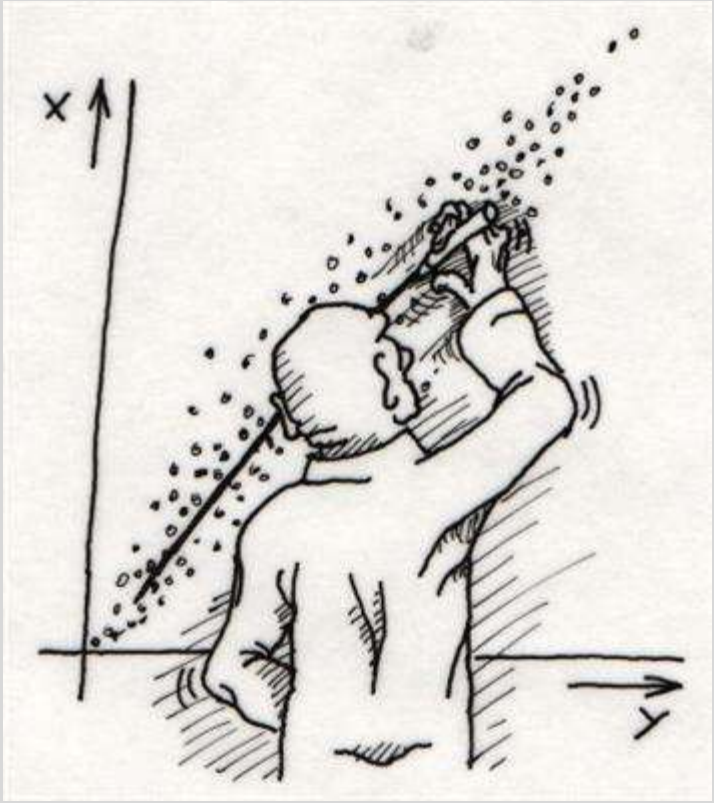


DigitalHouse > Coding School

DATA SCIENCE

UNIDAD 2
MÓDULO 3

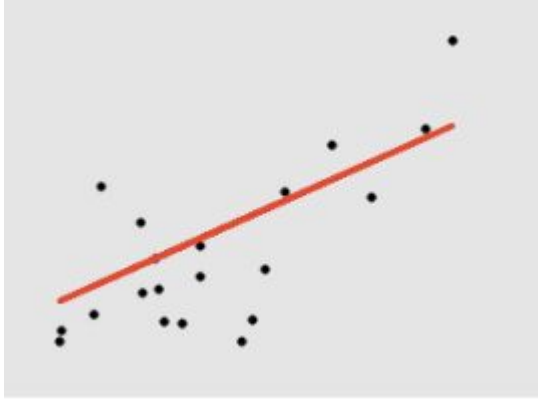
Normalización

Septiembre 2017

NORMALIZACIÓN

1. **Entender el concepto y utilidad de la normalización de datos**

2. **Usar el módulo de preprocesamiento de scikit-learn para normalizar datos**



NORMALIZACIÓN

¿Por qué normalizar?

- Manejo de cantidades en **diferentes unidades o escalas**
- Muchos **algoritmos** de machine learning toman la normalización como **requerimiento**
- Existen distintas razones por las cuales un algoritmo de ML requiere estandarizar los datos. Dependiendo

Muchos algoritmos de ML se basan en el cálculo de medidas de distancia que se calculan entre todos los puntos en base a distintos features.

La medida de distancia que viene implementada por default, es la distancia euclídea que requiere matemáticamente que todos los features sean numéricos.

Además, para no favorecer a ningún feature en particular a la hora de explicar la distancia, tenemos que estandarizar y deshacernos de las unidades.

Cómo normalizar eficazmente es uno de los desafíos más grandes a la hora de implementar modelos de ML y también uno de los errores más comunes a la hora de trabajar.

Para no cometer errores en este campo, es importante entender bien cómo funciona cada algoritmo.

A continuación dejamos un resumen con las técnicas que vamos a ver durante el curso y cómo se vinculan con la normalización.

> No es necesario entender todos los detalles de cada algoritmo ahora. Sí es necesario saber que tenemos que pensar en el problema de la normalización en cada nuevo método que aprendemos.

Estandarización			
Algoritmo	**Requiere**	**Var. Categóricas**	**Observaciones**
Reg. Lineal	No	Admite, pero no hace falta estandarizarlas	No es necesario estandarizar porque el valor de los coeficientes da cuenta de las unidades. Los resultados no cambian por estandarizar.
Reg. Ridge, Lasso, Elastic Net	Sí	Admite, hay que estandarizarlas	Estandarizamos para que el término de penalización no favorezca a ninguna variable en particular. Es el único caso en el cual estandarizamos variables dummy, para que puedan compararse con variables en todas las otras unidades.

KNN, GMM, Clustering (kmeans, DBScan y clustering jerárquico)	Sí	Hay que implementar el algoritmo con alguna medida de distancia que no sea la euclideana y que funcione para variables categóricas.	En todos los algoritmos que se basan en medidas de distancia, se debe modificar la implementación nativa de Scikit Learn.
PCA	Sí	No se puede aplicar PCA en variables categóricas.	
Redes neuronales	Si	Los inputs deben estar entre 0 y 1 por lo tanto se usa estandarización min-max. Las variables dummy ya están así codificadas y se pueden utilizar.	
Ensambles de árboles (Random Forest, XTrees, XGBoost	No	Admite. Dependiendo de la implementación es necesario o no hacer One Hot Encoding.	El valor elegido para hacer el split da cuenta de las unidades de cada columna.

¿Cómo normalizar?

- Existen algunas formas típicas de normalizar
- La ***estandarización***: $x_{norm} = (x - \mu)/\sigma$
- La normalización ***min-max:*** $x_{norm} = (x - min) / (max - min)$

- La elección entre minmax y estandarización depende del objetivo del método.

    - **min-max**: Tiene sentido en los casos donde importa que los features tengan las mismas unidades pero no necesariamente la misma varianza

    - **estandarización:** Tiene sentido donde se necesita que los features tengan las mismas unidades y también la misma varianza, como por ejemplo en componentes principales.

Demo

Normalizando con python

Práctica Guiada

Normalización L1 y L2

Conclusión

Usamos la normalización para:

- Manejo de cantidades en **diferentes unidades o escalas**
- Muchos **algoritmos** de machine learning toman la normalización como **requerimiento**
- Puede **aumentar la velocidad de convergencia** usando el método de gradiente

Existen diferentes métodos de normalización, como la estandarización, min-max y L1 y L2

Normalización