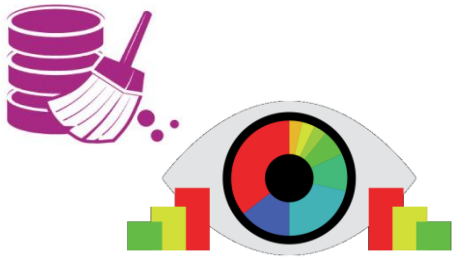
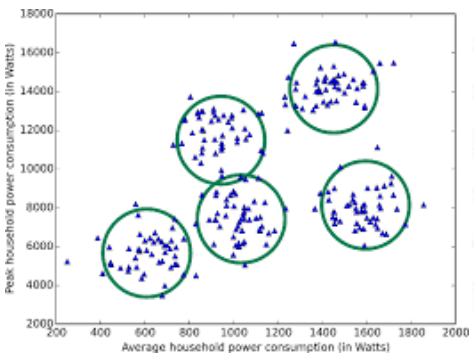
APRENDIZAJE AUTOMÁTICO

AGENDA

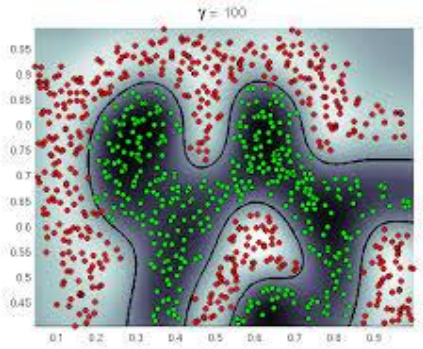


Exploratory Data Analysis (EDA)

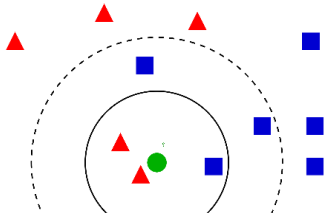
Aprendizaje automático



Aprendizaje no supervisado



Aprendizaje supervisado



KNN

EDA (EXPLORATORY DATA ANALYSIS)

Una vez uno obtiene un dataset, es necesario entenderlo para

- Estimar si puede servir para responder la pregunta de investigación
- Identificar relaciones entre las variables
- Identificar patrones y tendencias en los datos
- Identificar datos excepcionales
- Identificar problemas en la calidad de los datos y establecer como lidiar con ellos
  - Datos faltantes
  - Datos anómalos
  - Datos repetidos
  - Problemas de escala
  - Problemas de tipos de datos (enteros, numéricos, etc..)

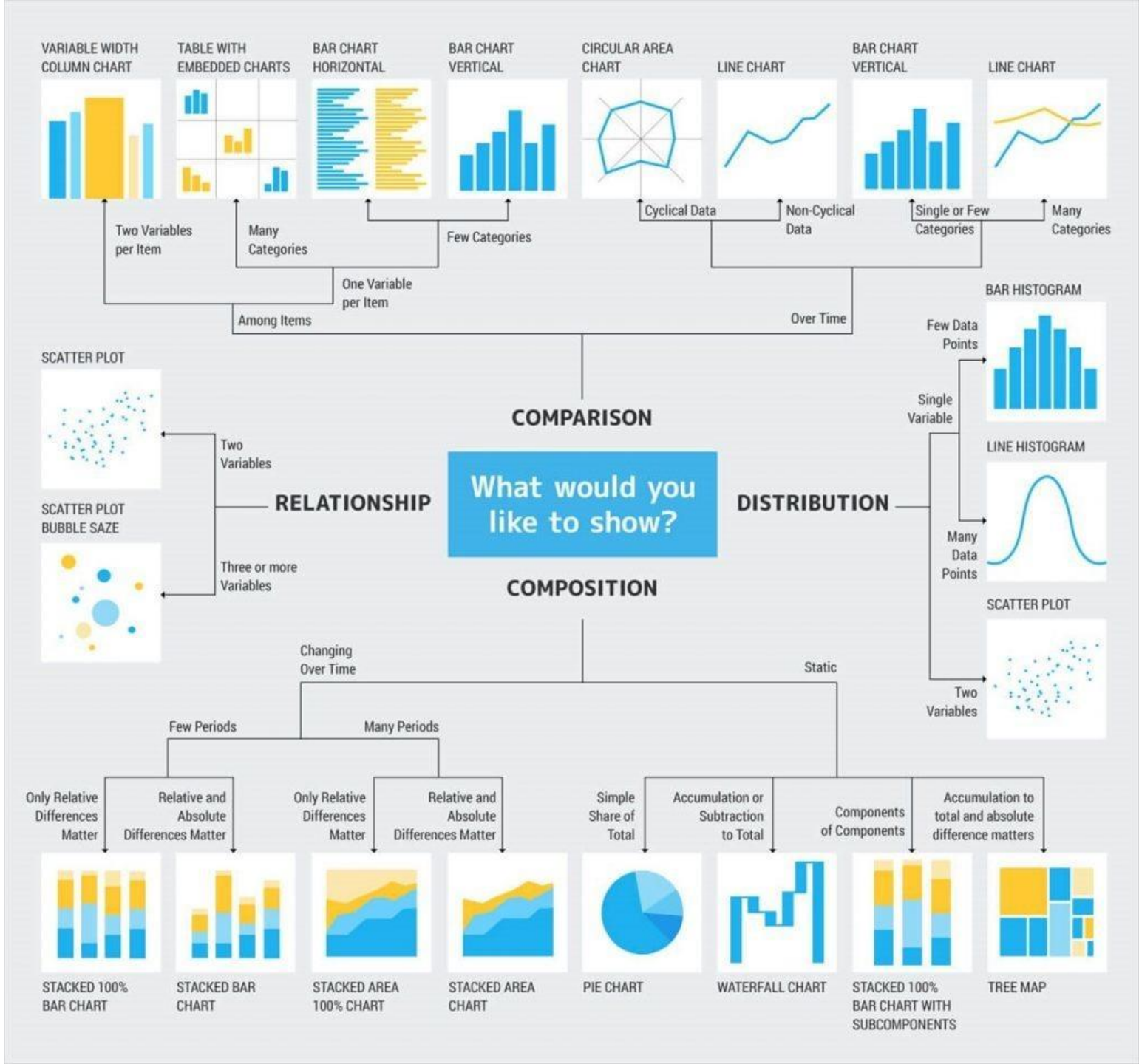
EDA (EXPLORATORY DATA ANALYSIS)

En Python tenemos los siguientes comandos que nos permiten entender mejor los datos, cuando se encuentran en un **dataframe**:

- El método **head** permite obtener los primeros registros de un dataframe.
- El objeto **dtypes** indica las clases de las columnas del dataframe
- El método **info** de un dataframe permite consultar información como el número de registros y de columnas con los tipos de datos correspondientes, el número de registros presentes (por oposición a los registros faltantes), y el tamaño que ocupa el dataframe en memoria.
- El método **describe** de un dataframe permite obtener un resumen de las columnas, con estadísticas descriptivas que permiten entender la distribución de cada variable.

EDA – VISUALIZACIÓN

Los gráficos son herramientas muy poderosas que permiten identificar y comunicar conceptos muy particulares de los datos

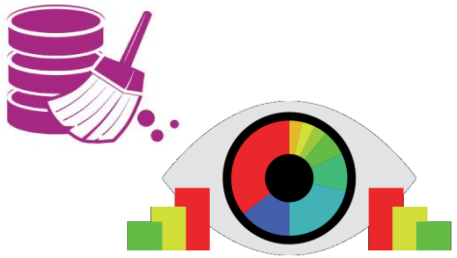


EDA (EXPLORATORY DATA ANALYSIS)

En Python podemos utilizar visualizaciones que nos permiten entender mejor los datos. Utilizamos la librería **seaborn** (que extiende una librería de base **matplotlib**)

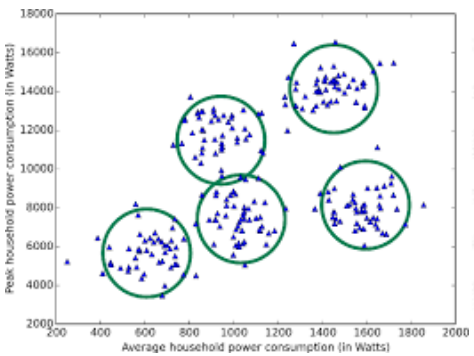
- Un gráfico de barras
- Un gráfico de lineas
- Un gráfico de densidades
- Un scatterplot
- Un boxplot

AGENDA

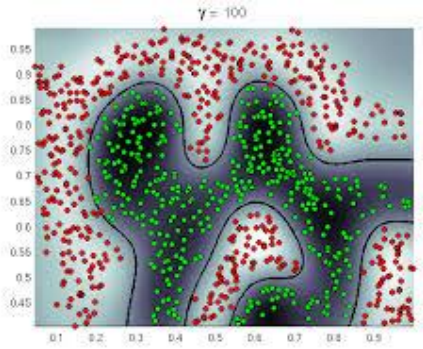


Exploratory Data Analysis (EDA)

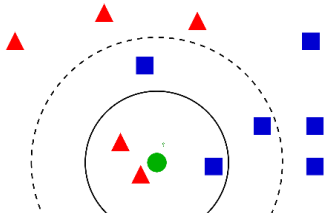
Aprendizaje automático



Aprendizaje no supervisado



Aprendizaje supervisado

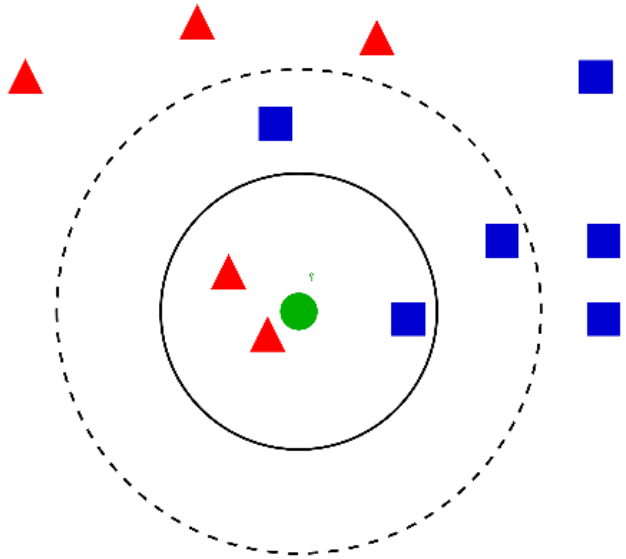


KNN

KNN

KNN (K Nearest Neighbors): K Vecinos más Cercanos

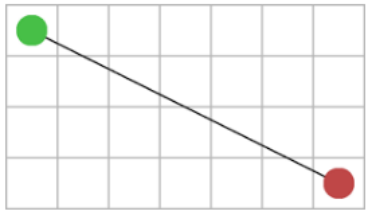
- Algoritmo de aprendizaje supervisado para **clasificación** y **regresión**
- **Sencillo**: asignar la clase o valor agregado de las instancias conocidas que se encuentran mas cerca de la instancia a predecir
- Basado en las **instancias** de aprendizaje, no en un modelo subyacente probabilístico/estadístico
- Aprendizaje **perezoso**: en realidad el algoritmo solo se ejecuta en el momento que se requiere predecir una nueva instancia a partir de una predicción local



Wikipedia, 2016

KNN – DISTANCIAS

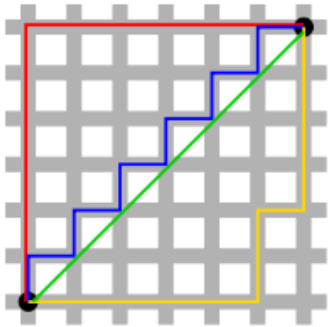
- Basado en una medida de **similitud** o **distancia** que hay que definir para encontrar los vecinos mas cercanos:
  - **Euclidiana**: tamaño del segmento linear que une las dos instancias comparadas.



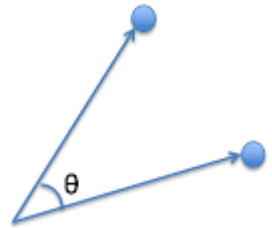
$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

  - **Manhattan**: basada en una organización en bloques rectilíneos



  - **Coseno**: coseno del ángulo entre las dos instancias comparadas → Alta dimensionalidad y **big data**



$$sim(\boldsymbol{x},\boldsymbol{y}) = \cos(\theta_{x,y}) = \frac{\boldsymbol{x}.\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} = \frac{\sum_i x_i * y_i}{\sqrt{\left(\sum_i x_i * x_i\right) * \sum_i y_i * y_i}}$$

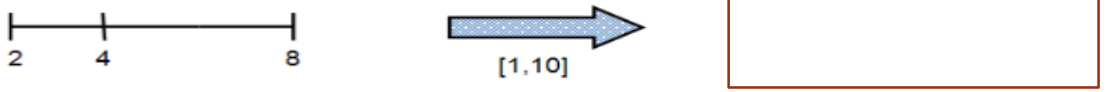
KNN - NORMALIZACIÓN

- Normalización [0, 1]

$$Y = \frac{X - mínimo\_original}{máximo\_original - mínimo\_original}$$

- Normalización [newmin, newmax] → Generalización, cambio de escala a otro intervalo cualquiera, no necesariamente [0,1], ni [oldmin, oldmax]

$$Y = \min + \frac{X - mínimo\_original}{máximo\_original - mínimo\_original}(\max - \min)$$

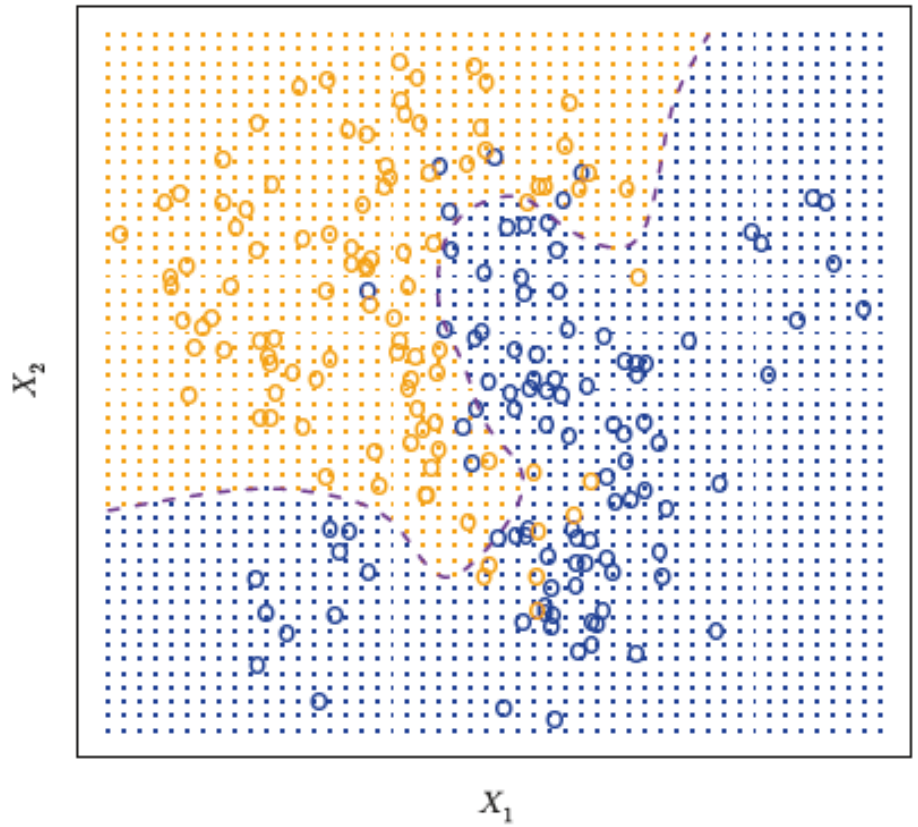


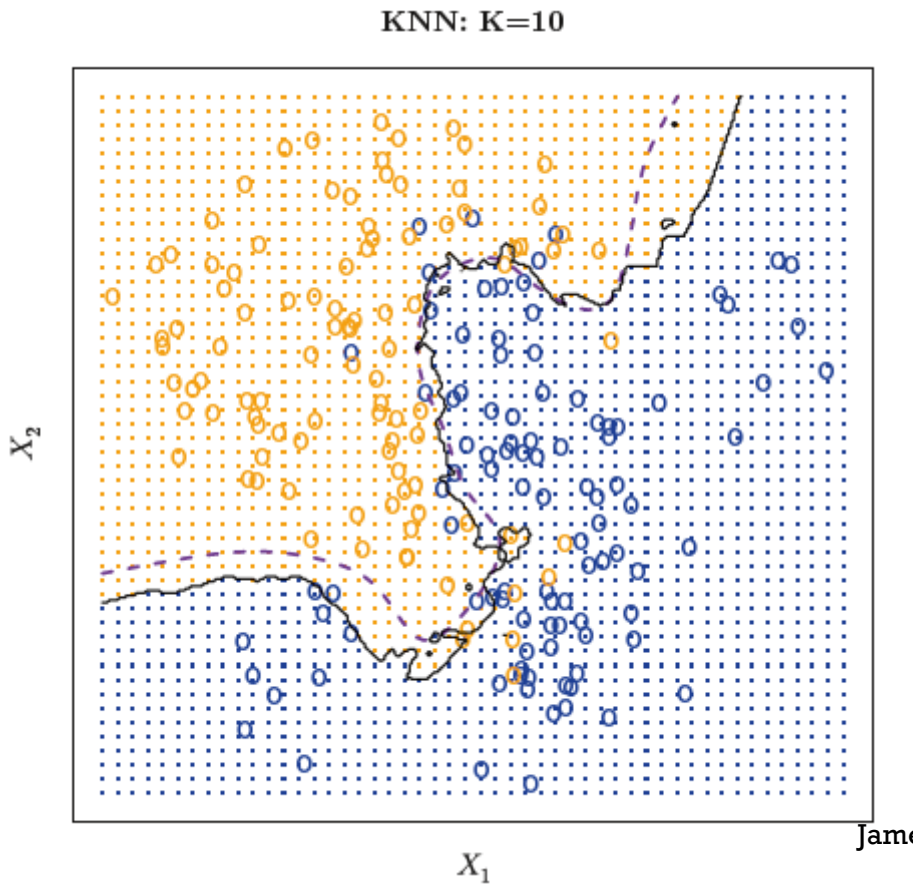
- Normalización z-score (estandarización)
  - Supuesto de distribución normal
  - Sea Z la representación estandarizada del dato
  - X la representación actual del dato
  - μ el valor promedio de los datos
  - σ la desviación estándar del campo

$$Z = \frac{X - \mu}{\sigma}$$

KNN – K

- **Parámetro K**: número de vecinos mas cercanos a considerar para establecer la clase o valor de una nueva instancia

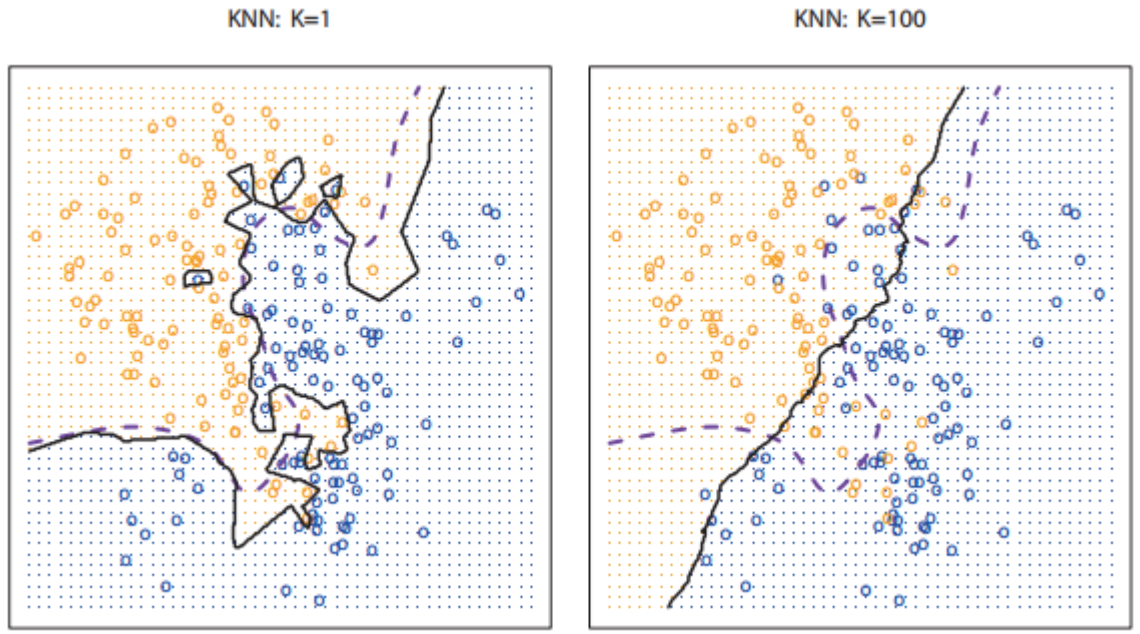




James et al, ISLR, 2013

KNN – K

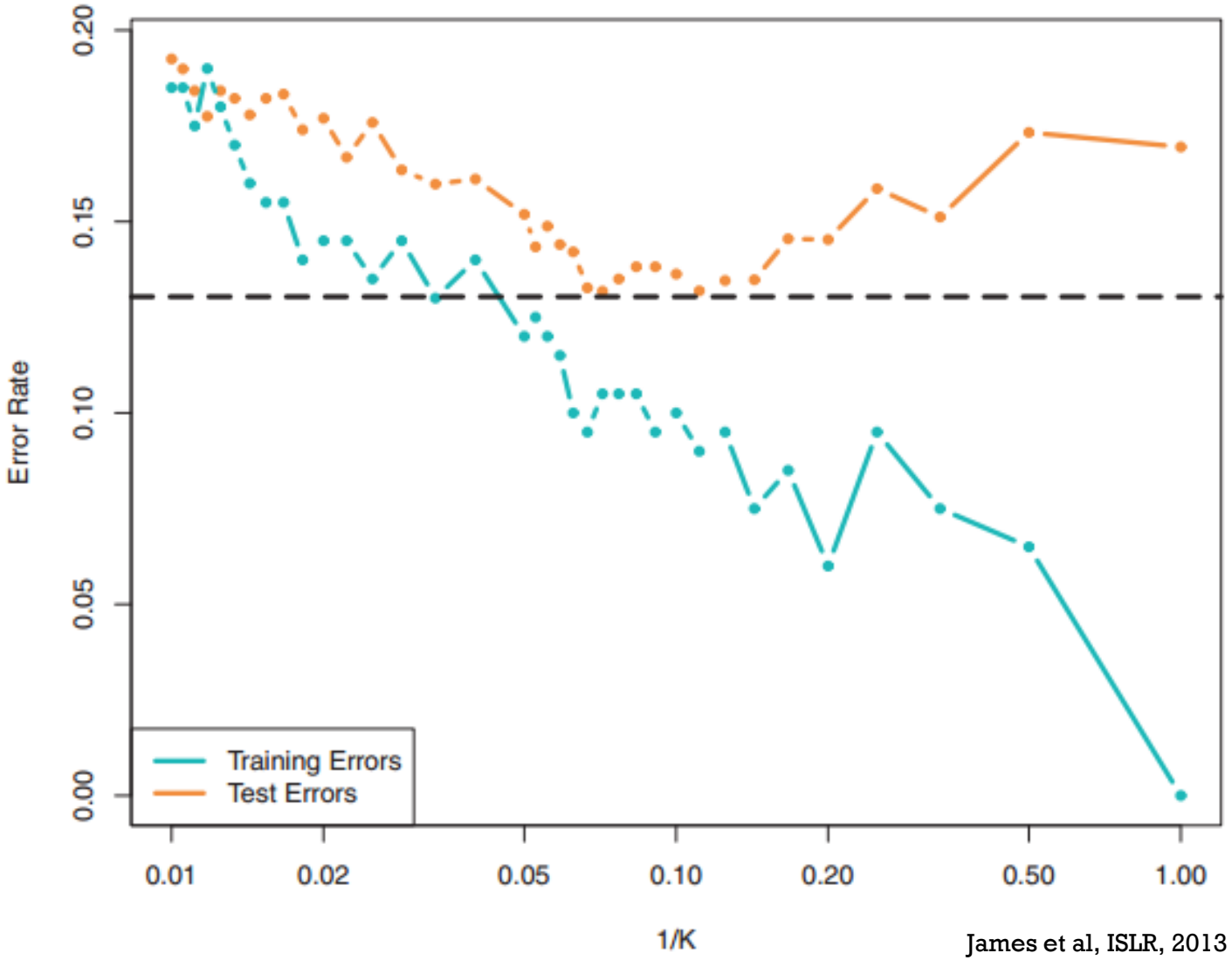
- **Parámetro K**
  - El resultado puede ser drásticamente diferente para diferentes valores de K
  - Un valor de K grande suavizará los límites entre clases/valores (alto sesgo, baja varianza)
  - Un valor de K pequeño resultará en límites muy flexibles (bajo sesgo, alta varianza)
  - El valor de K óptimo se encuentra empíricamente



James et al, ISLR, 2013

KNN – K

- **Overfitting**: (sobre aprendizaje) a considerar en el momento de escoger el K.
- Modelos mas sencillos previenen el overfitting → K mas grandes
- Igualmente, cuidado con el **underfitting** (sub aprendizaje)

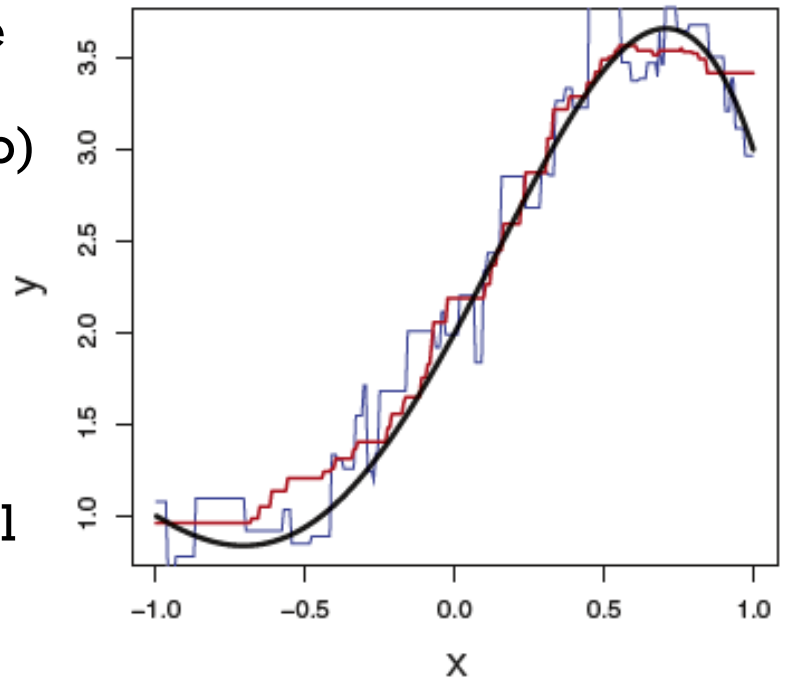


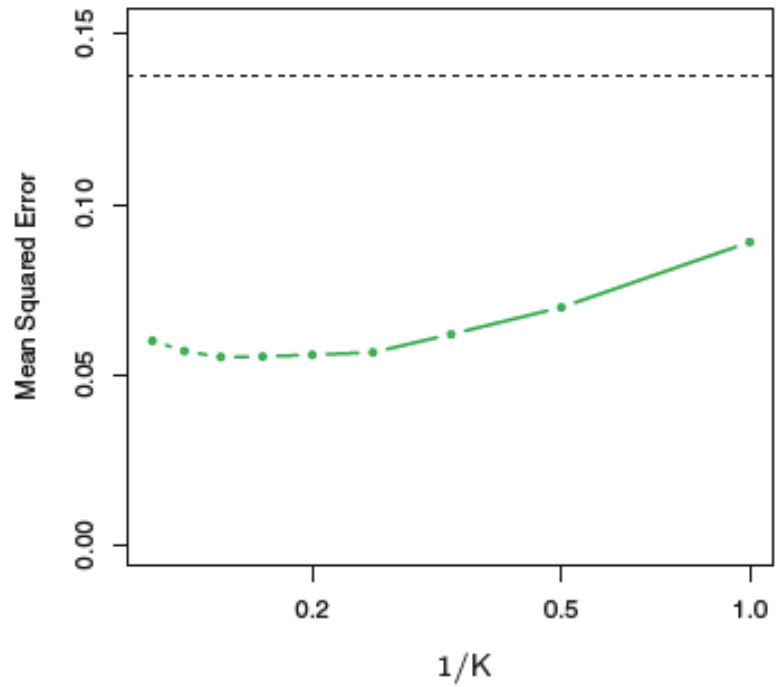
James et al, ISLR, 2013

KNN – K

En el caso de la utilización de KNN para la regresión las mismas consideraciones aplican

- En el panel izquierdo: se aplica KNN con un valor de K=1 (azul) y K=9 (rojo)
- En el panel derecho, se puede ver el valor de RMSE para diferentes valores de K (en verde). También se puede ver, por comparación el nivel de error de la regresión lineal simple (punteada en negro)





James et al, ISLR, 2013

KNN

Consideraciones:

- Perezoso (Lazy learning)
- No paramétrica y no lineal
- **Método local, no generalizable (no hay un modelo construido como tal)**:
  - Puede encontrar particularidades muy específicas a ciertas regiones
  - Su uso (sobre todo en regresión) sólo permite estimaciones en los rangos de las variables del set de aprendizaje (extrapolación no tiene mucho sentido)
- Maldición de la **dimensionalidad**
- Muy sensible a la **unidad de medida** de los atributos (se deben **normalizar** las variables para evitar diferencias en sus importancias finales), y a atributos que no aportan poder predictivo (e.g. el color de los ojos no debería considerarse para predecir la edad de una persona)
- No sabe que hacer con los **missing values**, ni con variables **categóricas**
- **Variaciones**: K-nn ponderado por la distancia, basado en un radio dado.

CNN (CONDENSED NEAREST NEIGHBORS)

- Dificultad de aplicación de KNN cuando se tienen **muchos registros**
- No todos los registros son necesarios para la correcta clasificación
- Aproximación de KNN utilizando un conjunto de datos reducido
- Escogencia de **prototipos** que permitan una clasificación con K=1 lo más parecida al resultado utilizando el dataset completo
- Algoritmo: Siendo **X** el conjunto de datos inicial y **U** el conjunto reducido:
  - Identificar todos los elementos x de **X** cuyo vecino más cercano sea de clase diferente
  - Retirar los x identificados (son prototipos) de **X** y agregarlos a **U**
  - Repetir hasta que no se agreguen más prototipos a **U**

TALLER DE CLASIFICACIÓN CON KNN

- DATASET: 150 ejemplos pertenecientes a 3 especies diferentes de la flor Iris
- 4 Atributos: largo y ancho del sépalo, largo y ancho del pétalo
- Reproducir el taller

Iris setosa

Iris versicolor

Iris virgínica