Aprendizaje No Supervisado: K-Means

CURSO GRUPO BANCOLOMBIA

Universidad Nacional de Colombia

K-Means

Los algoritmos de agrupamiento buscan aprender, a partir de las propiedades de los datos, una división óptima o un etiquetado discreto de grupos de puntos.

El algoritmo k-Means busca un número predeterminado de clústeres dentro de un conjunto de datos multidimensional sin etiquetar.

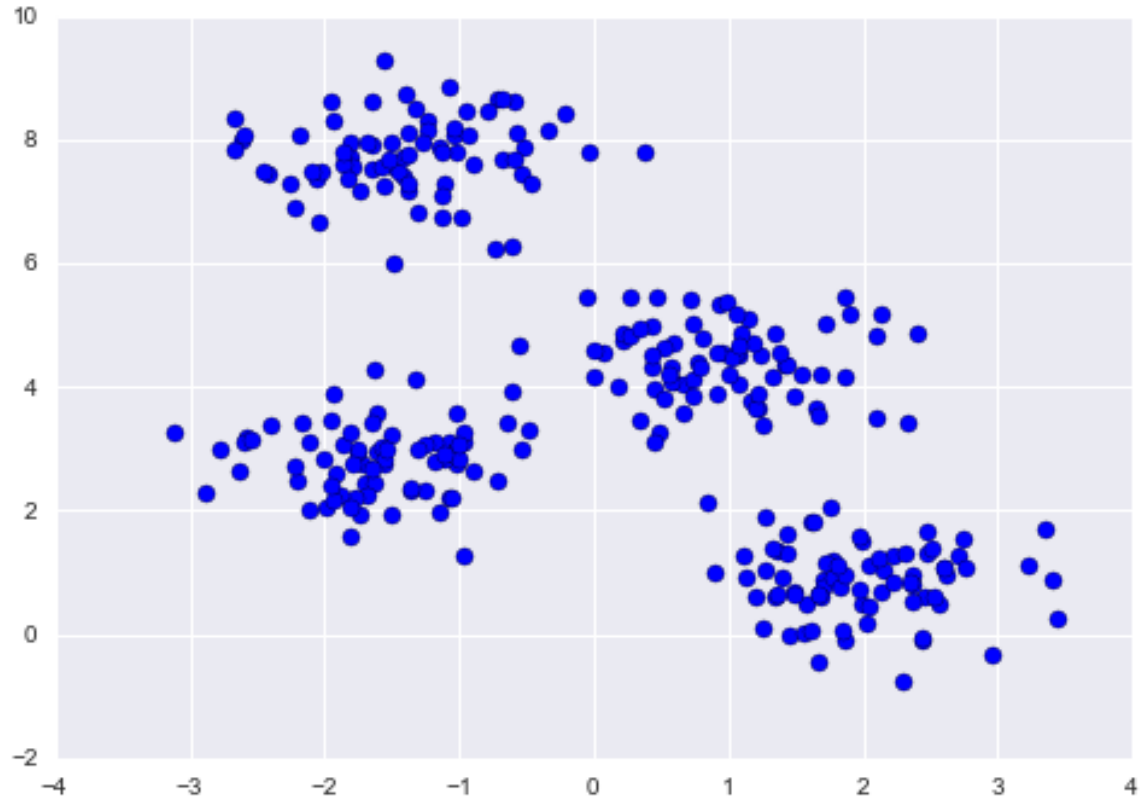
Se logra así:

- El "centro del grupo" es la media aritmética de todos los puntos que pertenecen al grupo.
- Cada punto está más cerca de su propio centro de clúster que de otros centros de clúster.

Esos dos supuestos son la base del modelo k-means.

Ejemplo

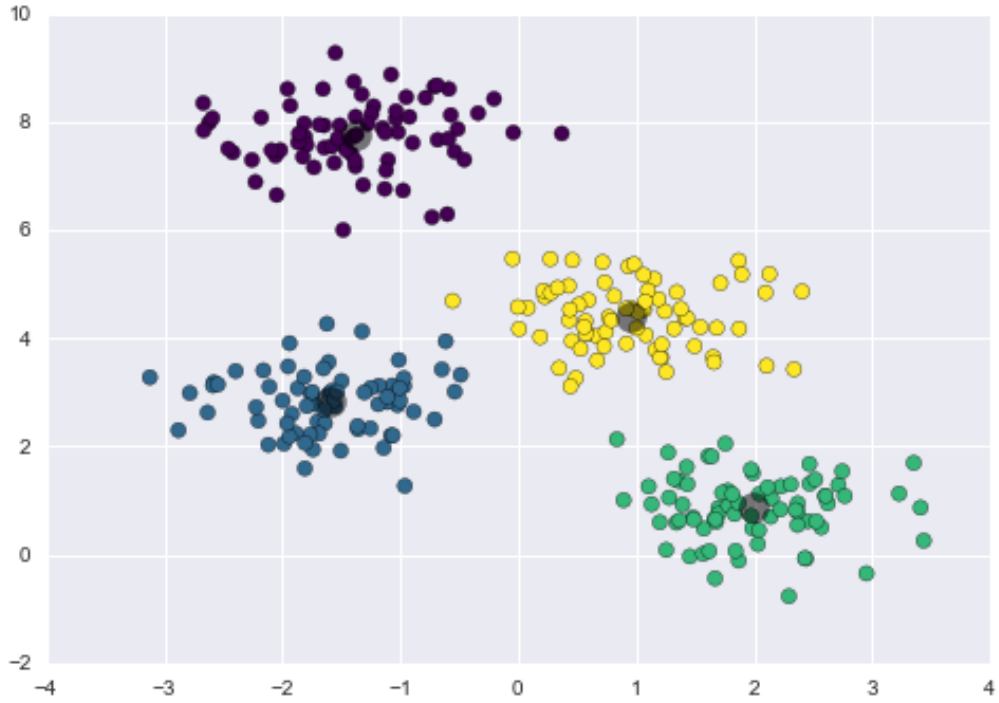
Se genera un conjunto de datos bidimensional. Debido a que es un algoritmo no supervisado, se dejan las etiquetas fuera de la visualización



Tomado de: Python Data Science Handbook, Jake VanderPlas

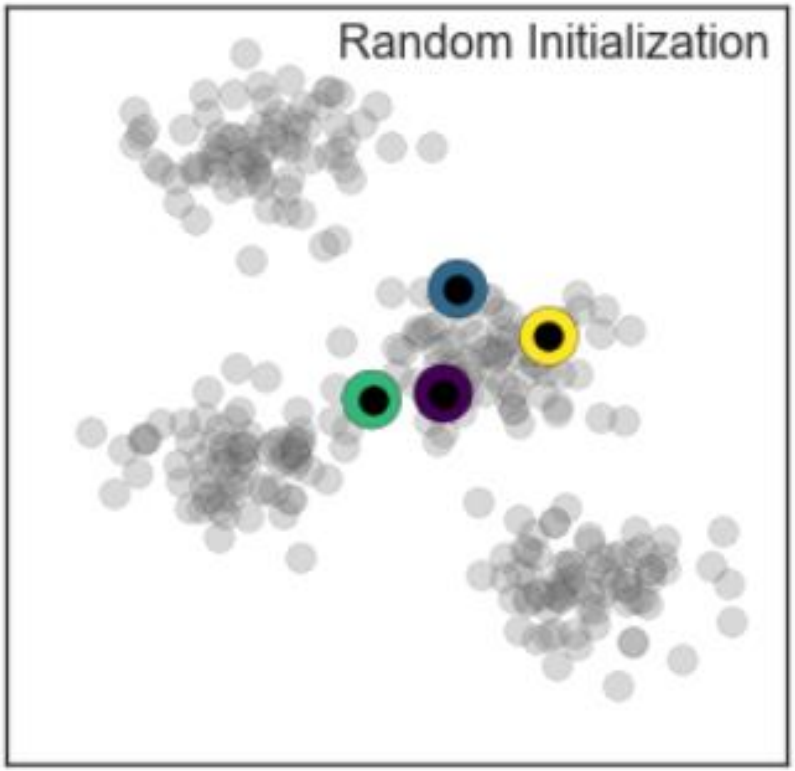
Ejemplo

- Es fácil elegir visualmente los cuatro grupos.
- El algoritmo k-means asigna los puntos a los clústeres de manera muy similar a cómo podríamos asignarlos a simple vista.
- El algoritmo k-means hace esto automáticamente.
- Se trazan los centros de clúster según lo determine el estimador.



Tomado de: Python Data Science Handbook, Jake VanderPlas

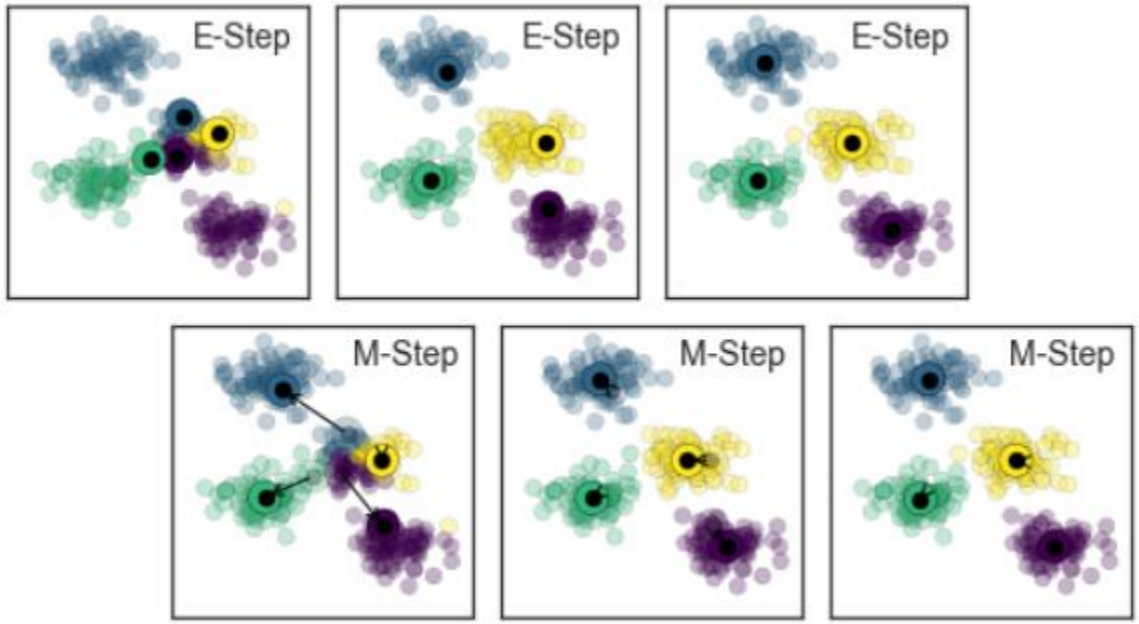
Los centros iniciales son aleatorios



Tomado de: Python Data Science Handbook, Jake VanderPlas

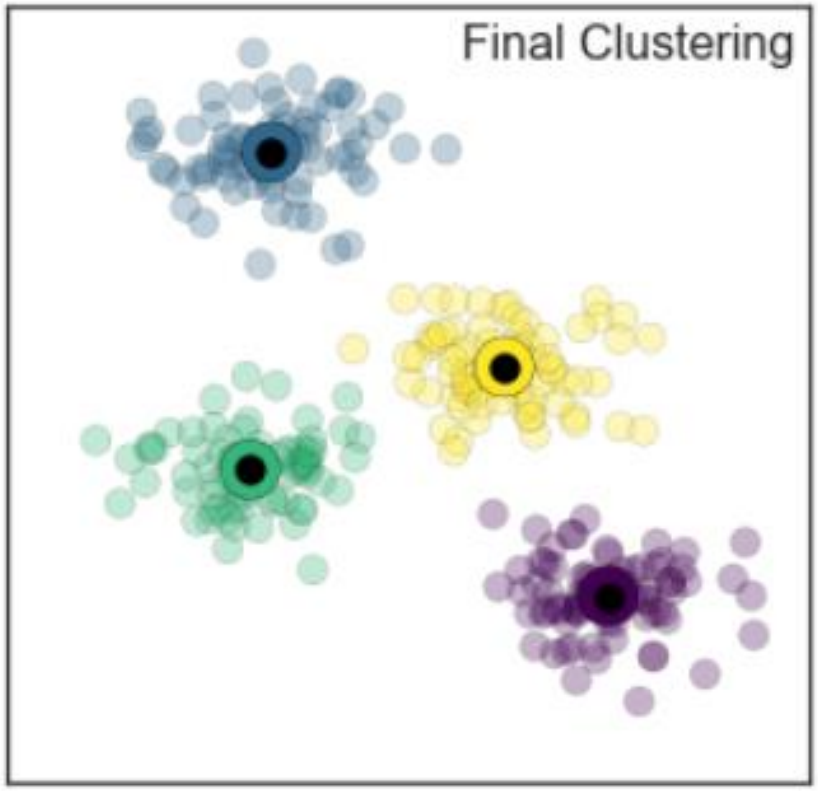
Desarrollo del algoritmo

- E-Step: asigna los puntos al centro de clúster más cercano. También llamado "paso de expectativa" implica actualizar a qué grupo pertenece cada punto.
- M-Step: establece los centros del clúster en la media.
- Se repite hasta converger.



Tomado de: Python Data Science Handbook, Jake VanderPlas

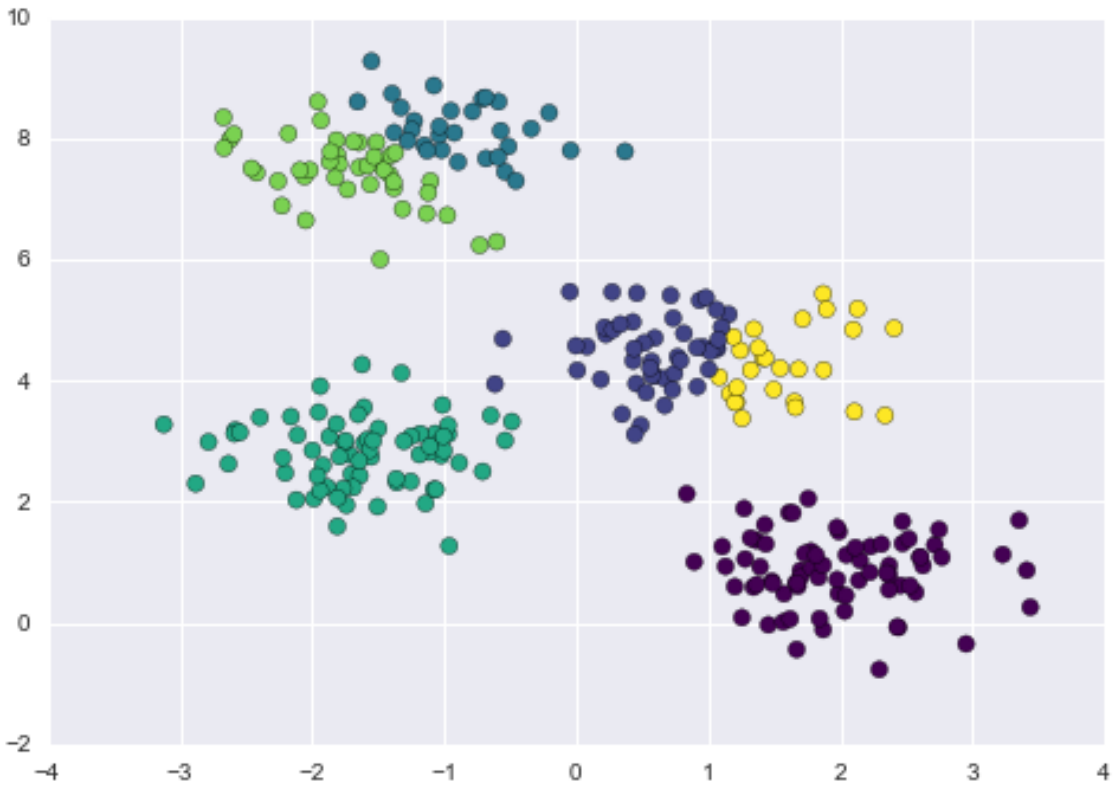
Resultado



Tomado de: Python Data Science Handbook, Jake VanderPlas

Ejemplo 6 grupos

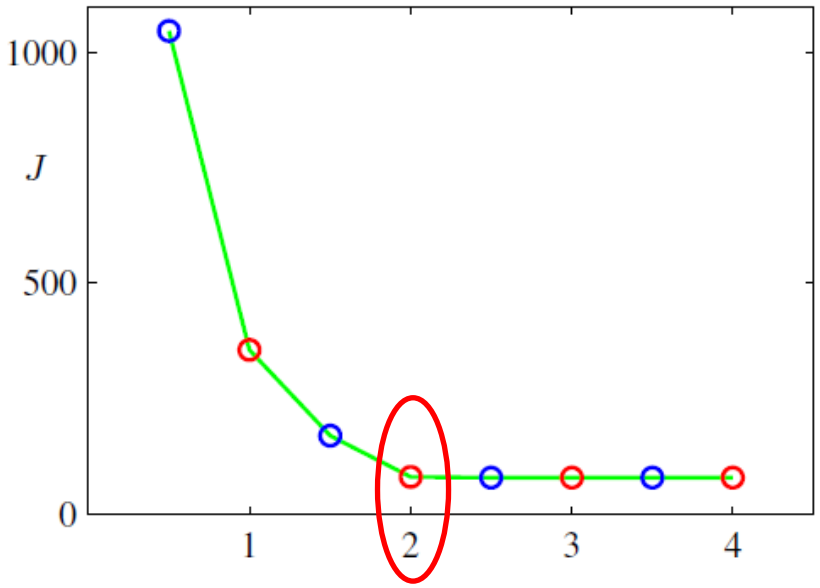
El número de grupos debe seleccionarse de antemano, el algoritmo procederá a encontrará los mejores 6 grupos



Tomado de: Python Data Science Handbook, Jake VanderPlas

¿Cómo seleccionar el valor de K?

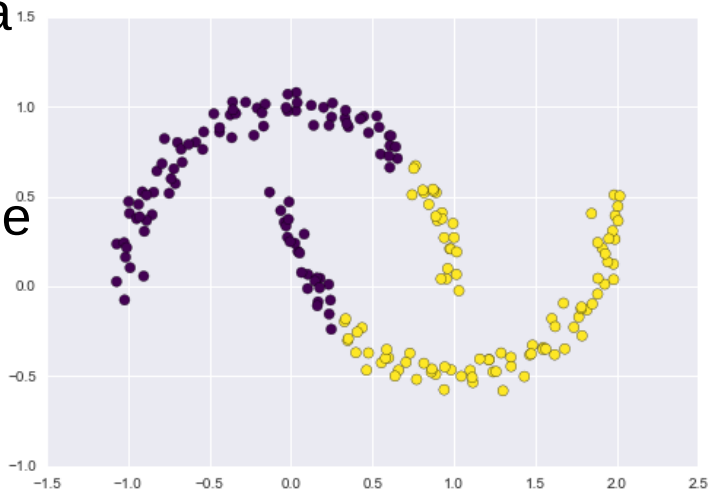
- Se ejecuta el algoritmo para diferentes valores de K y se gráfica la **Suma del error cuadrático** (S con respecto a K.
- El SSE es la suma de las distancias, al cuadrac de las muestras a su clúster asignado.
- $SSE = \sum_{i=1}^{N}(x_i - \bar{x})$, donde N es el número de datos, x_i es el valor de cada punto y $\bar{x}$ es el val del centro del clúster al que pertenece.
- Se selecciona el valor de K para el codo de la gráfica (K=2).
- Esto porque SSE disminuirá a medida que se



Tomado de: Python Data Science Handbook, Jake VanderPlas

Limitaciones lineales

- Los supuestos fundamentales del modelo de k-Means, significa que el algoritmo será ineficaz si los grupos tienen geometrías complicadas.
- En particular, los límites entre los grupos de k-Means siempre serán lineales, lo que significa que fallará para los límites más complicados.
- El algoritmo puede ser lento para grandes cantidades de datos, porque en cada iteración debe acceder a todos los datos.

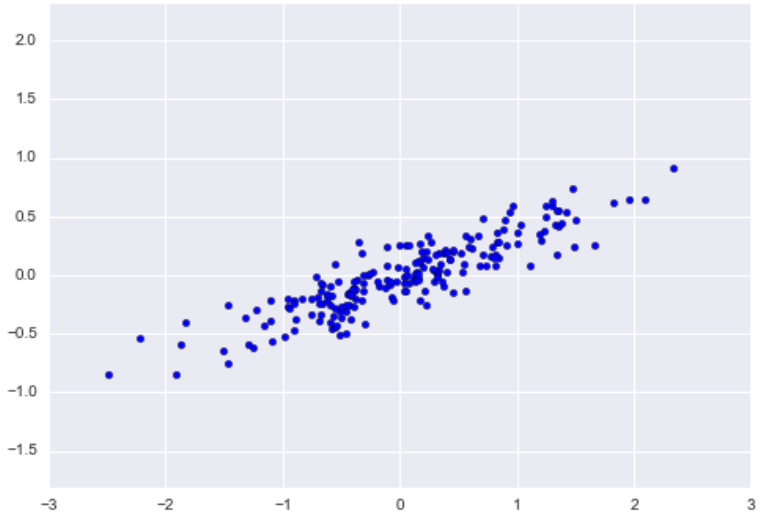


Tomado de: Python Data Science Handbook, Jake VanderPlas

Análisis de componentes principales (PCA)

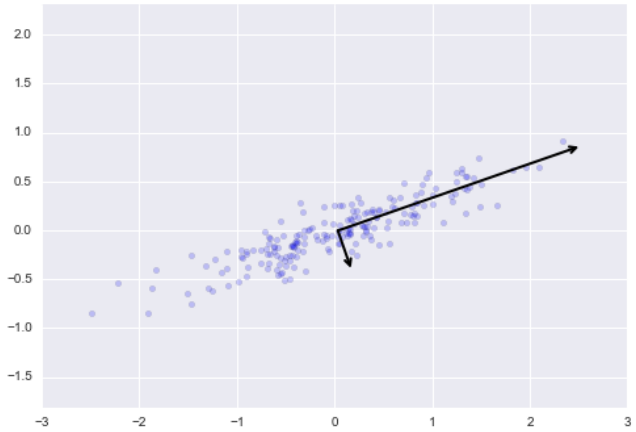
PCA

- Es un estimador no supervisados, que puede resaltar aspectos interesantes de los datos.
- Considere los siguientes 200 puntos.
- En lugar de intentar predecir los valores de y a partir de los valores de x, el algoritmo intenta aprender sobre la relación entro los valores de x y y.
- Esta relación se cuantifica al encontrar una lista de los ejes principales en los datos y al usar dichos ejes para describir el coniunto de los datos



PCA

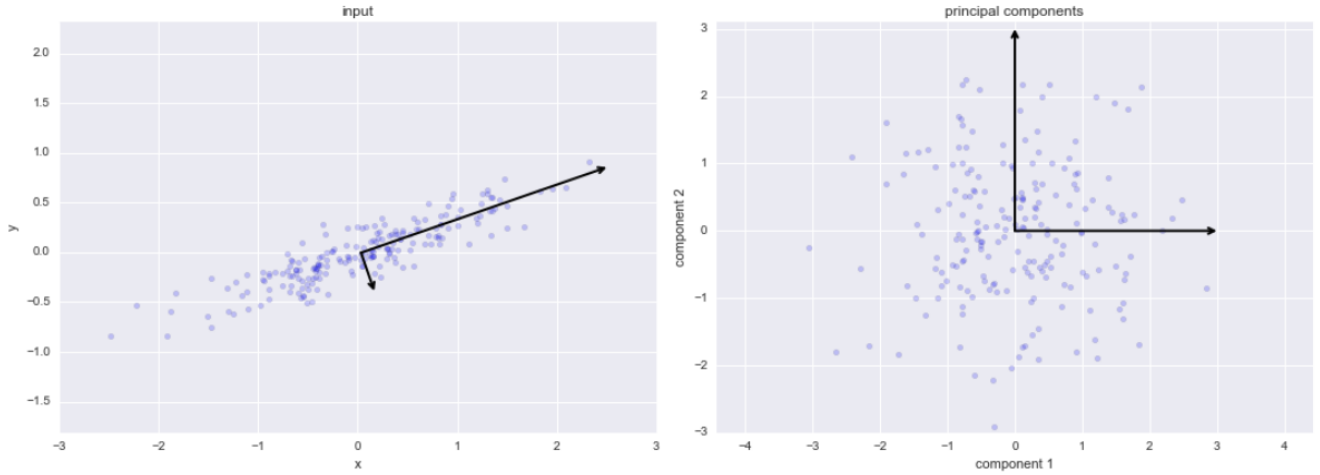
- Se visualizan los vectores sobre los datos de entrada.
- Los vectores representan los ejes principales de los datos. Se usan las "componentes" para definir la dirección del vector y la "varianza" para definir la longitud del vector.
- La varianza representa cuán "importante" es ese eje para describir la distribución de los datos.
- La proyección de cada punto de datos en los ejes principales son los "componentes principales" de los datos.



Tomado de: Python Data Science Handbook, Jake VanderPlas

PCA

- Se pueden ver las gráficas de los datos originales y los componentes principales.
- La transformación de los ejes de los datos a los ejes de los componentes principales es una transformación afín, lo que básicamente significa que se compone de una traslación, rotación y escala uniforme.



Tomado de: Python Data Science Handbook, Jake VanderPlas

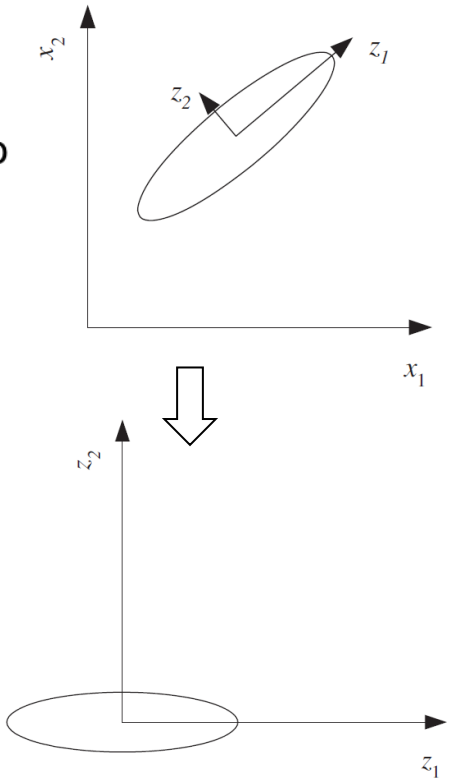
PCA como reducción de dimensionalidad

- Se ponen en cero los componentes principales de menor longitud, es decir, los ejes con varianza más pequeña.
- Resulta en una proyección de menor dimensión de los datos que conserva la máxima varianza.
- Para comprender el efecto de esta reducción de dimensionalidad, podemos realizar la transformación inversa de estos datos reducidos y trazarlos junto con los datos originales

Tomado de: Python Data Science Handbook, Jake VanderPlas

PCA

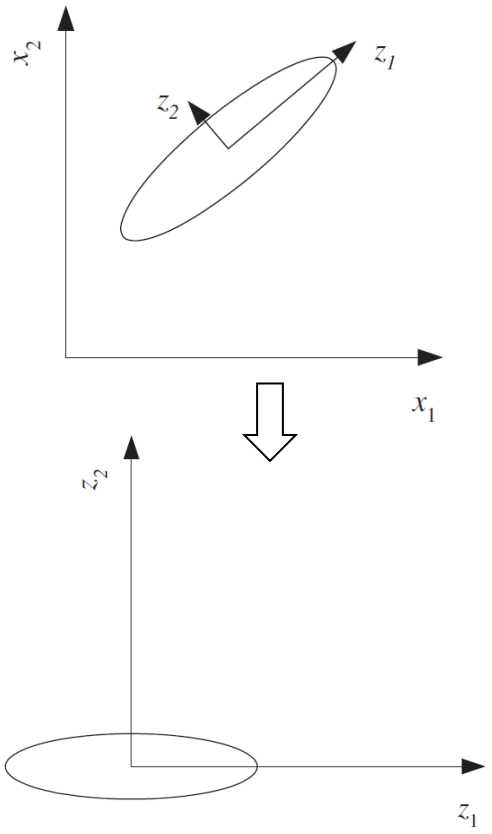
- El criterio a maximizar en PCA es la varianza, es decir, la variabilidad que representan los datos
- Se mapean los datos en un espacio d-dimensional a un nuevo espacio $(k < d)$-dimensional
- La proyección de x en la dirección de w es: $z = w^T x$.
- w son componentes principales: $w = \begin{vmatrix} \vdots & \vdots & \vdots \\ w_1 & w_2 & \dots\ w_m \\ \vdots & \vdots & \vdots \end{vmatrix}$
- Entonces cuando los datos se proyectan en w, la muestra está más extendida y la diferencia entre las clases se hace más evidente. Para una solución única y evidente se espera que $\|w_1\| = 1$.
- Se tiene que la varianza $Var(z_1) = w_1^T \sum w_1$.
- Se busca w_1 de modo que $Var(z_1)$ sea máxima, sujeto a la restricción $w_1^T \sum w_1 = 1$.



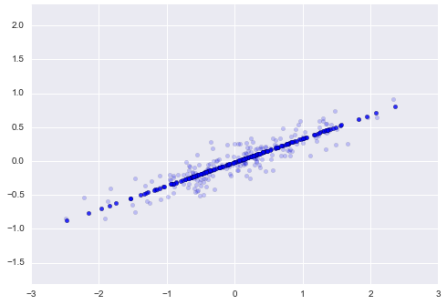
Tomado de: Introduction to Machine Learnin, Ethem Alpaydin, 2010

PCA

- Para que $Var(z_1)$ se maximice se aplica la deriva.
- Así: $z = w^T(x - m)$
- Los k vectores con valores propios distintos de cero son las dimensiones del espacio reducido.
- El primer vector propio w_1, es el componente principal que explica la mayor parte de la varianza; el segundo explica el segundo más grande; y así.
- Después de esta transformación lineal, llegamos a un espacio k-dimensional cuyas dimensiones son los vectores propios, y las variaciones sobre estas nuevas dimensiones son iguales a los valores propios.



Tomado de: Introduction to Machine Learnin, Ethem Alpaydin, 2010



- Los puntos claros son los datos originales, mientras que los puntos oscuros son la versión proyectada.
- Se elimina la información a lo largo del eje o ejes principales menos importantes, dejando solo el componente (s) de los datos con la mayor varianza.
- La fracción de varianza que se corta (proporcional a la extensión de puntos sobre la línea formada en esta figura) es aproximadamente una medida de cuánta "información" se descarta en esta reducción de dimensionalidad.
- Este conjunto de datos de dimensión reducida es en algunos sentidos "lo suficientemente bueno" para codificar las relaciones más importantes entre los puntos: a pesar de reducir la dimensión de los datos en un 50%, la relación general entre los puntos de datos se conserva principalmente.

Tomado de: Python Data Science Handbook, Jake VanderPlas

¡A codificar!