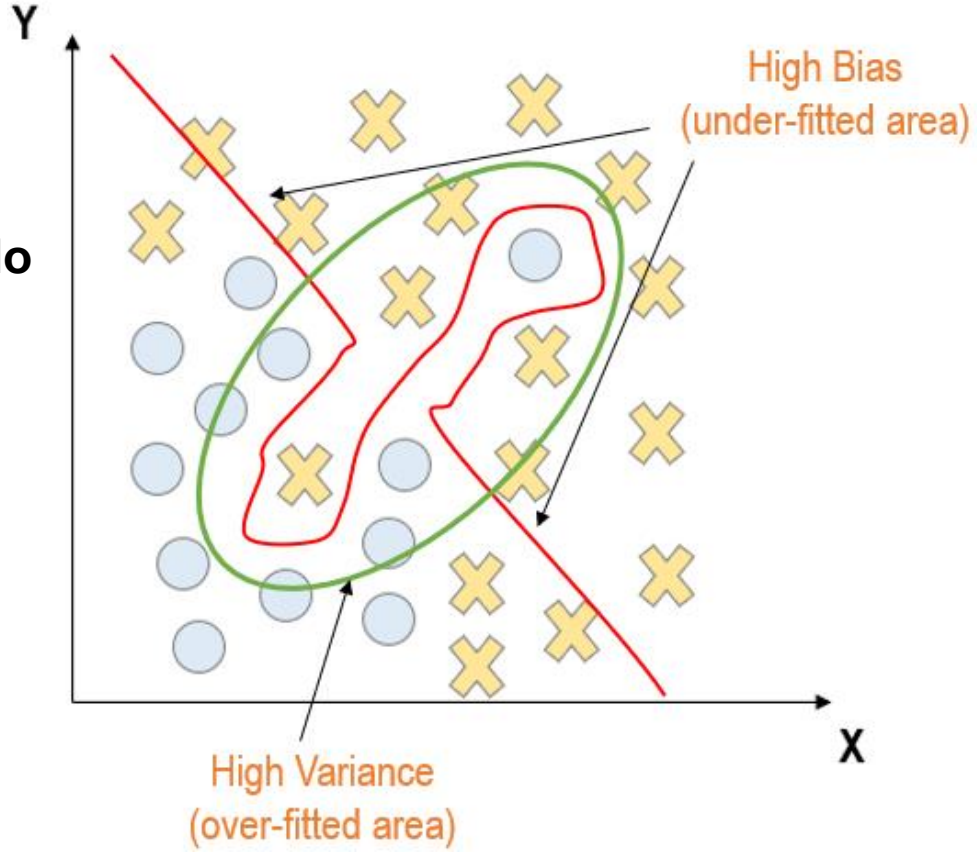
Introducción a la Ciencia de Datos

BIAS - VARIANCE

- Cuando admitimos errores de **sesgo** provenientes de la data (por ejemplo target desbalanceado) y **no realizamos alguna modificación al respecto** (modelo simple o underfitting) **es cuando obtenemos un alto bias.**

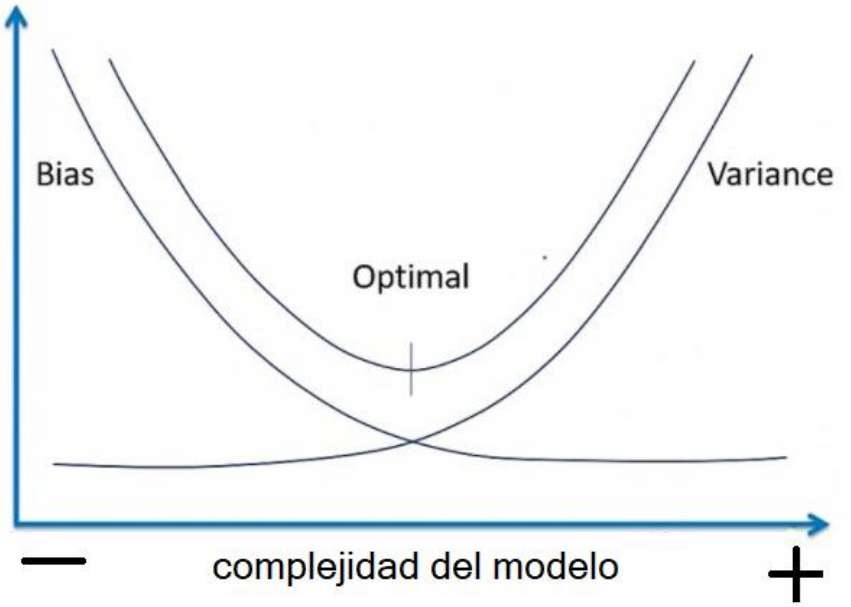
- Cuando no admitimos **el más mínimo error** al predecir el target ejecutando **la mayor cantidad de modificaciones** (modelo complejo u overfitting) **es cuando obtenemos un alto variance** (respecto a data real distinta al train)



BIAS - VARIANCE

- Por eso es importante siempre **realizar un preprocesamiento de los datos,** de esa manera disminuyes el sesgo que puede haber por data desbalanceada u otras características del train.

- También es recomendable **realizar una validación cruzada** para entrenar al modelo con diferentes tipos de datos y esté preparado para **data real** que puede **comportarse distinta al train**

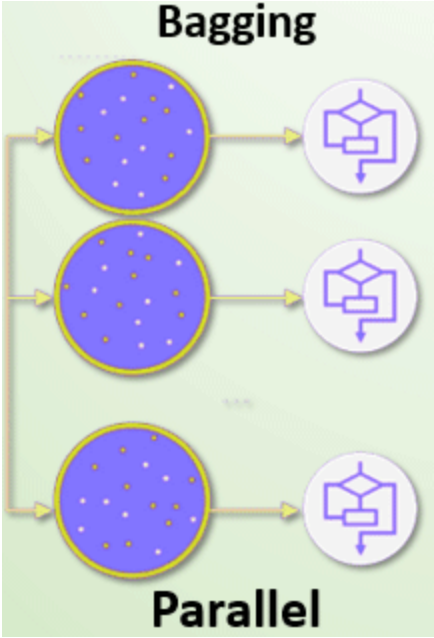


Aprendizaje Supervisado

Modelos de Boosting

BAGGING

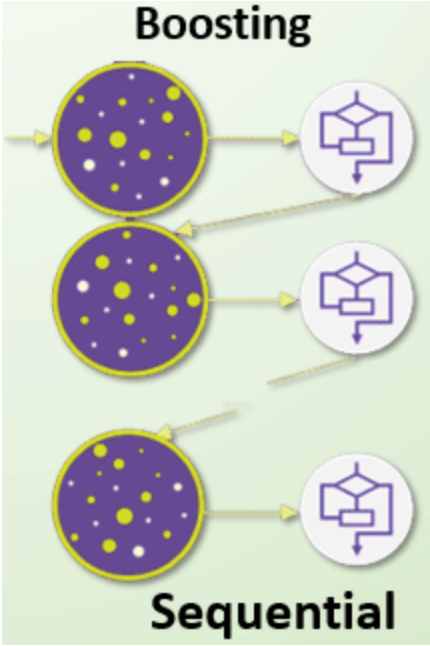
- Entrena con **diferente datos y variables** de forma **paralela e independiente**
- Si el objetivo es **clasificación** la predicción final será la **categoría más frecuente**
- Si el objetivo es **regresión** la predicción final será **el promedio de los target**



BOOSTING

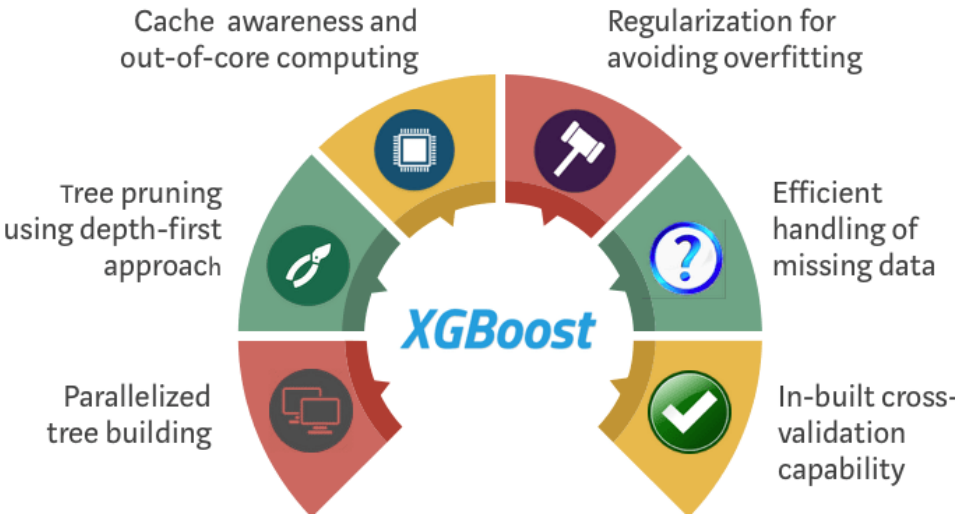
- Entrena con **diferente datos y variables** de forma **secuencial.** De esta manera **cada iteración aprende de la anterior.**

- Esta mejora dio inicio a la aplicación de modelos de **boosting.** El primero modelo se denominó **AdaBoost.**



XG BOOST

- Admite la presencia de **valores nulos y outliers**
- Tiene una **validación cruzada interna**
- Óptimo para **gran volumen** da datos
- Tiene un **mayor costo** computacional por ser **más complejo**



XG BOOST

learning rate: 0.1

Se indica para limitar el ratio de aprendizaje (decimal menor a 1) según el criterio del **descenso de la gradiente**

max depth: 3

Es la profundidad máxima de cada árbol

min child weight: 1

Previene el overfitting si aumenta el valor ya que requiere mayor cantidad de elementos a clasificar en cada nodo

objective: reg:squarederror

Especifica el objetivo (clasificación o regresión) y el tipo de target

num class: Especifica el número de categorías en target **(solo para clasificación)**

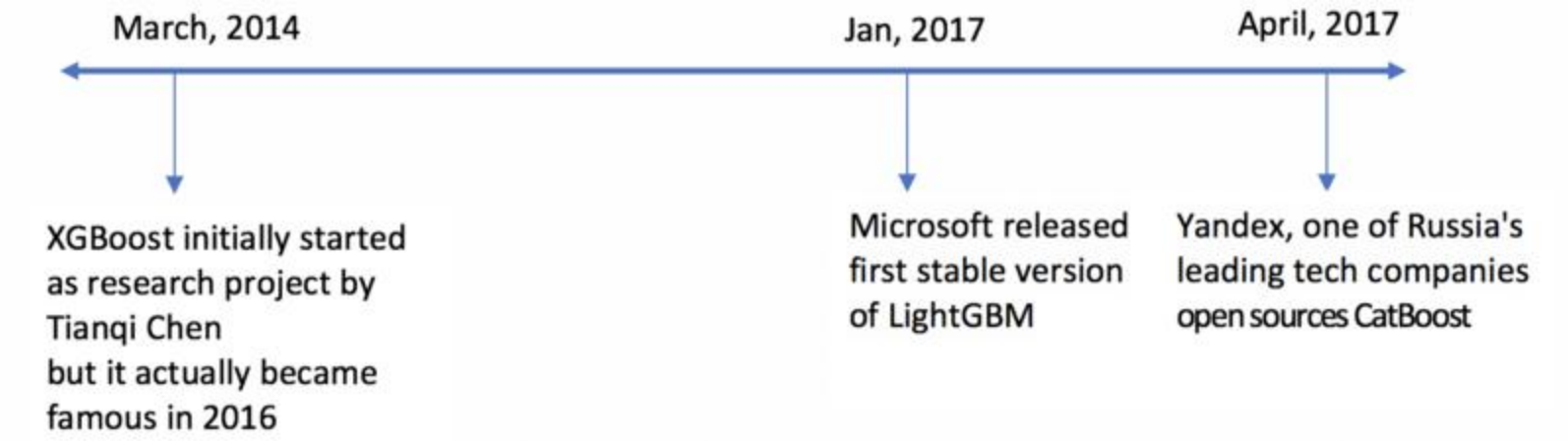
eval metric: de acuerdo al target (f1-score, auc, etc.)

eval set: [Xtest, y test] (referencia a comparar en cada iteración)

*n estimators y early stopping **rounds**:* número entero que limita las iteraciones

- Mayor información : https://xgboost.readthedocs.io/en/latest/parameter.html

LIGHT GBM



- **Mejor performance** respecto al **XGBoost**
- Admite **variables categóricas** sin necesidad de **encoding**

LIGHT GBM

learning rate: 0.1

Se indica para limitar el ratio de aprendizaje (decimal menor a 1) según el criterio del **descenso de la gradiente**

max depth: 3

Es la profundidad máxima de cada árbol

min child weight: 1

Previene el overfitting si aumenta el valor ya que requiere mayor cantidad de elementos a clasificar en cada nodo

objective: reg:squarederror

Especifica el objetivo (clasificación o regresión) y el tipo de target

num class: Especifica el número de categorías en target **(solo para clasificación)**

***metric**: de acuerdo al target* (f1-score, auc, etc.)

eval set: X train, y train (referencia a comparar en cada iteración)

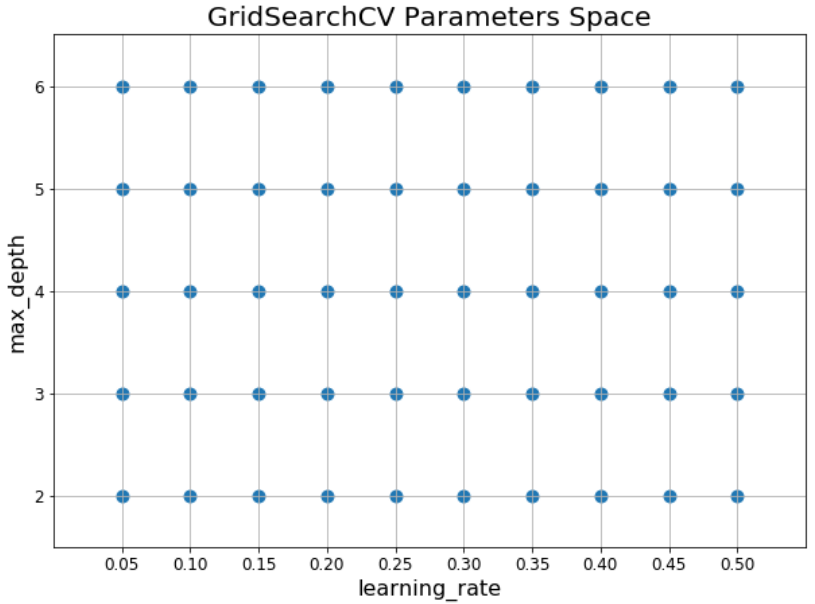
n estimators: número entero que limita las iteraciones

***categorical feature*:** permite indicar una lista de **variables categóricas** en train

- Mayor información: https://lightgbm.readthedocs.io/en/latest/Parameters.html

GRID SEARCH CV

- Esta librería está enfocada en **la mejora de los parámetros propios del modelo**
- Recuerda **previamente especificar el rango de cada parámetro.**
- También es recomendable aplicar un **early stopping** para optimizar el tiempo y costo computacional



https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html