Es muy normal que en nuestros modelos de mML tengamos muchos features y que la relazion con nuestra variable de salida no sea muy importante

¿Cuando es necesario aplicar PCA?

Nuestro dataset tiene un numero alto de features y no estamos seguros de que todos puedan ser utiles

**Hay una alta correlacion entre los features**

**Overfitting para reducir la complejidad de nuestro modelo**

Alto coste computacional

Si tenemsos un buen modelo o si llegamos a un buen resultado, pero nos cuesta mucho computacionalmente. Podemos usar PCA para reducir los features y tener menor carga computacional

¿Qué hace PCA?

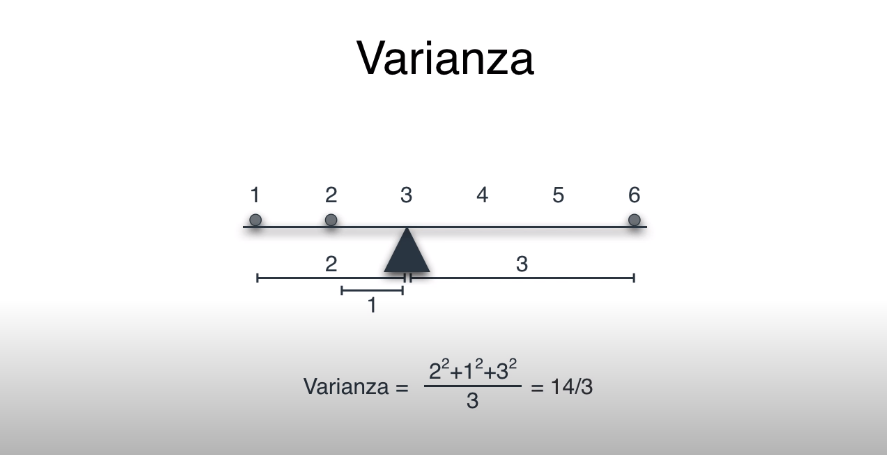
intentar combinar diferentes features en features artificiales, pero que mantengan la misma informacion (la varianza en esos features)

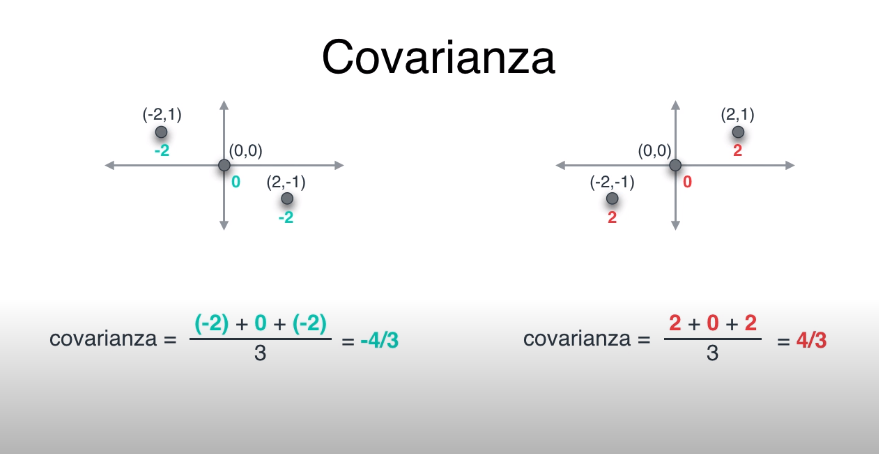
Recuerda que PCA utiliza estos conceptos:

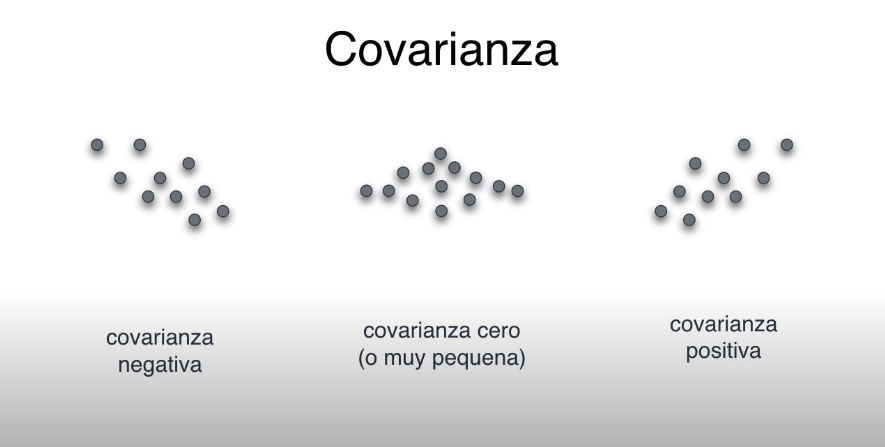
Promedio: punto medio entre 2 puntos (centroide)

Varianza: distancia entre 2 puntos al cuadrado para no tener problemas con los signos

(distancia entre puntos)

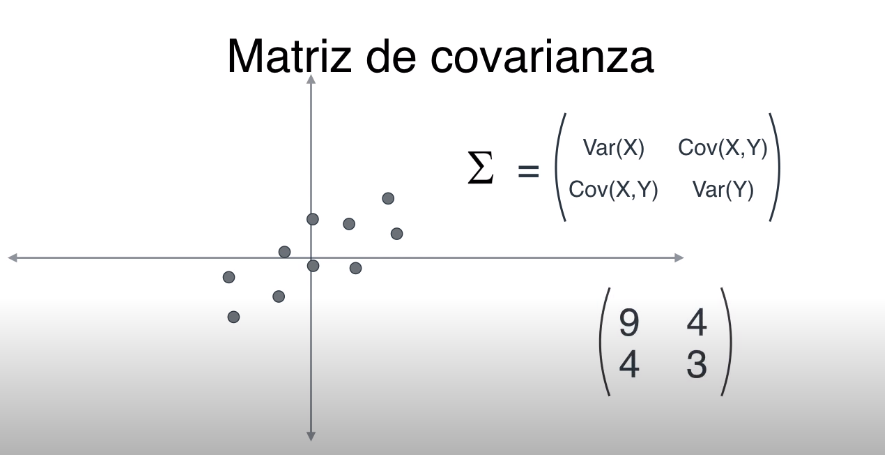


Covarianza: es el producto de las coordenadas

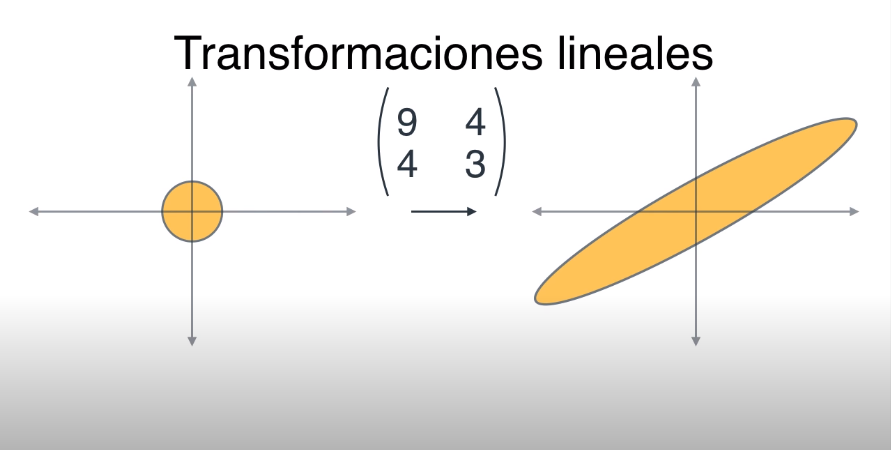


Luego de entendido esto.

Procedemos a obtener nuestra matriz de covarianza:



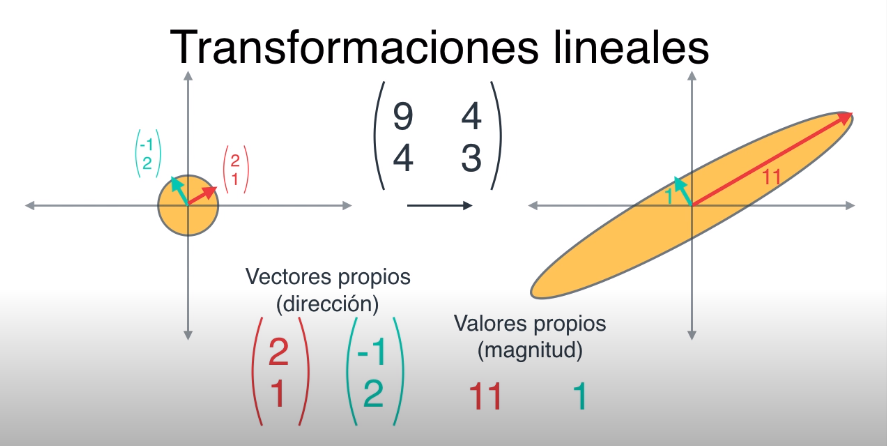
Esto nos va a ser de mucha ayuda para realizar nuestra **transformación lineal**



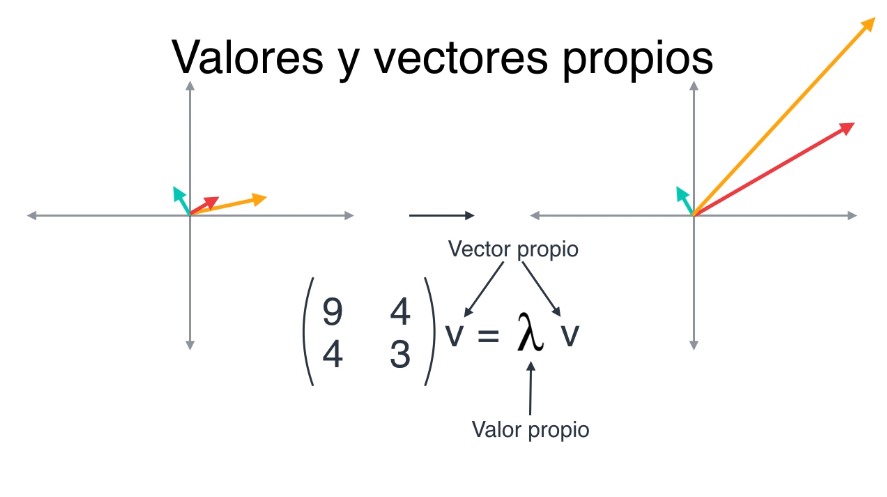
Vemos que la matriz transforma a nuestro gráfico. Esto es debido a que le estamos **haciendo un transformación lineal a cada punto** de nuestra figura.

Fíjate muy bien que hay 2 vectores que, luego de la transformación lineal, **no cambiaron de dirección.**

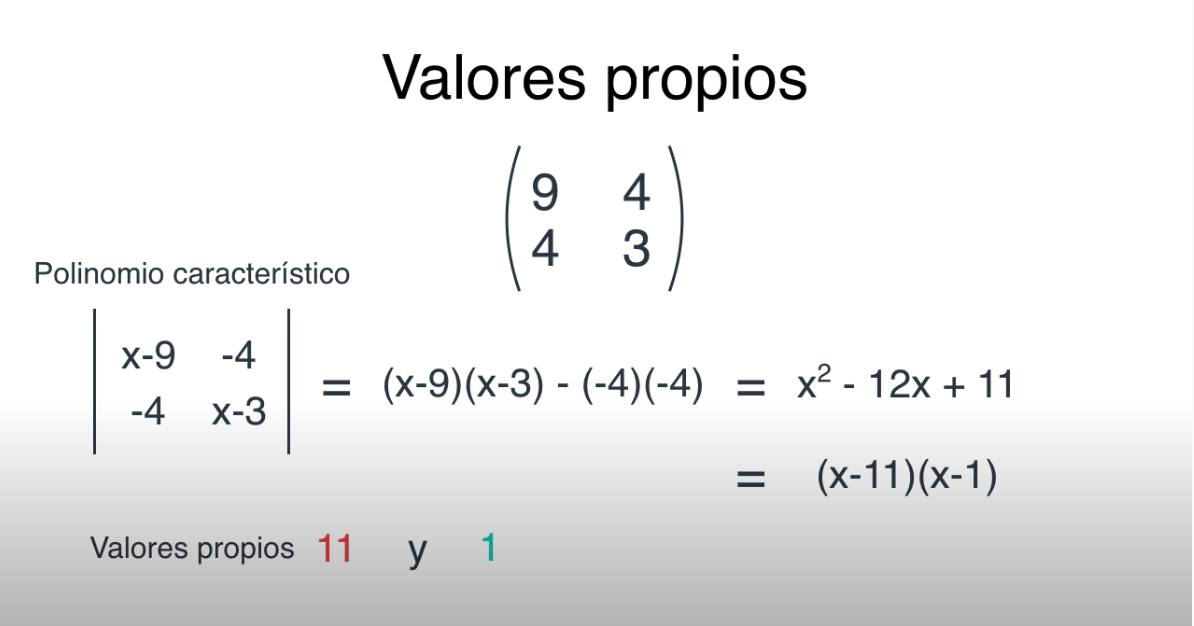
Estos vectores son **los autovectores o vectores propios**



Visto que estos vectores no cambiaron de dirección, notamos que realizar la transformación lineal a estos vectores es lo mismo que hacerle una multiplicación por un escalar, ya que solo está cambiando de magnitud. A estos escalares los llamamos **autovalores.** Recuerda que es un autovalor para cada autovector.



Lo podemos hallar de la siguiente manera:



Visto ello, podemos observar que hay un vector en cuya **dirección hay mayor captura de puntos**, por lo que tomaremos este vector para **proyector todos esos puntos y reducir nuestra dimensión** a 1.

Estamos implementando **PCA**

