INTRODUCCIÓN AL DL

Javier Diaz, PhD

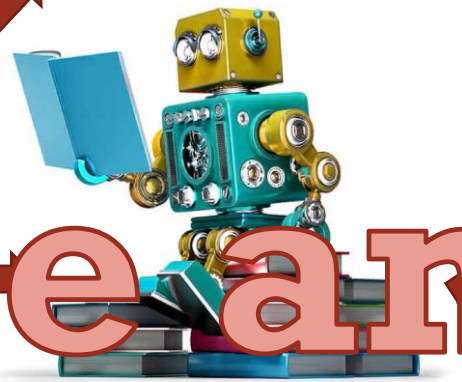
UNIVERSIDAD ICESI

AGENDA

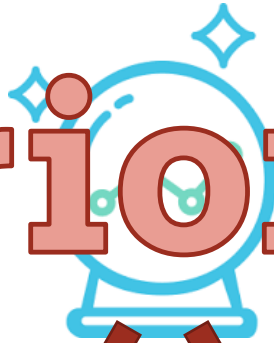
Aplicaciones

Aprendizaje no supervisado

Deep Learning

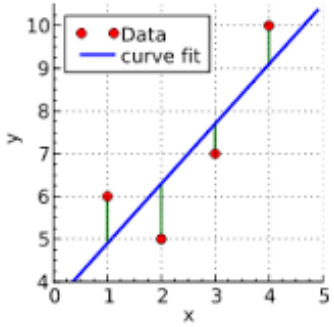


Machine Learning (Aprendizaje Automático)

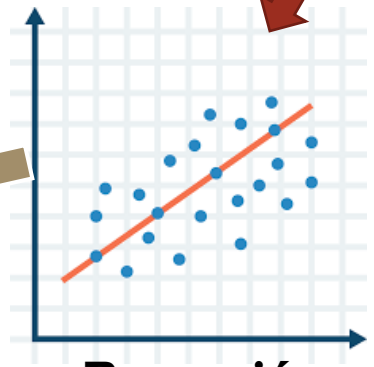


Aprendizaje supervisado

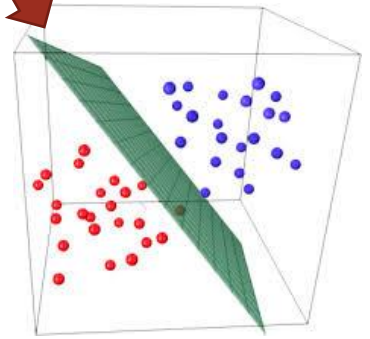
Clase anterior



Mínimos cuadrados ordinarios



Regresión

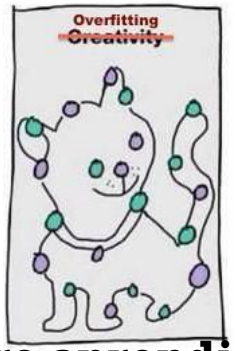


Clasificación

UNIVERSIDAD ICESI

AGENDA

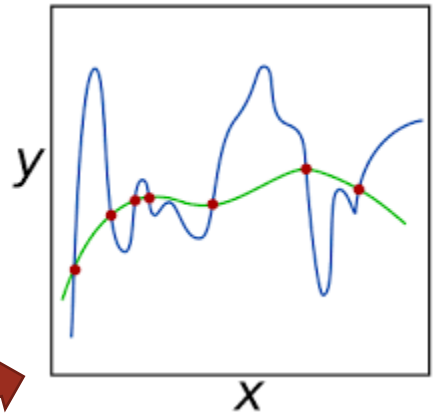
Deep Learning



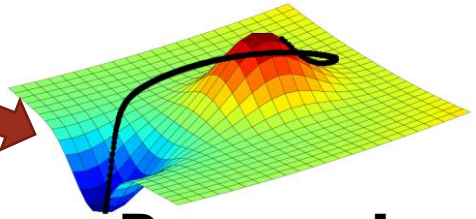
Sobre aprendizaje (Overfitting)

Aprendizaje supervisado

Protocolos de evaluación

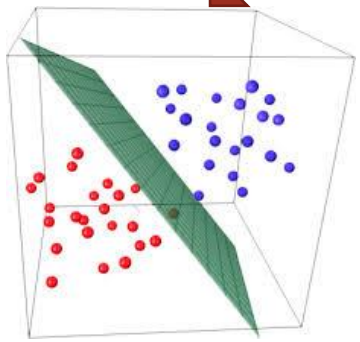


Regularización

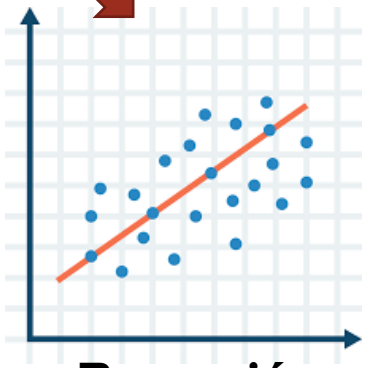


Descenso de gradiente

Métricas de Evaluación de la clasificación

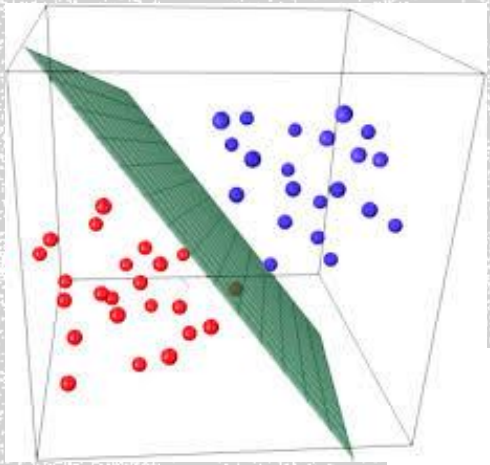


Clasificación



Regresión

CLASIFICACIÓN: EVALUACIÓN



MÉTRICAS DE EVALUACIÓN

- Necesidad de evaluar la calidad de los modelos de aprendizaje automático
- Diferentes criterios a tener en cuenta:
  - Correctitud de la predicción
  - Simplicidad (parsimonia)
  - Interpretabilidad
  - Tiempo de aprendizaje o de predicción
  - Escalabilidad (importante para Big Data)

MÉTRICAS DE CLASIFICACIÓN

- Se usa una **matriz de confusión** para evaluar diferentes métricas de correctitud/error
- Se utilizan dos calificadores para describir cada una de sus casillas:
  - Un calificador de la correctitud de la predicción con respecto a la realidad: Verdadero o Falso
  - Un calificador del tipo de la predicción: Positivo o Falso, con respecto a cada clase de interés (i.e churn)
- Dependiendo del contexto los tipos de error pueden ser mas graves que otros (costos diferentes)

		Predicción	
		Churn P	No churn N
Realidad	Churn +	VP	FN - Tipo II
	No churn -	FP - Tipo I	VN

- La diagonal (en verde) muestra las instancias correctamente clasificadas. Las demás casillas resume diferentes tipos de error:
  - Tipo I: Falsos positivos
  - Tipo II: Falsos negativos

¿Qué pasa cuando hay mas de dos clases?

MÉTRICAS DE CLASIFICACIÓN

- **Interpretarían el caso de la detección de un email spam**

  TP, TN:

  FP: , consecuencia:

  FN: , consecuencia:

- **Interpretar el caso del diagnóstico de una enfermedad grave?**

  TP, TN:

  FP: , consecuencia:

  FN: , consecuencia:

		Predicción	
		Churn P	No churn N
Realidad	Churn +	VP	FN - Tipo II
	No churn -	FP - Tipo I	VN

- **Interpretar el caso de la prospección de clientes de un crédito de consumo (baja aceptación)**

  TP, TN:

  FP: , consecuencia:

  FN: , consecuencia:

MÉTRICAS DE CLASIFICACIÓN

- Tasa de correctitud (accuracy) = (VP+VN)/(VP+VN+FP+FN)
- Error de mala clasificación (contrario de *accuracy*) = (FP+FN)/(VP+VN+FP+FN): probabilidad de error
- Precisión= VP / (VP+FP): valor de predicción positiva, P(Real+|Predicho+)
- *Recall* (o TPR o sensibilidad)= VP / (VP+FN): qué proporción de todos los positivos reales pude identificar como tal, P(Predicho+|Real+)
- Especificidad (o TNR): = VN / (VN+FP): qué proporción de todos los negativos reales pude identificar como tal, P(Predicho-|Real-)
- Valor de predicción negativa (FPR) = VN / (VN+FN)
- F1-Measure = $2 * \frac{precision * recall}{precision + recall}$ (promedio armónico)

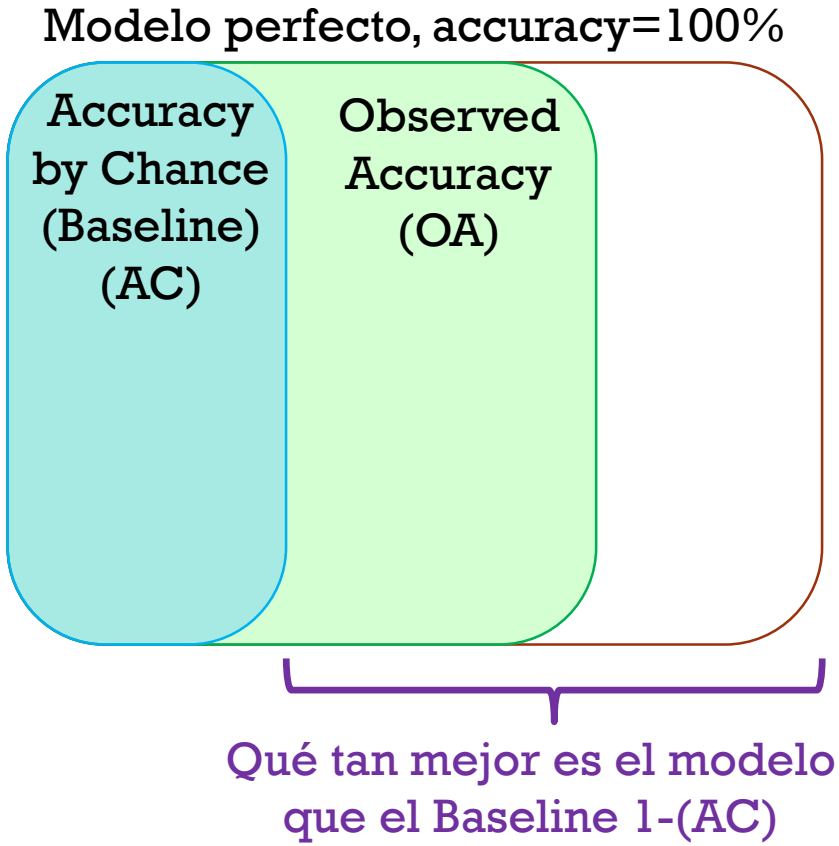
		Predicción	
		Churn P	No churn N
Realidad	Churn +	VP	FN - Tipo II
	No churn -	FP - Tipo I	VN

Imaginemos el problema de detección de spam mail e interpretemos cada métrica

Imaginemos el problema de diagnóstico de cáncer e interpretemos cada métrica

MÉTRICAS DE CLASIFICACIÓN

- Coeficiente de concordancia **Kappa**
  - Para datos nominales u ordinales
  - Concordancia entre las predicciones y las clases reales
  - Sustrae el efecto de concordancia por suerte (AC) del valor del **accuracy** (concordancia observada - OA)
  - Valores van de 0 a 1
  - Muy útil sobretodo cuando las clases no están balanceadas
    - Diagnóstico de enfermedades raras
    - Clientes que acepten productos de crédito)
  - Kappa = $\frac{OA-AC}{1-AC}$



MÉTRICAS DE CLASIFICACIÓN

- Coeficiente de concordancia **Kappa**
  - Para datos nominales u ordinales
  - Concordancia entre las predicciones y las clases reales
  - Sustrae el efecto de concordancia por suerte (AC) del valor del **accuracy** (concordancia observada - OA)
  - Valores van de 0 a 1
  - Muy útil sobretodo cuando las clases no están balanceadas
    - Diagnóstico de enfermedades raras
    - Clientes que acepten productos de crédito)
  - Kappa = $\frac{OA-AC}{1-AC}$

		Predicciones +	Predicciones -	TOTAL
reales	+	10	4	14
	-	3	2	5
	TOTAL	13	6	19

OA = 0,63
AC = 0,59
Kappa = 0,11

Accuracy (OA) = (10+2)/19=0,63
(AC) = (13/19 * 14/19) + (6/19 * 5/19) = 0,59
Kappa = (OA-AC)/(1-AC) =0,11

		Predicciones +	Predicciones -	TOTAL
reales	+	0	3	3
	-	0	97	97
	TOTAL	0	100	100

OA = 0,97
AC = 0,97
Kappa = 0,00

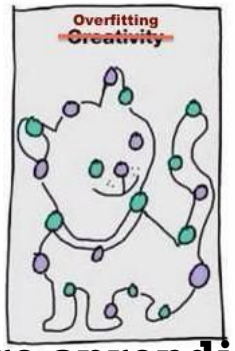
Accuracy (OA) = (0+97)/100=0,97
(AC) = (0/100 * 3/100) + (100/100 * 97/100) = 0,97
Kappa = (OA-AC)/(1-AC) =0

		Predicciones +	Predicciones -	TOTAL
reales	+	1475	988	2463
	-	556	1981	2537
	TOTAL	2031	2969	5000

OA = 0,69
AC = 0,50
Kappa = 0,38

AGENDA

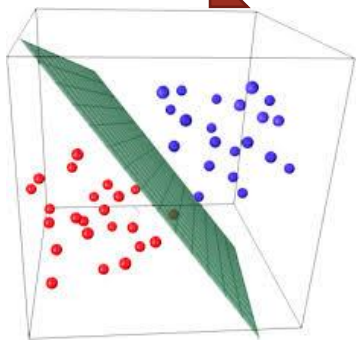
Deep Learning



Sobre aprendizaje (Overfitting)

Aprendizaje supervisado

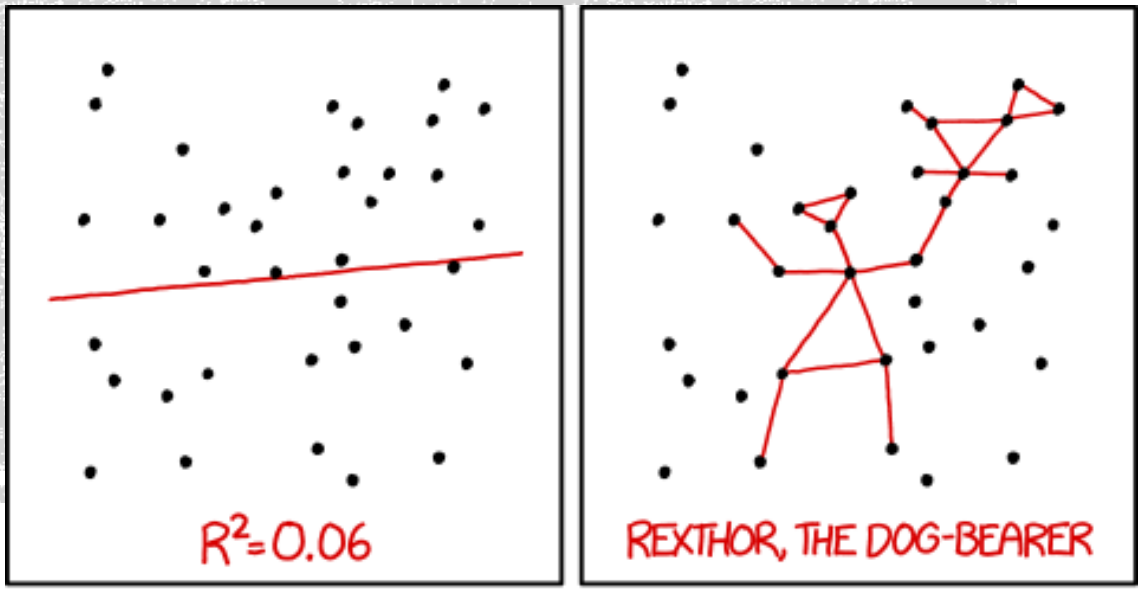
Protocolos de evaluación



Clasificación

Métricas de Evaluación de la clasificación

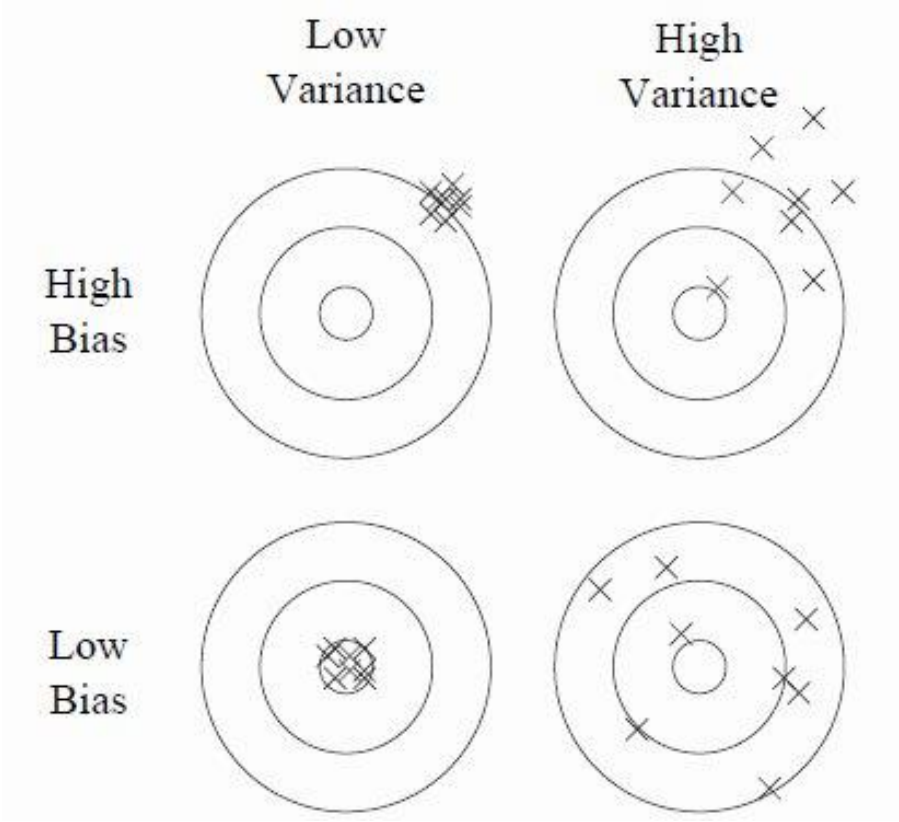
OVERFITTING, PROTOCOLOS DE EVALUACIÓN



I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

SESGO/ VARIANZA

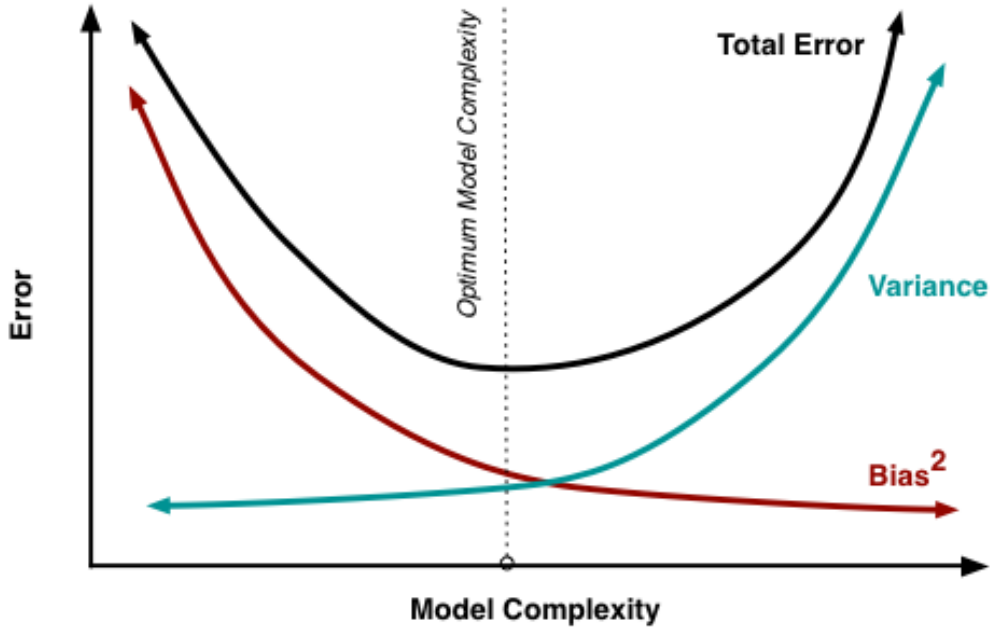
- **Sesgo** (bias): que tan lejos está el modelo de la verdad
- **Varianza**: Qué tanto varían los datos de la predicción para una misma instancia



Domingo, 2012

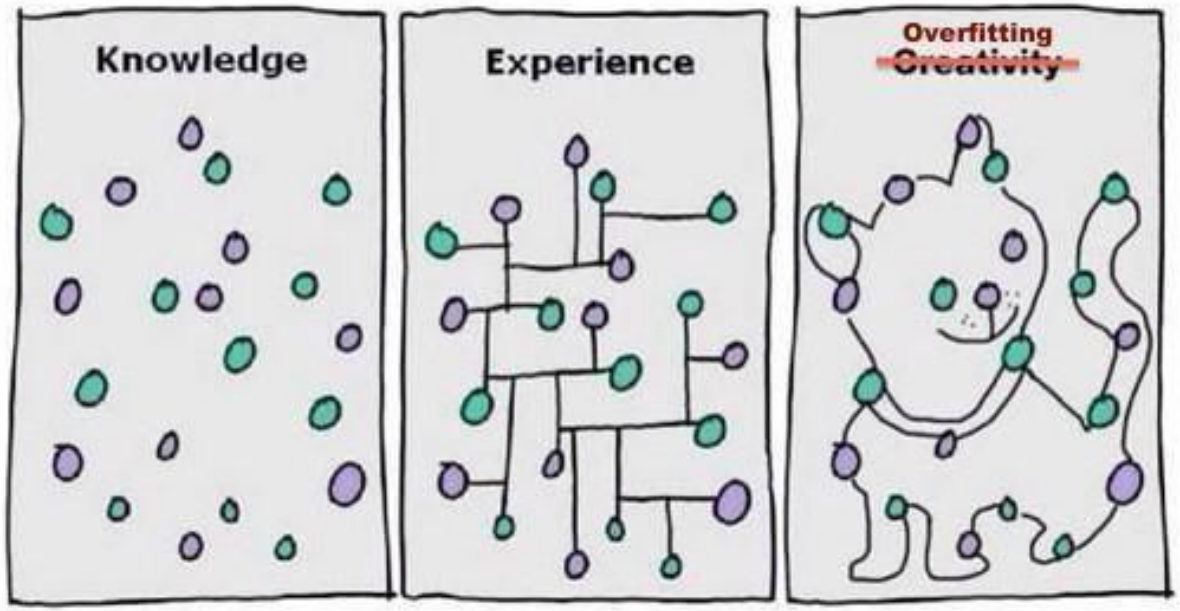
SESGO/ VARIANZA

- Ambos son fuente de error
- Se debe determinar un **compromiso** entre ambos tipos de error
- Parámetros de los modelos controlan la complejidad



http://scott.fortmann-roe.com/docs/BiasVariance.html

SOBRE APRENDIZAJE (OVERFITTING)

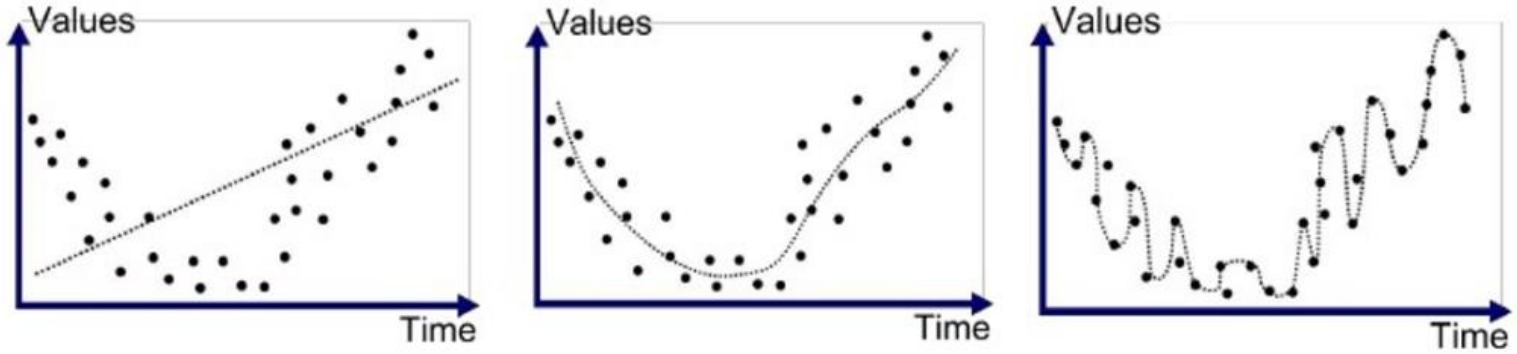


http://blog.algotrading101.com/design-theories/what-is-curve-fitting-overfitting-in-trading/

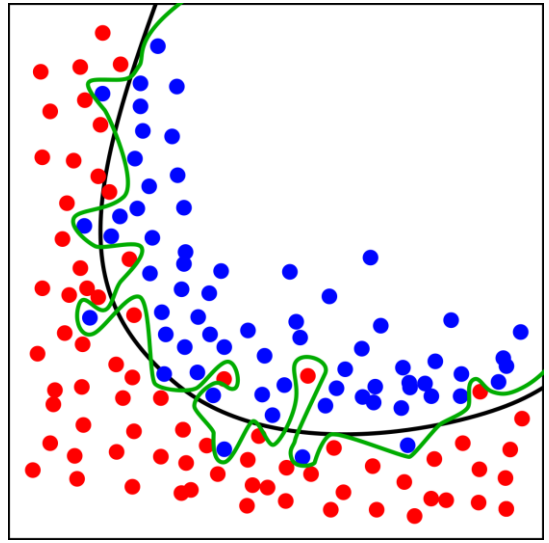
- **Sobre aprendizaje**: Los modelos aprenden a describir los errores aleatorios o el "ruido" del conjunto de entrenamiento.
- Ocurre cuando un modelo se vuelve excesivamente **complejo**

SOBRE APRENDIZAJE (OVERFITTING)

Regresión



Clasificación



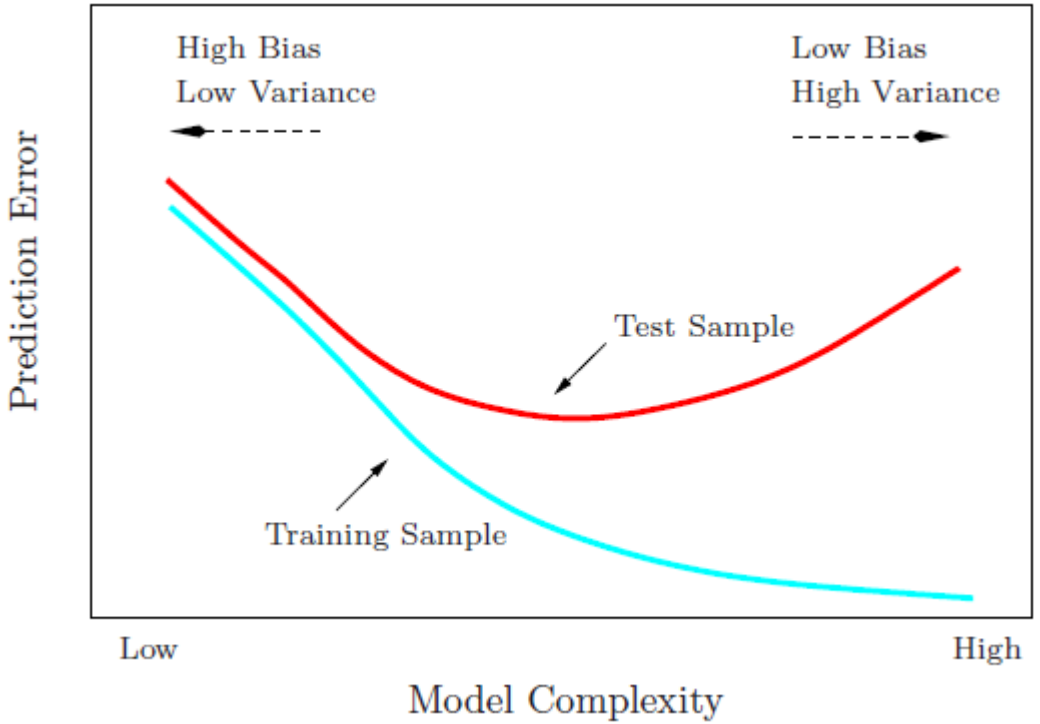
¿Cómo es el sesgo y la varianza de estos modelos?

- La **complejidad** de un modelo debe ajustarse de tal manera que permita la **generalización**, al utilizarse con datos que no haya conocido durante el proceso de entrenamiento

https://en.wikipedia.org/wiki/Overfitting

SOBRE APRENDIZAJE (OVERFITTING)

- Los modelos tienden a ajustarse al conjunto de datos usado para su aprendizaje → el **error de entrenamiento** es un mal estimador
- Queremos encontrar la complejidad del modelo que nos permita minimizar el **error de test**



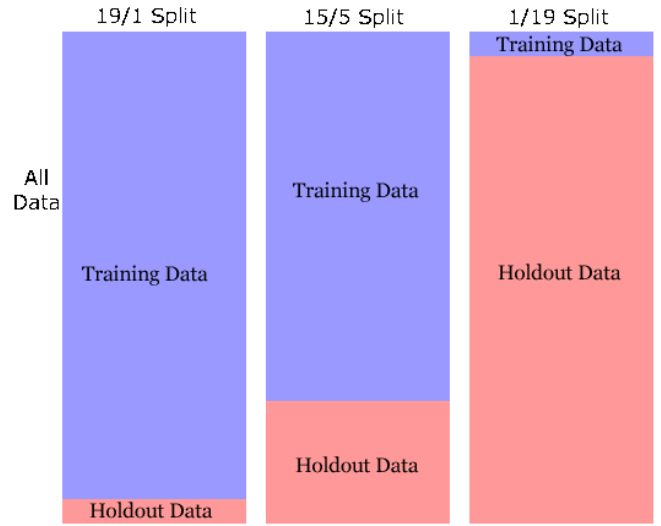
https://onlinecourses.science.psu.edu/stat857/node/160

PROTOCOLOS DE EVALUACIÓN

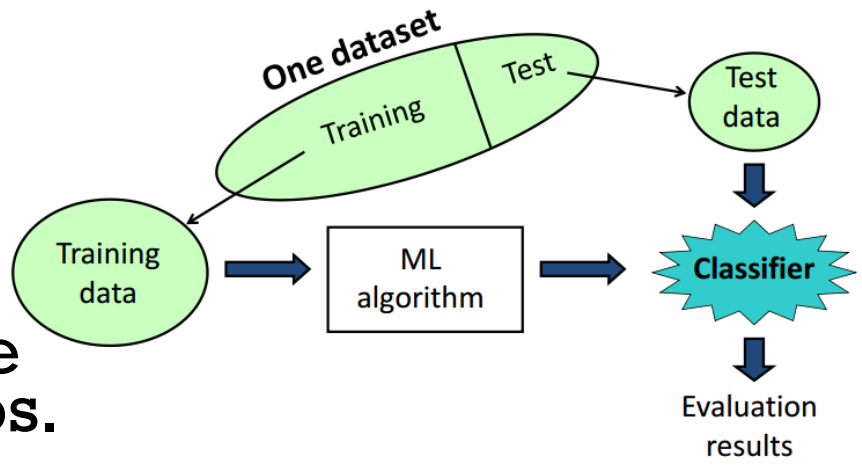
- Aplican para aprendizaje supervisado en general (tanto para clasificación como para regresión.
- Evaluar cual sería la capacidad de **generalización** del modelo a datos nuevos
- Diferenciar entre el **error de entrenamiento** y el **error de test.** Evitar el sesgo causado por la **subestimación del error** al evaluar con el mismo set de entrenamiento.
- Permitir establecer un compromiso entre sesgo y varianza, luchando contra el **sobre aprendizaje**, en busca de un modelo con buenas **capacidades predictivas**

PROTOCOLOS DE EVALUACIÓN

- **Holdout**: particionar el conjunto de datos en 2:
  - **Conjunto de entrenamiento**: con el que se aprende el algoritmo de clasificación
  - **Conjunto de validación o test**: separa al comienzo del procedimiento y no se considera en el aprendizaje
  - **Aleatoriedad** del particionamiento
  - **Compromiso**: entre mas datos mejor el aprendizaje, entre mas datos mejor la evaluación
- **Repeated holdout**: repetir el procedimiento y agregar las métricas de evaluación
- **Set de validación**: nunca se usa durante el afinamiento de los parámetros. Se evalúa sobre él al final para comparar los modelos obtenidos.



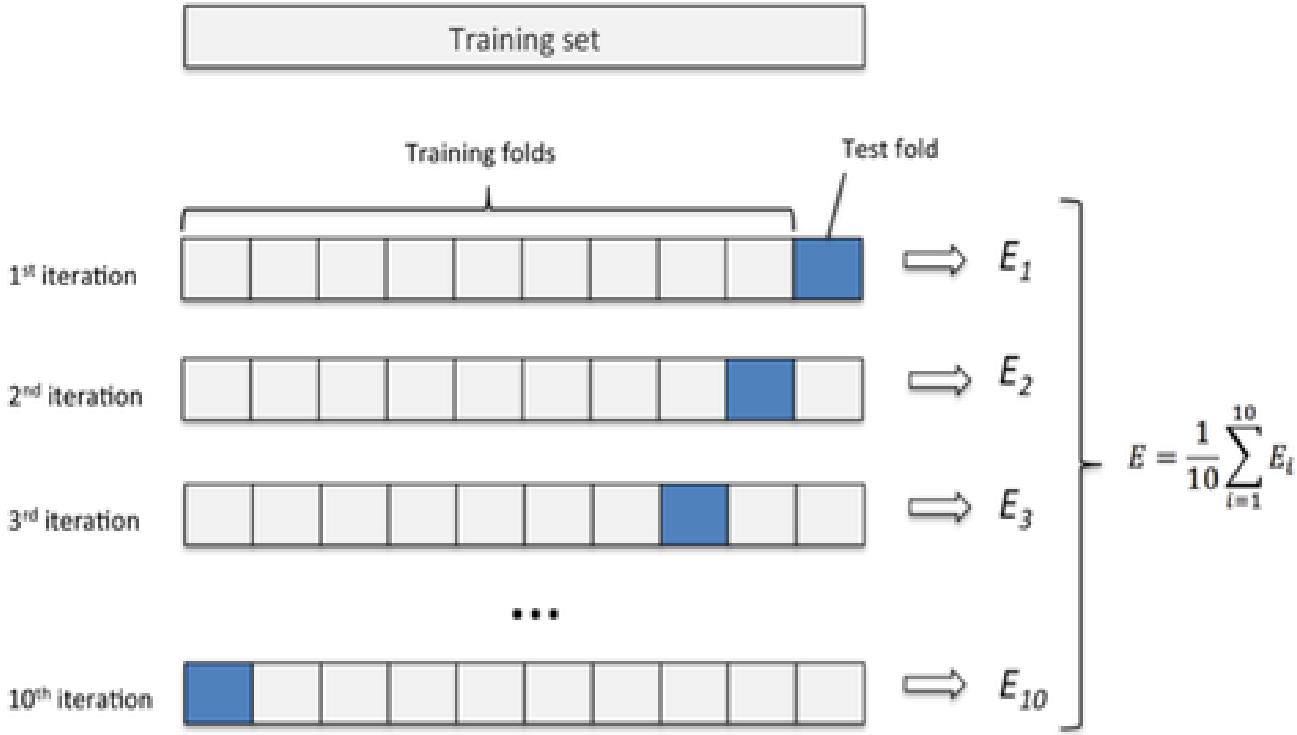
https://webdocs.cs.ualberta.ca/~aixplore/learning/DecisionTrees/InterArticle/6-DecisionTree.html



Ian Witten, Weka MOOC

PROTOCOLOS DE EVALUACIÓN

- **K-fold cross-validation**:
  - Particionar el set de datos en K conjuntos disyuntos del mismo tamaño
  - K-1 partes se usan para entrenamiento, 1 parte se usa para el test
  - Se repite el proceso K veces
  - Se agregan las métricas de evaluación



Sebastian Raschka, 2015

PROTOCOLOS DE EVALUACIÓN

- **K-fold cross-validation,**
  Escogencia del K:
  - Permite balancear entre sesgo y varianza
  - **LOOCV** (Leave One Out Cross-Validation): partes de tamaño 1
  - Por defecto se estima que los mejores resultados se obtienen con un valor de K entre 5 y 10

Sebastian Raschka, 2015

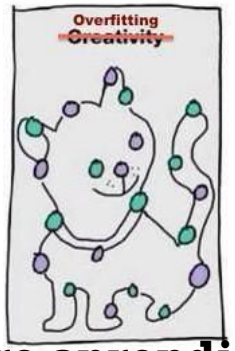
TALLER MÉTRICAS DE CLASIFICACIÓN EN EXCEL Y PROTOCOLOS DE EVALUACIÓN

Con el dataset de MNIST (que viene en Keras):

- Desarrollar el taller de Python que compara un clasificador Naïve Bayes de ML tradicional, con uno que utiliza un red neuronal convolucional (CNN)
- Utilizar un protocolo Holdout para evaluar la calidad de los modelos aprendidos, estableciendo si se esta en overfitting, underfitting o si se tiene un buen compromiso entre sesgo y varianza
- Para el clasificador NaIveBayes, en una hoja de Excel, calcular las métricas de clasificación generales accuracy, error y kappa, así como las de precision, recall, especificidad específicas para cada una de las categorías que se presentan en los resultados del clasificador de MNIST con NaiveBayes.

AGENDA

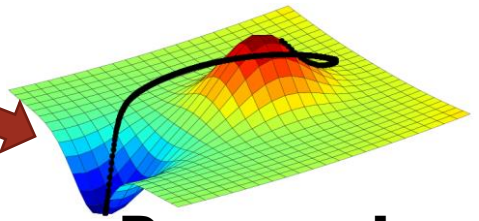
Deep Learning



Sobre aprendizaje (Overfitting)

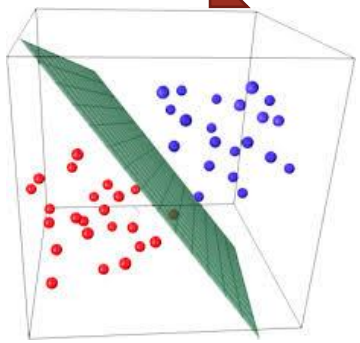
Aprendizaje supervisado

Protocolos de evaluación

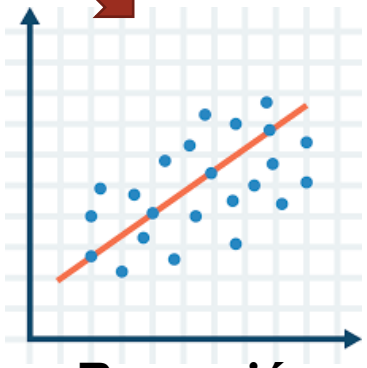


Descenso de gradiente

Métricas de Evaluación de la clasificación



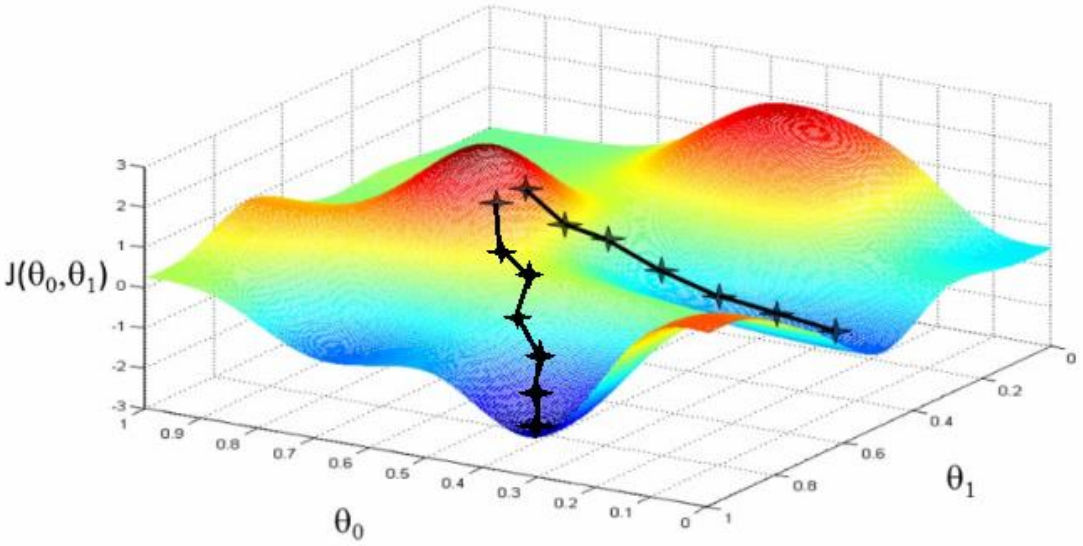
Clasificación



Regresión

DESCENSO DE GRADIENTE

Aplicación a la regresión lineal



DESCENSO DE GRADIENTE

- **Notación**:
  - Los parámetros de los modelos se pueden denotar por θ_k o w_k, y el número total es n
  - El número de instancias de aprendizaje van de 1 a m
  - La k-ésima variable independiente (de un total de n), para la i-ésima instancia de aprendizaje se denotan como $x_k^{(i)}$
- Ilustraremos el proceso de descenso de gradiente para la **regresión lineal.**
- **Función de costo o de pérdida (loss) J**: función objetivo que se debe optimizar durante el proceso de aprendizaje, a partir del ajuste de los parámetros del modelo.
  - Varía en función de los parámetros θ_i del modelo.
  - En el caso de la regresión lineal múltiple:

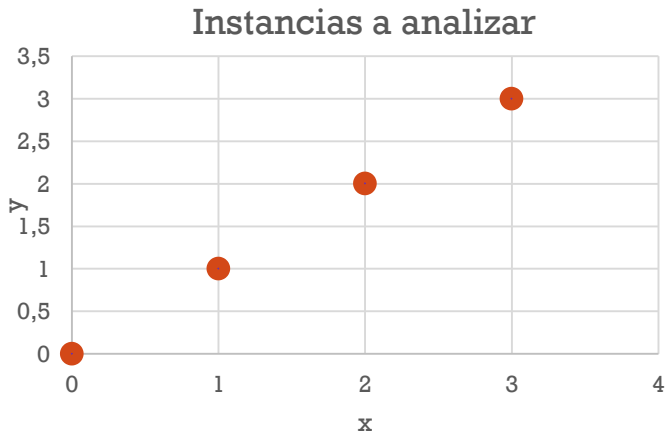
$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x_1^{(i)} + \cdots + \theta_n x_n^{(i)} - y^{(i)})^2$$

→ Cada combinación de parámetros $\theta_0, \theta_1, \dots, \theta_n$ corresponde a un modelo lineal diferente

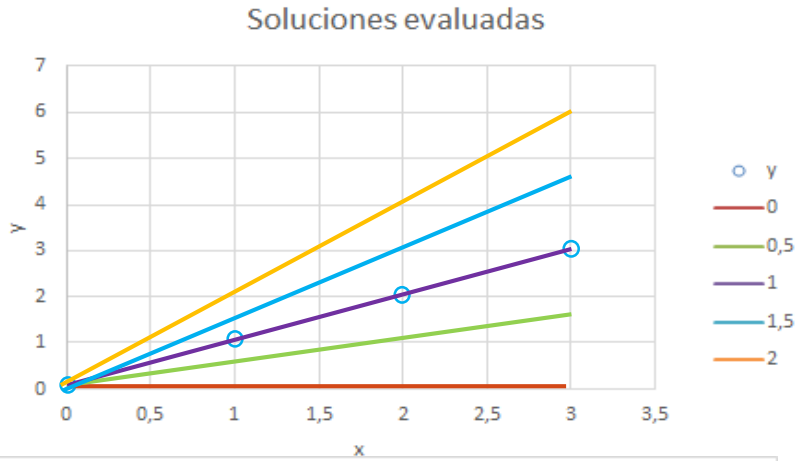
DESCENSO DE GRADIENTE

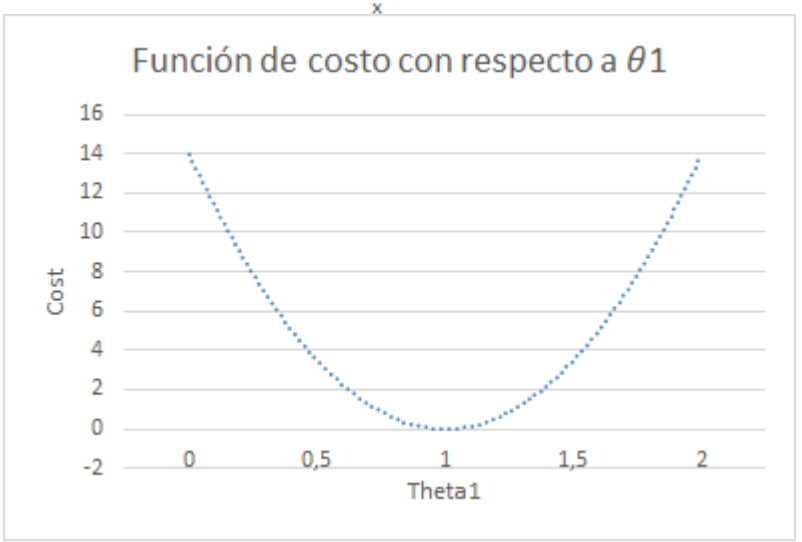
		Valores de θ1									
		0	0,5	1	1,5	2	0	0,5	1	1,5	2
x	y	y est. = θ1*x					residuo=(θ1*x-y est.)^2				
0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0,5	1	1,5	2	1	0,25	0	0,25	1
2	2	0	1	2	3	4	4	1	0	1	4
3	3	0	1,5	3	4,5	6	9	2,25	0	2,25	9
						J	14	3,5	0	3,5	14

- **Ejemplo**: tenemos los datos siguientes.
  - ¿Cuáles son los valores de θ_0 y θ_1?



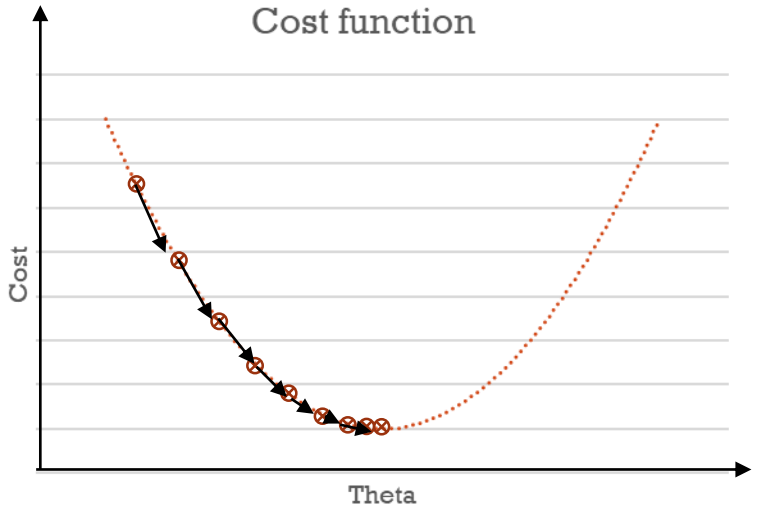
  - $\theta_0 = 0$ y $\theta_1 = 1$
  - Partiendo del hecho de que sabemos que $\theta_0 = 0$, evaluemos la función de costo para θ_1, haciendo variar su valor entre $\{0, 0.5, 1, 1.5, 2\}$

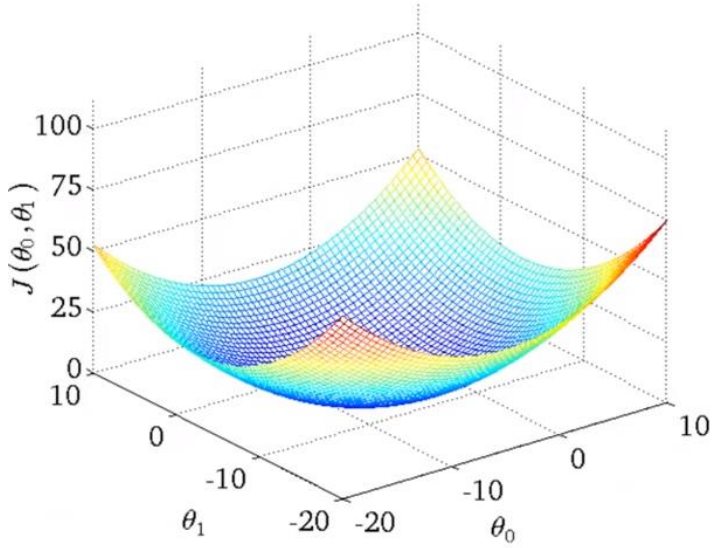


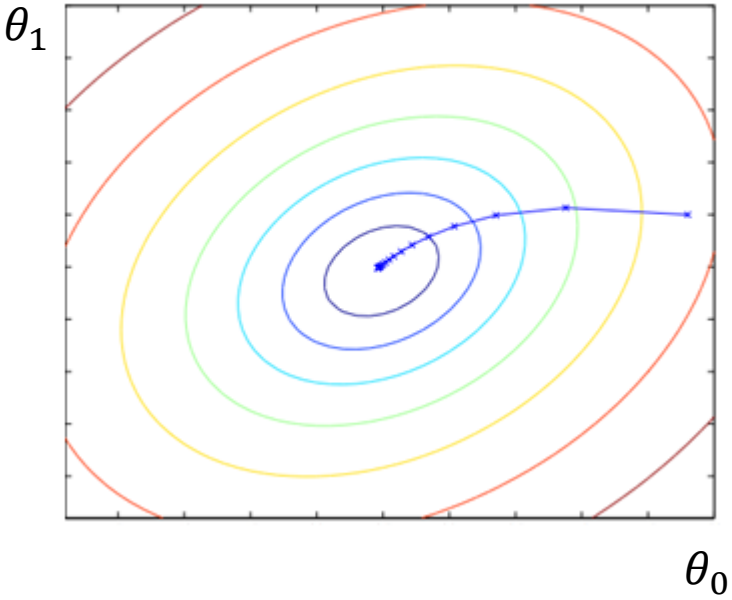


DESCENSO DE GRADIENTE

Descenso de gradiente







DESCENSO DE GRADIENTE

- Algoritmo:
  1. Escoger aleatoriamente valores para cada parámetro θ_i.
  2. Actualizar todos los θ_i **simultáneamente**: $\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\Theta)$. Donde $J(\Theta)$ es la **función de costo** que deseamos minimzar. Nos basamos en las derivadas parciales para encontrar la dirección que se debe seguir para actualizar los parámetros de tal manera que se minimice la función de costo.
  3. Parar cuando se llegue a convergencia (mínimo del costo)
- α es el **learning rate** (**taza de aprendizaje)** y controla el nivel de actualización de los parámetros. Es importante no escoger un α ni muy pequeño, ni muy grande (como veremos más adelante)
- No hay garantía de llegar a un mínimo **global**, puede que se alcance un mínimo **local**

DESCENSO DE GRADIENTE

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x_1^{(i)} + \cdots + \theta_n x_n^{(i)} - y^{(i)})^2$$

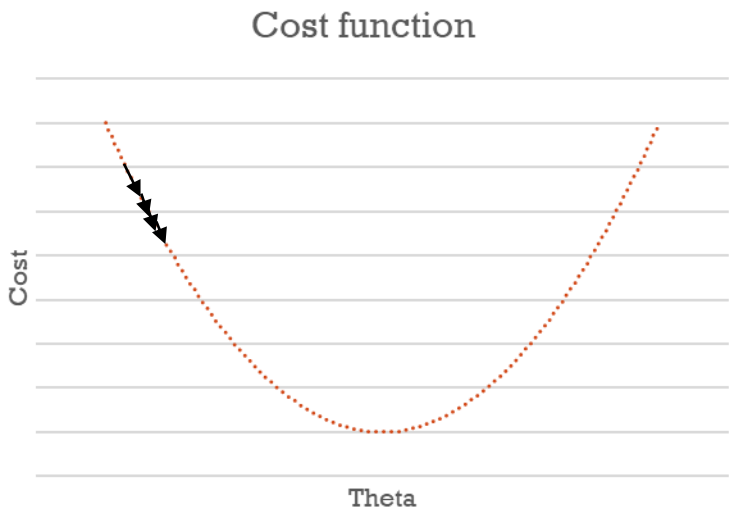
$$\left.\begin{array}{l} \frac{\partial}{\partial \theta_0} J(\Theta)?? \\ \frac{\partial}{\partial \theta_1} J(\Theta)?? \end{array}\right\} \theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\Theta)$$

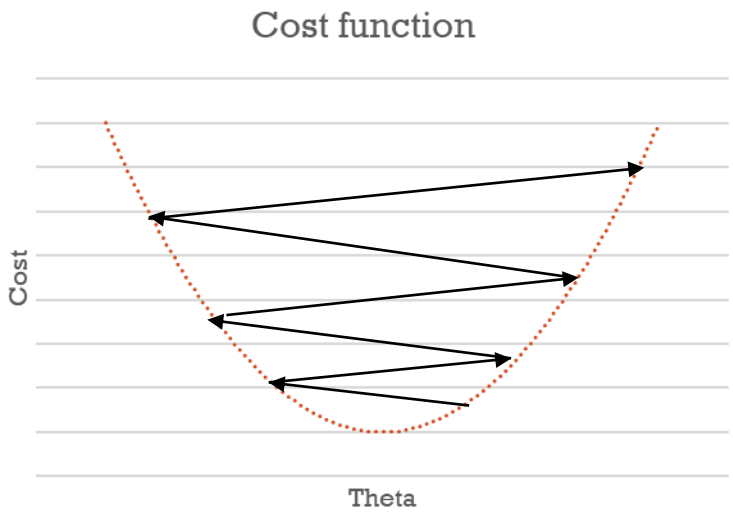
→ Desarrollar analíticamente las soluciones

- Solución para la regresión lineal múltiple:
  - Para la intercepción: $\theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x_1^{(i)} + \cdots + \theta_n x_n^{(i)} - y^{(i)}\right)$
  - Para los coeficientes: $\theta_j := \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x_1^{(i)} + \cdots + \theta_n x_n^{(i)} - y^{(i)}\right) * x_j^{(i)}$

DESCENSO DE GRADIENTE

- Escogencia de la **taza de aprendizaje** α:
  - Si α muy pequeño: demorado llegar a convergencia
  - Si α muy grande: demorado llegar a convergencia, peligro de divergencia



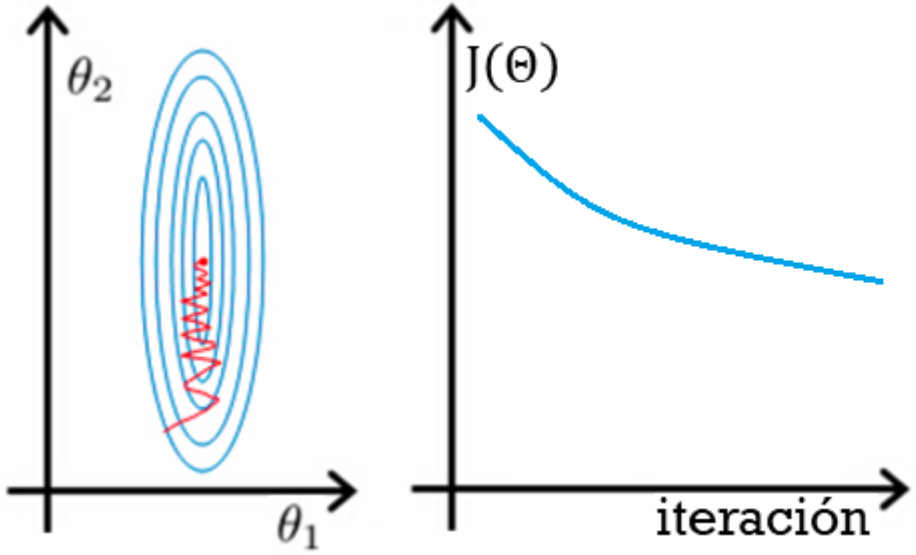


  - α debe decrecer siempre en cada iteración; en caso contrario, se debe reducir el valor de α
  - Se debe intentar con varios valores de α: 0.001, 0.01, 0.1, 1

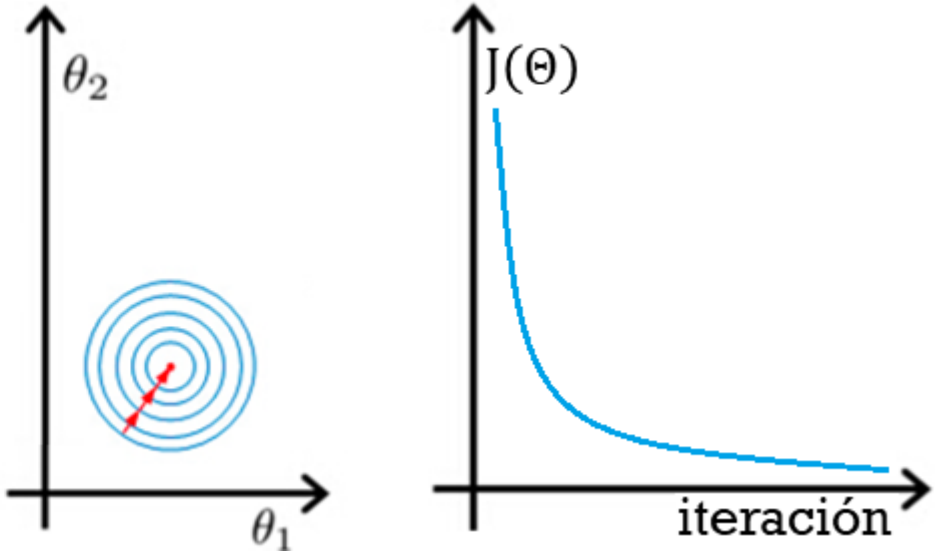
DESCENSO DE GRADIENTE

Feature Scaling

- Si escalas muy diferentes: demorado llegar a convergencia, sobre influencia de las derivadas parciales de las variables de mayor escala iteración



- Si misma escala: influencia igual de las derivadas parciales de todos los parámetros

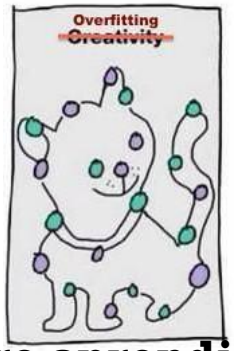


TALLER GRADIENT DESCENT

Desarrollar el taller de gradient descent para la regresión lineal utilizando la librería numpy

AGENDA

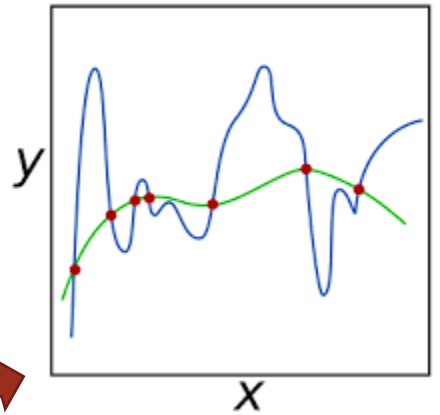
Deep Learning



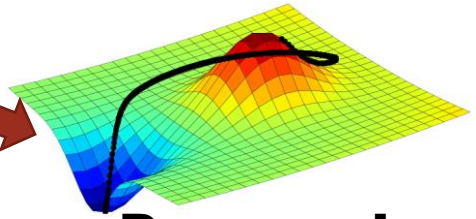
Sobre aprendizaje (Overfitting)

Aprendizaje supervisado

Protocolos de evaluación

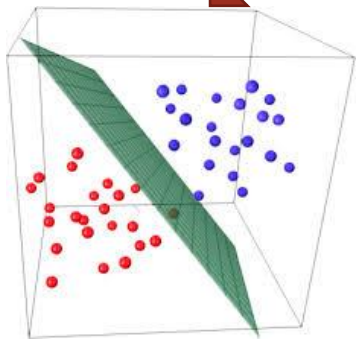


Regularización

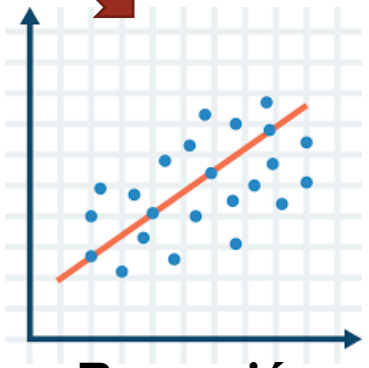


Descenso de gradiente

Métricas de Evaluación de la clasificación

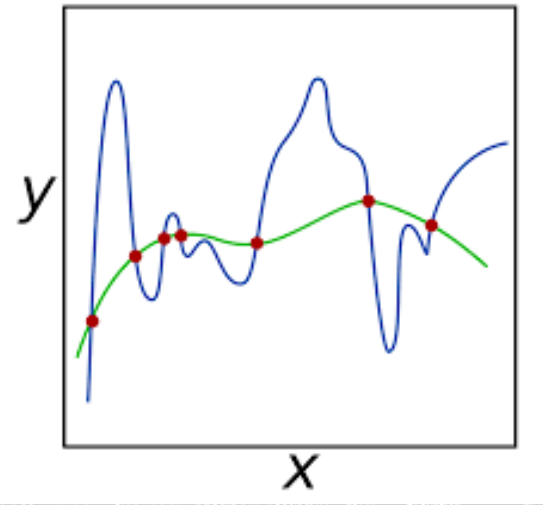


Clasificación



Regresión

REGULARIZACIÓN: RIDGE Y LASSO



Aplicación a la regresión lineal

REGULARIZACIÓN DE LA REGRESIÓN

- **Técnicas de regularización**: penalizan la magnitud de sus coeficientes de las variables independientes al mismo tiempo que se trata de minimizar los errores de predicción.
- Se quiere minimizar la complejidad de los modelos
  - Disminuir la posibilidad de sobre-aprendizaje del modelo
  - Controlar los requerimientos computacionales de tener muchas variables independientes (big data)
- Se cambia la función de costo a minimizar:
  - **Ridge**:

$$J(\Theta) = \sum_{i=1}^{n} (\theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m - y_i)^2 + \frac{\lambda}{2} * \sum_{j=1}^{m} \theta_i^2$$

  - **Lasso**:

$$J(\Theta) = \sum_{i=1}^{n} (\theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m - y_i)^2 + \lambda * \sum_{j=1}^{m} |\theta_i|$$

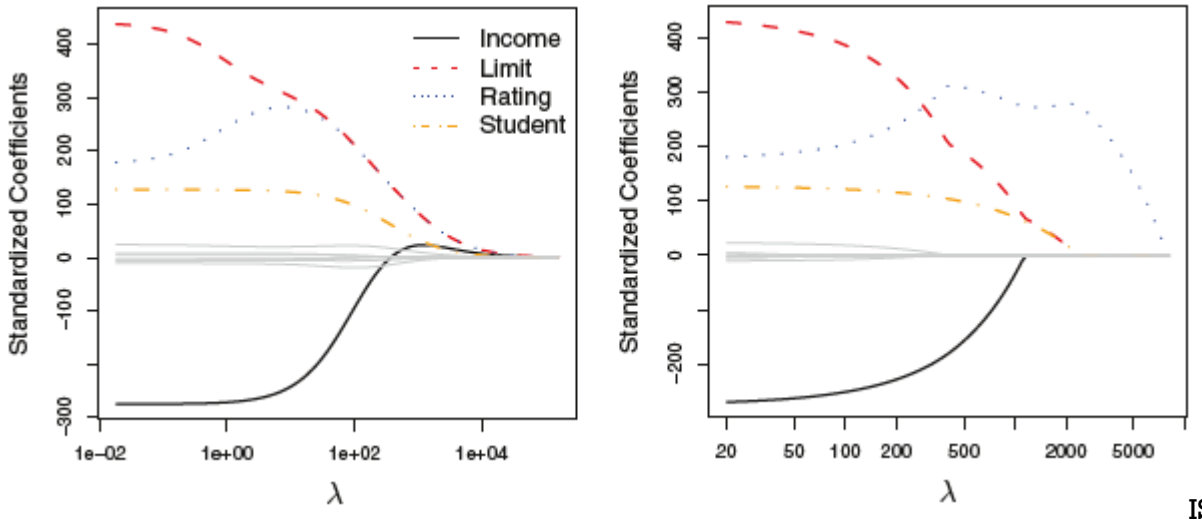
No se penaliza θ_0

REGULARIZACIÓN DE LA REGRESIÓN

- El parámetro λ sirve para controlar el impacto relativo de los dos términos en la función de costo (el que minimiza el error, y el que limita la complejidad)
  - Cuando $\lambda = 0$, la penalidad no tiene efecto
  - Entre más grande λ, los coeficientes obtenidos van a ser más pequeños
  - Seleccionar un valor de λ es crítico; se usa Cross-Validation
- Se actualiza la actualización de los coeficientes en el algoritmo de descenso de gradiente.
  - Ridge: $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{j=1}^{m} \left(\theta_0 + \theta_1 x_1^{(j)} + \cdots + \theta_n x_n^{(j)} - y^{(j)} \right) * x_j^{(j)} + \frac{\lambda}{m} \theta_j$
  - Lasso: [muy complicado, incluye definición de subgradiente, dado que la función de valor absoluto no es derivable]
- ElasticNet: modelo de regresión con regularización que combina las penalizaciones de Ridge y Lasso

REGULARIZACIÓN DE LA REGRESIÓN

- Ridge lleva a coeficientes pequeños de los atributos
  - →Todos los atributos predictores están presentes en el modelo
  - →El modelo resultante es difícil de interpretar si hay muchas variables predictivas
- Lasso hace desaparecer los coeficientes menos importantes
  - →Método de selección de atributos, forzando los coeficientes de los atributos eliminados a 0, si λ es suficientemente grande



ISLR, 2013

REGULARIZACIÓN DE LA REGRESIÓN

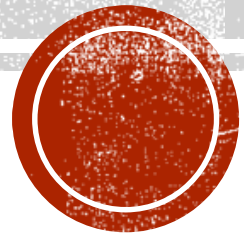
Consideraciones:

- Ridge:
  - Correlación de las variables independientes: funciona bien, pues aunque incluya todas las variables sus coeficientes van a distribuirse considerando sus correlaciones
  - Usada para prevenir el overfitting
- Lasso:
  - Correlación de las variables independientes:
    - Escoge arbitrariamente cualquiera de las variables altamente correlacionadas, poniendo en 0 los coeficientes de las demás.
    - Puede haber problemas en caso de términos polinomiales que puedan desaparece al estar correlacionados entre ellos.
  - Produce modelos mas simples, robustos con respecto al overfitting, y fáciles de interpretar.
  - Usada como método de selección de atributos a considerar cuando hay miles de variables independientes → produce modelos "dispersos" (sparse)

TALLER LASSO Y RIDGE

Desarrollar el taller de regresión Lasso y Ridge sobre el dataset de Jugadores de Baseball (Hitters)

GRACIAS



UNIVERSIDAD ICESI

REFERENCIAS

- *Introduction to Statistical Learning with Applications in R (ISLR)*, G. James, D. Witten, T. Hastie & R. Tibshirani, Springer, 2014
- *Python Machine Learning (2nd ed.)*, Sebastian Raschka & Vahid Mirjalili, Packt, 2017
- *Data Science for Business*, Foster Provost & Tom Fawcett, O'Reilly, 2013
- *Machine Learning,* Tom M. Mitchell, McGraw-Hill, 1997