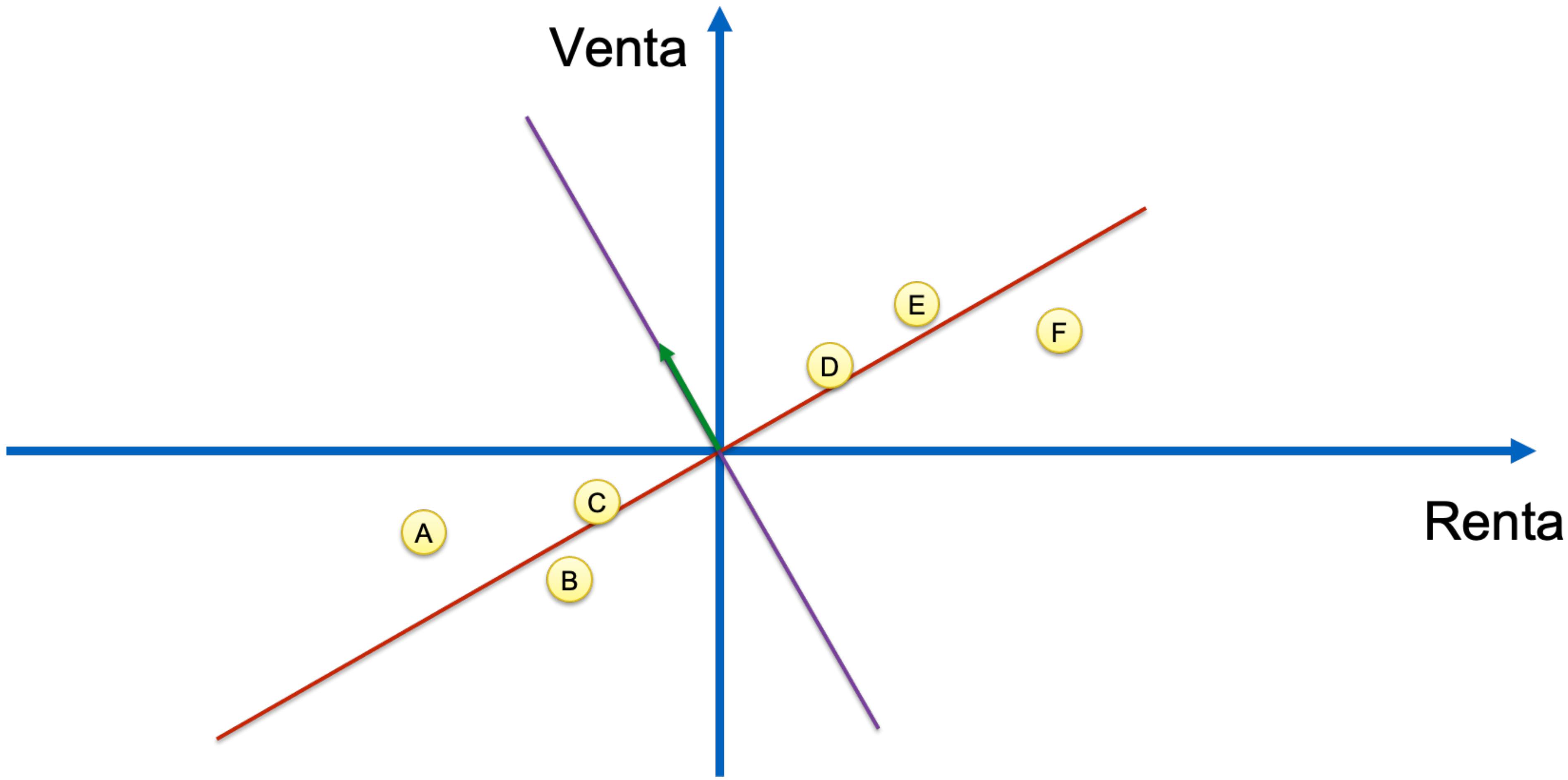
Componentes Principales

Luego si en vez de llevar el *dataset* a una dimensión queremos dos, buscamos **PC2**

Seguimos el mismo procedimiento, pero asegurándonos que **PC2** sea ortogonal a **PC1**

PCA

Componentes Principales

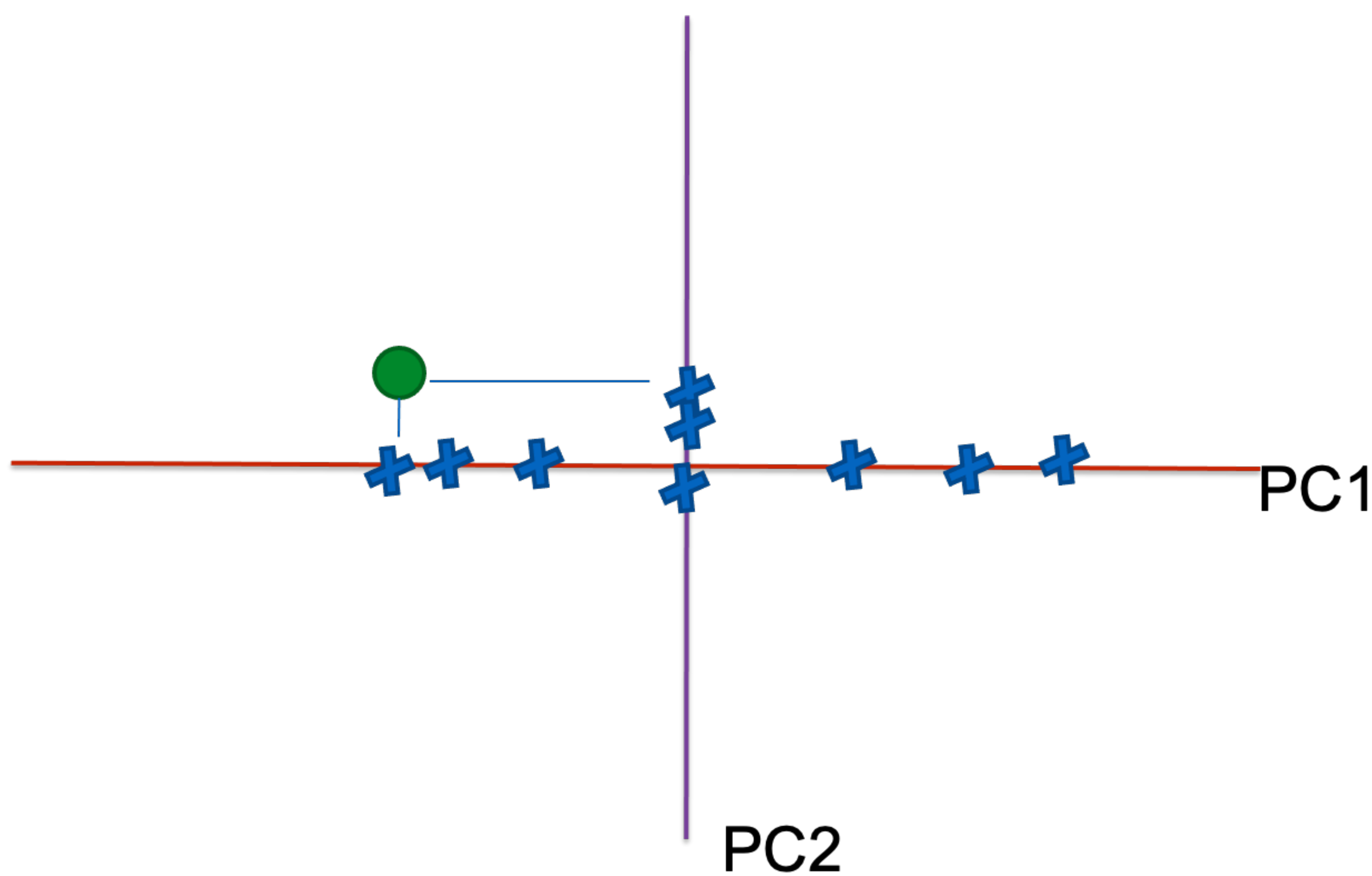


Ojo. en este caso partimos de dos dimensiones, así que volver a dejar el *dataset* en dos dimensiones es solo para mostrar cómo se buscaría **PC2**

PCA

Componentes Principales

Aquí proyectamos cada punto en los ejes, rotamos, y tenemos nuestra transformación



PCA

Componentes Principales

Si tenemos m dimensiones, podemos transformar nuestro *dataset* a uno de $d \leq m$ dimensiones

Para esto, tomamos las primeras d componentes principales y proyectamos los puntos

Además, como sabemos el valor SSD de cada componente, sabemos **el porcentaje de variación que acumula**

PCA

Encontrando las componentes

Ahora en la práctica, ¿cómo encontramos las componentes principales?

Votemos:

a) Probamos todas las combinaciones de rectas para proyectar nuestros datos
b) Usamos una técnica de álgebra lineal

PCA

Descomposición de valores singulares

Para encontrar los vectores unitarios que nos dan los ejes para proyectar los datos usamos la descomposición de valores singulares

Sea X nuestro *dataset*, buscamos expresar X como:

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

Estudiaremos cada término a continuación

PCA

Descomposición de valores singulares

En $X = U\Sigma V^T$, tenemos que:

- U son los vectores propios de XX^T
- V son los vectores propios de X^TX
- Σ es una matriz diagonal que tiene los valores singulares de X (los valores singulares son las raices cuadradas de los valores propios de X^TX o XX^T)

PCA

Descomposición de valores singulares

¿Qué tiene que ver esto con PCA?

Las columnas de V son los vectores que nos dan los ejes sobre los que vamos a proyectar nuestros datos

Los valores singulares nos señalan "la varianza que guarda cada columna"

PCA

Descomposición de valores singulares

Muy importante! Los valores singulares y los vectores de V vienen ordenados del más importante al menos importante

Por lo mismo, sabemos que **PC1** va a guardar más varianza que **PC2**, y **PC2** va a guardar más varianza que **PC3**, y así sucesivamente

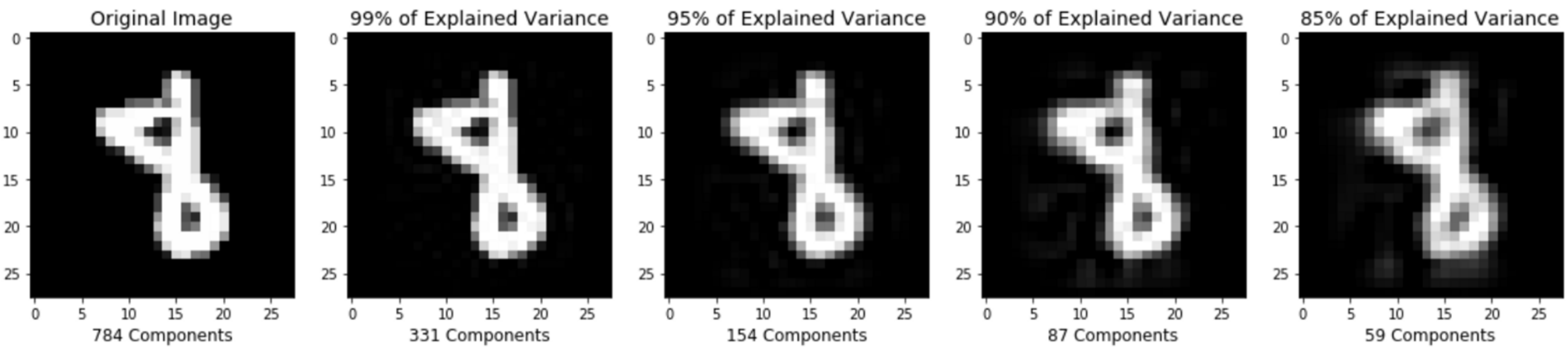
PCA

Escogiendo el número de dimensiones

Como sabemos el porcentaje de la varianza total que acumula cada componente, podemos comprimir el *dataset* hasta guardar gran parte de la varianza (e.g. 95%)

PCA
Ejemplo

Podemos comprimir imágenes con PCA



Hay varios detalles adicionales que vamos a revisar en el código

Fundamentos de Ciencias de Datos

Semana 10 - Resumen Clasificador MNIST