A Snapshot of the Frontiers of Fairness in Machine Learning++

Communications of the ACM, May 2020, Vol. 63, N°5.

Introducción

Supongamos que queremos hacer un modelo predictor para asignar el salario de los trabajadores de una empresa

Introducción

Supongamos que queremos hacer un modelo predictor para asignar el salario de los trabajadores de una empresa

Para hacer este modelo predictor vamos a usar los datos históricos de la empresa

Introducción

Supongamos que queremos hacer un modelo predictor para asignar el salario de los trabajadores de una empresa

Para hacer este modelo predictor vamos a usar los datos históricos de la empresa

Tenemos razones para creer que estos datos son sesgados (por ejemplo, por el género de la persona)

¿Tomará nuestro modelo decisiones justas?


METALLICA
...AND JUSTICE FOR ALL

Antes unos preliminares

Ejemplo

Asistencia a un curso

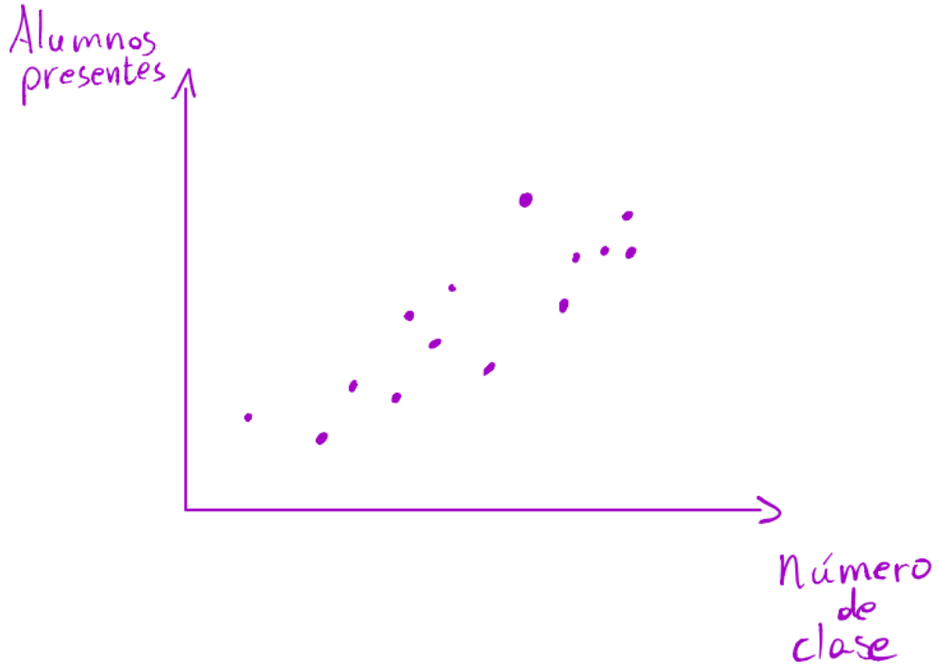
Queremos predecir la asistencia a la i-ésima clase de *Testing* y tenemos los siguientes datos:

Número de clase	Asistencia
1	10
2	20
...	...

Ejemplo

Asistencia a un curso

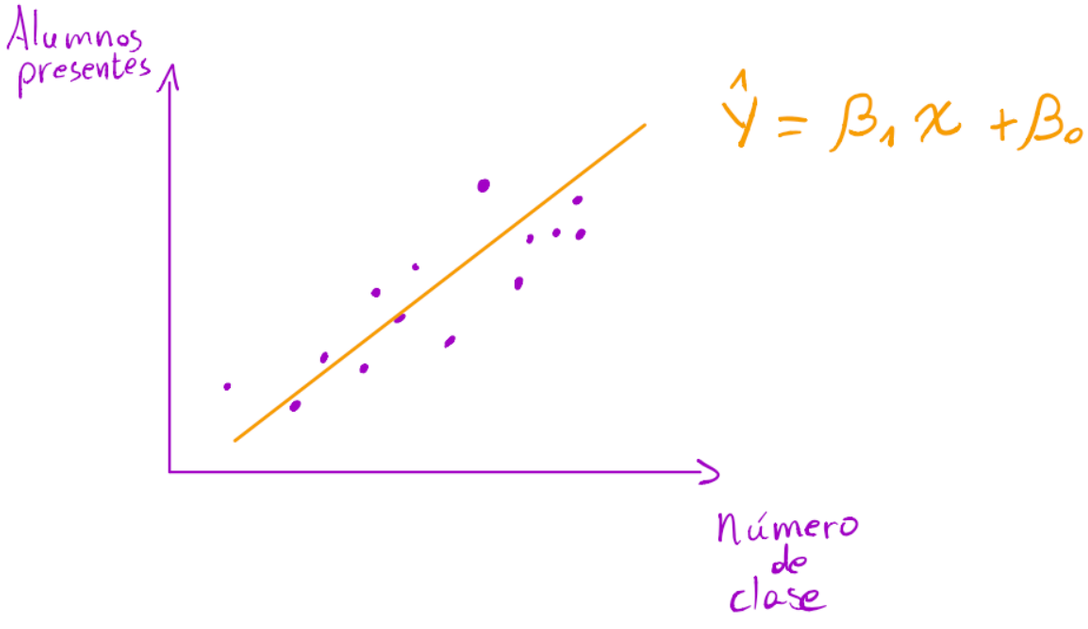
Al visualizar el *dataset* obtenemos lo siguiente:



Ejemplo

Asistencia a un curso

Esta es una tarea de **regresión**, en la que buscamos predecir un valor numérico



Lo más típico es hacer una regresión lineal, pero hay muchas opciones!

Ejemplo

Asistencia a un curso

Ojo, aquí partimos con **datos** y a partir de estos **aprendimos** los valores de los coeficientes de la recta:

$$\hat{y} = \beta_1 x + \beta_0$$

Y de ahora en adelante, cuando tengamos una clase i-ésima (desconocida) podremos predecir su asistencia

Ejemplo

Predecir cuando un programa fallará

Ahora vamos a intentar predecir si un programa va a terminar exitosamente en función a las líneas de código y a la cantidad de pruebas realizadas

Ejemplo

Predecir cuando un programa fallará

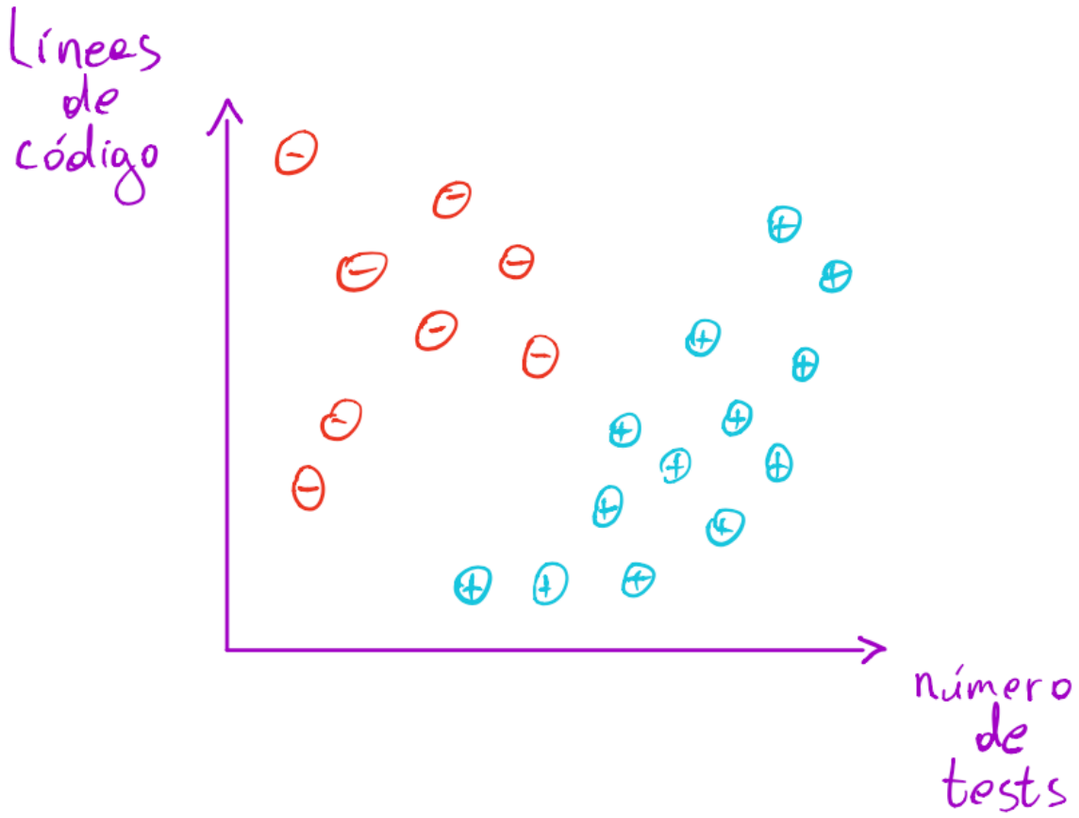
Nuevamente, tenemos datos históricos:

Número de lineas	Cantidad de pruebas	Finaliza exitosamente
10	0	No
10	4	Sí
...	...	...

Ejemplo

Predecir cuando un programa fallará

Al graficar nuestro *dataset* obtenemos lo siguiente:



Ejemplo

Predecir cuando un programa fallará

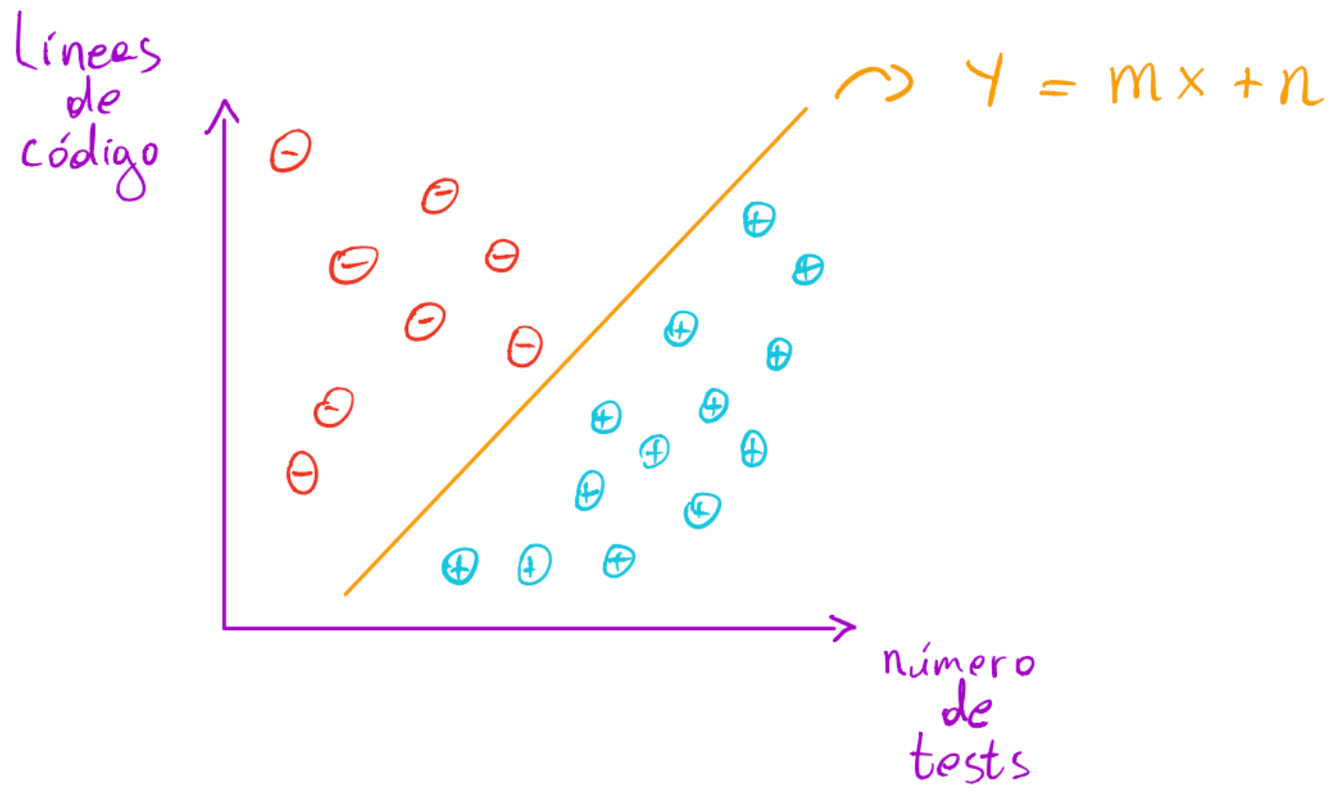
Ahora lo que queremos es encontrar una recta que divida el plano, así al encontrarnos con instancias desconocidas veremos de que lado de la recta queda esta instancia, y así la **clasificaremos**

Así, decimos que esta es una tarea de **clasificación**

Ejemplo

Predecir cuando un programa fallará

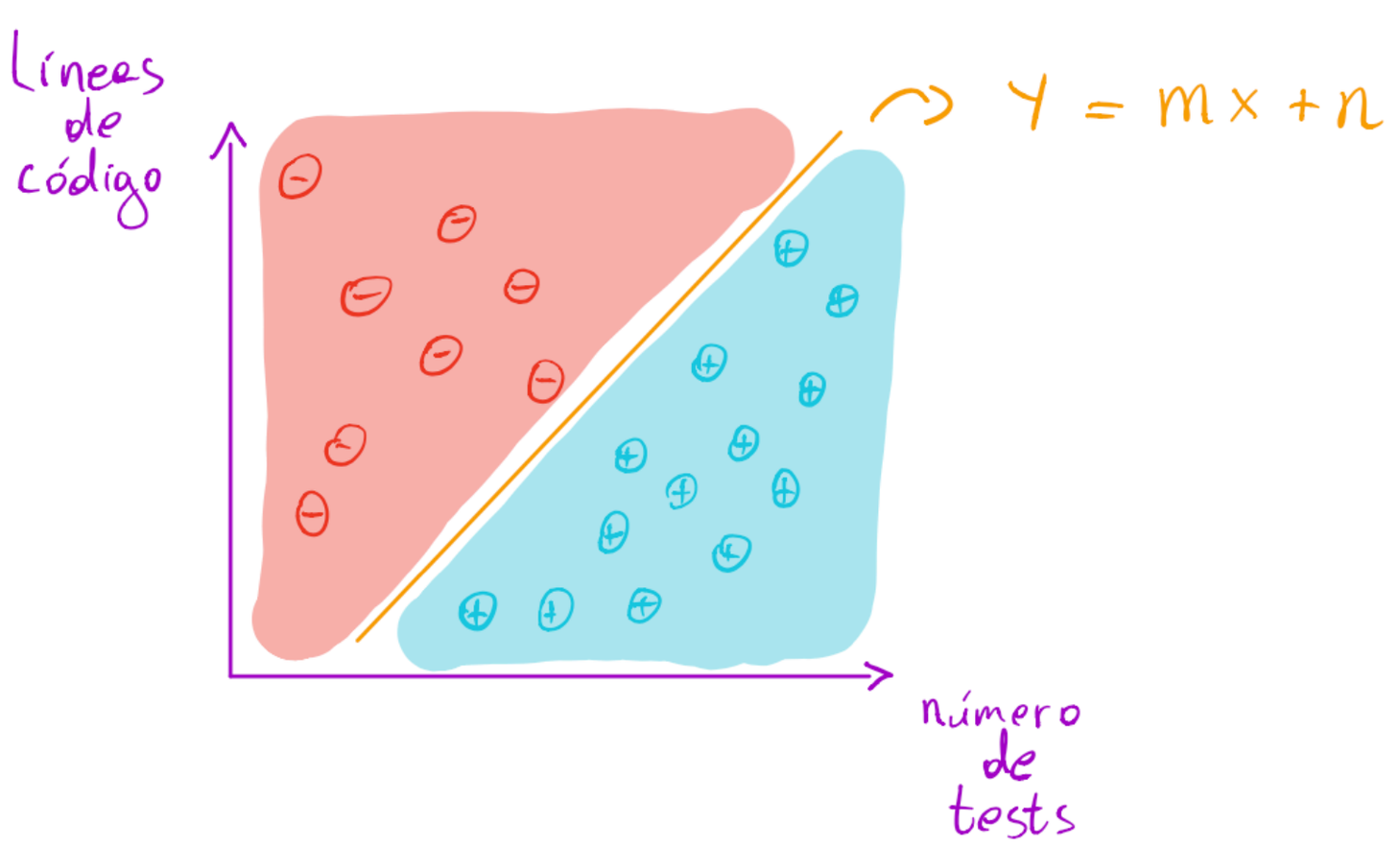
Así, al aprender de los datos históricos, generamos la siguiente recta:



Ejemplo

Predecir cuando un programa fallará

Y cuando queramos predecir una **instancia desconocida** vamos a predecir la clase representada por el lado de la recta en que quedó



Ejemplo

Predecir cuando un programa fallará

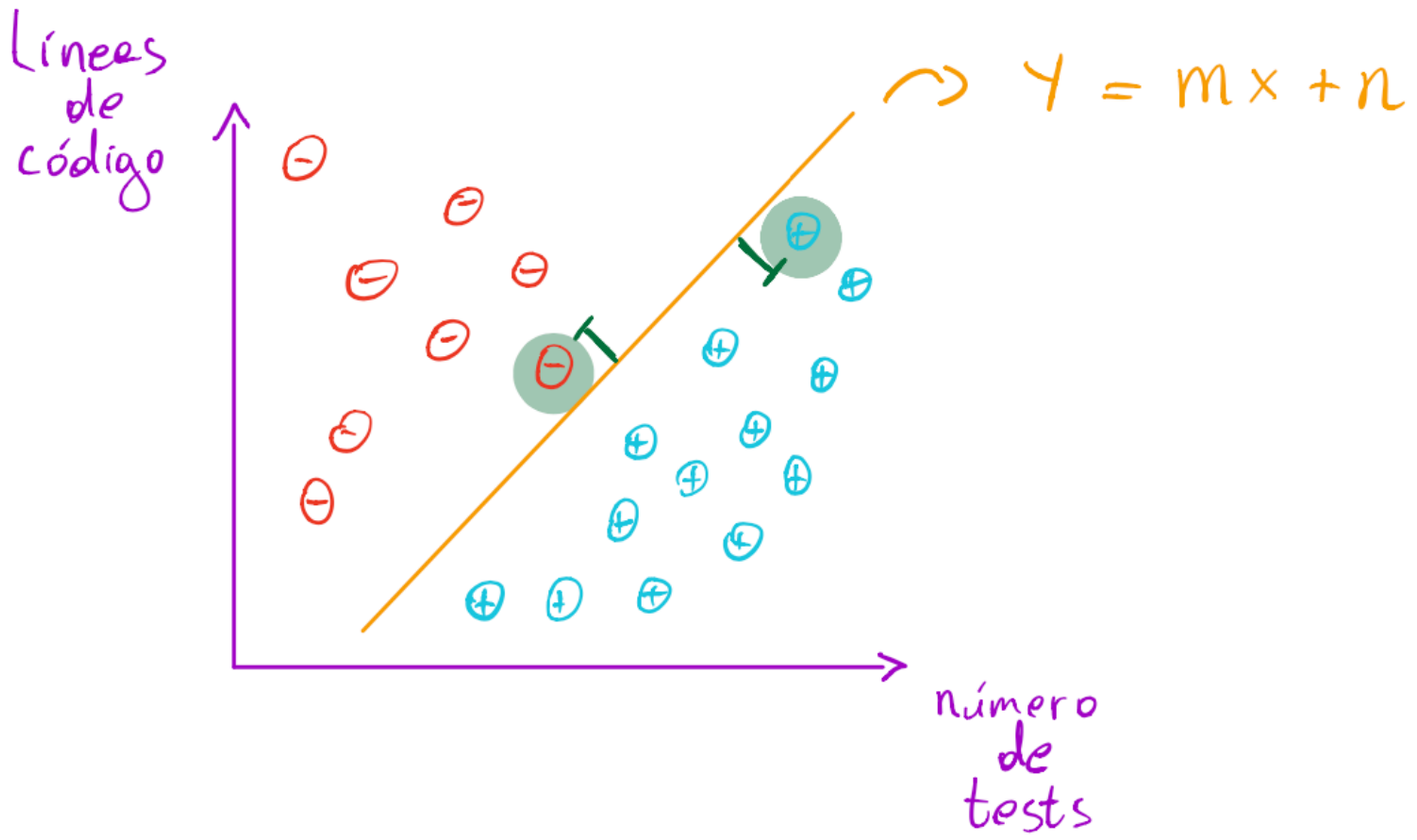
Existen varios algoritmos para encontrar los coeficientes de la recta:

- Regresión logística
- SVM
- Redes Neuronales
- ...

Ejemplo

Predecir cuando un programa fallará

Por ejemplo, SVM busca una recta que maximice el margen entre las dos clases:

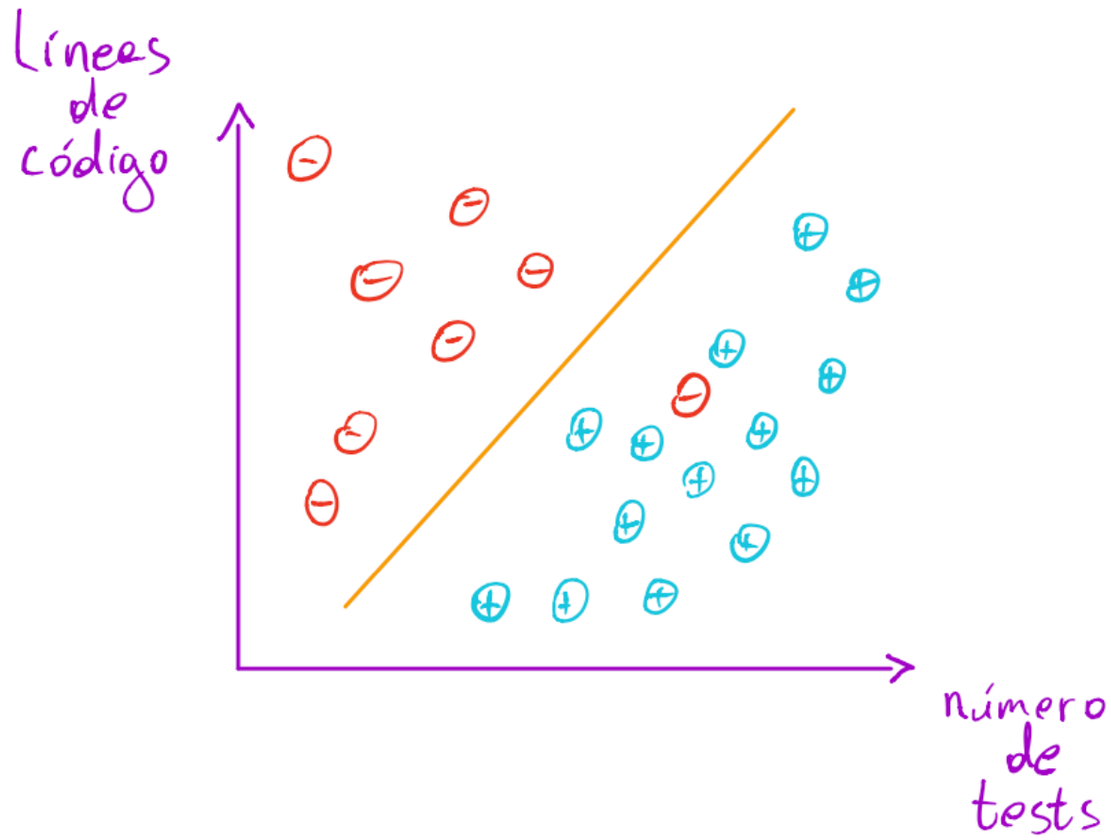


Modelos predictores

Vamos a entender un modelo predictor como una función que, después de aprender de datos históricos, asigna un valor (una clase o valor numérico) a una instancia desconocida

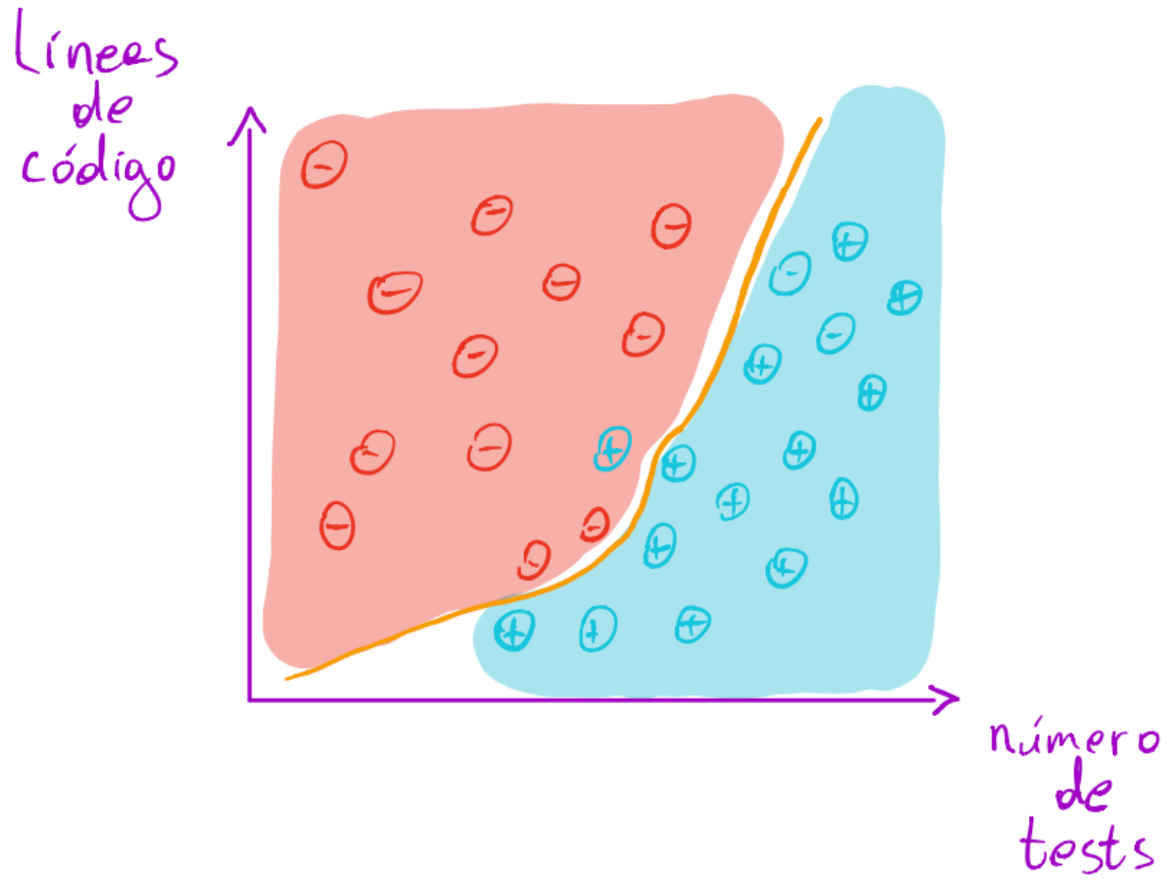
Modelos predictores

Por su puesto, cuando los *datasets* no son "separables, funcionan igual:



Modelos predictores

Y también podemos hacer clasificación no lineal:

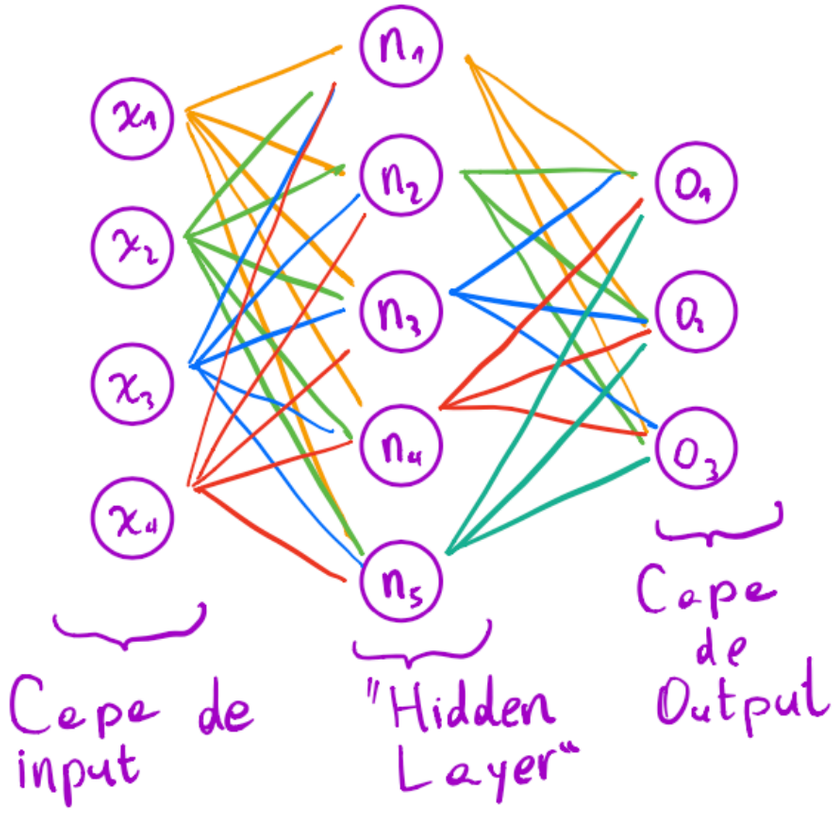


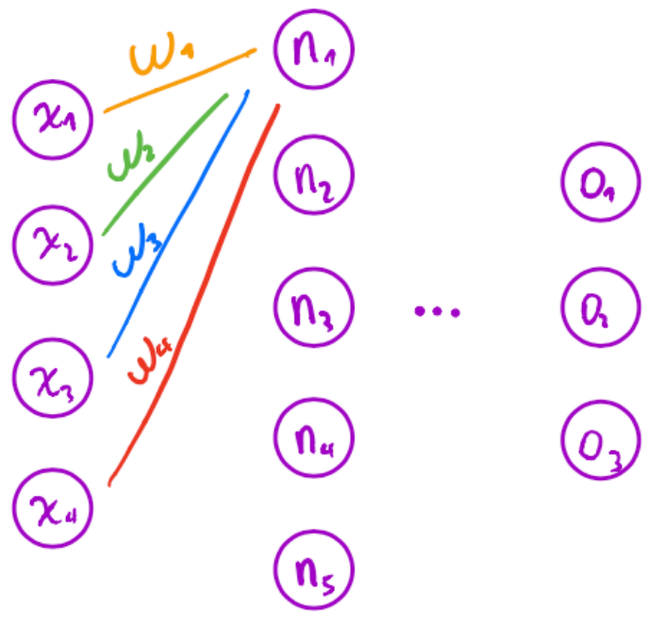
Modelos predictores

Pero en general siempre se trata de aprender coeficientes de datos históricos, y luego predecir es ingresar la instancia desconocida a la función que ya aprendió los coeficientes

Modelos predictores

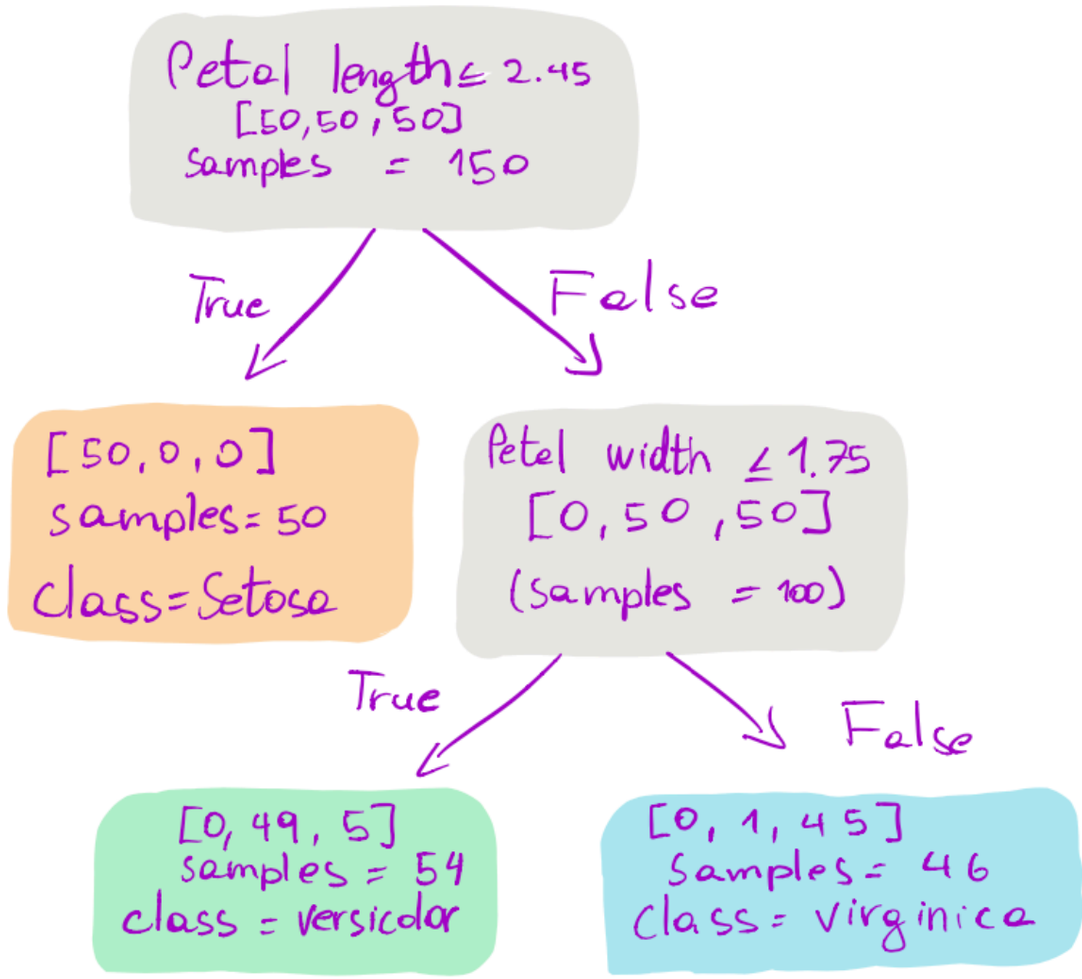
Por ejemplo, en una red neuronal tenemos:





Modelos predictores

O en un árbol de decisión:



Modelos predictores

Y obviamente esto funciona en las dimensiones que queramos (no solo dos), porque buscamos hiperplanos que nos dividan el espacio

Además podemos clasificar a más de dos clases, como por ejemplo, cuando reconocemos dígitos escritos a mano

Modelos predictores

Y en general, aprender significa resolver un problema de optimización, por ejemplo en SVM:

$$\min \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum \alpha_i$$

$$\text{s.a. } \alpha_i \geq 0$$

En el ejemplo anterior queremos encontrar los valores óptimos de α_i, que nos permitirán encontrar los coeficientes de nuestra recta

Modelos predictores

Estos problemas de optimización se suelen resolver de forma numérica (e.j. *Gradient Descent*, programación cuadrática, ADAM)

Y es importante notar que los x e y en la función objetivo son nuestros datos!

Evaluación de desempeño

En general, dividimos nuestro *dataset* en la parte de entrenamiento y la de prueba

Entrenamos (i.e. aprendemos nuestros parámetros) con los datos de entrenamiento

Evaluamos el desempeño con los datos de prueba (e.j. *accuracy*, matriz de confusión,...)

Evaluación de desempeño

Matriz de confusión

Sirve para entender nuestros errores

		Valor predicción	
		Negativo (0)	**Positivo (1)**
Valor instancia	**Negativo (0)**	TN	FP
	Positivo (1)	FN	TP

Ahora un poco de historia

Machine Learning, 20, 273–297 (1995)

Support-Vector Networks

CORINNA CORTES corinna@neural.att.com
VLADIMIR VAPNIK vlad@neural.att.com
AT&T Bell Labs., Holmdel, NJ 07733, USA

Editor: Lorenza Saitta

Abstract. The *support-vector network* is a new learning machine for two-group classification problems. The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space a linear decision surface is constructed. Special properties of the decision surface ensures high generalization ability of the learning machine. The idea behind the support-vector network was previously implemented for the restricted case where the training data can be separated without errors. We here extend this result to non-separable training data.

High generalization ability of support-vector networks utilizing polynomial input transformations is demonstrated. We also compare the performance of the support-vector network to various classical learning algorithms that all took part in a benchmark study of Optical Character Recognition.

Keywords: pattern recognition, efficient learning algorithms, neural networks, radial basis function classifiers, polynomial classifiers.

Ahora seguimos con el paper de *fairness*

Fairness en Machine Learning

Los modelos de Machine Learning cada vez abarcan más cambios, por ejemplo:

- Filtrar postulantes para otorgar créditos
- Selección de personal
- Aplicaciones en seguridad
- Apoyo en decisiones de salud

Sin embargo, a la vez hay preocupaciones de que estos modelos introduzcan y perpetúen **prácticas discriminatorias o injustas**

Se ha mostrado que los modelos que aprenden en base a datos, pueden aprender sesgos humanos e incluso introducir algunos nuevos

Fairness en Machine Learning

El tópico de Fairness en Machine Learning ha pasado de ser tópico "nicho" a ser el mayor subcampo del área, con las siguientes preguntas:

- ¿Qué debería significar *fairness* en este contexto?
- ¿Cuáles son las causas que introducen injusticia en un modelo de *machine learning*?
- ¿Cómo evitamos las injusticias?
- ¿Cuáles son los *trade-off* que debemos transar?

Causas de *unfairness*

Datos sesgados. Como discutimos en el ejemplo, los datos vienen sesgados por causas humanas. De aquí podemos tener varias causas de *unfairness*

Causas de *unfairness*

Minimizar el error y poblaciones mayoritarias. Imaginemos un modelo de *machine learning*; para minimizar su función objetivo, en caso de no poder ajustarse a todos los datos, puede buscar ajustarse más a las poblaciones más grandes

Esto producirá errores futuros para las clases minoritarias

Causas de *unfairness*

La necesidad de explorar. Supongamos que queremos hacer un modelo que trabaje sobre los efectos de la dosis un medicamento; para obtener datos, necesitaremos probar opciones sub-óptimas

¿Es esto justo?

Definiciones de fairness

Statistical Fairness. Se fija un número pequeño de grupos protegidos G, para después pedir paridad sobre medidas estadísticas en esos grupos (tasa de falsos positivos, negativos, etc)

Lo bueno de estas medidas es que son simples y fáciles de comprobar, pero no dan garantías a miembros individuales, sino que da garantías al promedio de los miembros de los grupos protegidos

Definiciones de fairness

Ejemplo - *Demographic Parity*

En el contexto de *fairness* es común hablar de atributos protegidos

Podemos entender un atributo protegido p como una *feature* que representa cosas como sexo, etnia, entre otros

No queremos que las decisiones dependan de este atributo!

Definiciones de fairness

Ejemplo - *Demographic Parity*

Una medida que los toma en cuenta es *Demographic Parity*

Sea una predicción $\hat{y}$, donde $y = \{0, 1\}$, tener *Demographic Parity* implica que:

$$Pr(\hat{y} \mid p) = Pr(\hat{y})$$

Con p un atributo protegido

Definiciones de fairness

Ejemplo - *Demographic Parity*

Así, en general vamos a medir la *Statistical Parity Difference*

$$SPD = Pr(\hat{y} = 1,\, p = 1) - Pr(\hat{y} = 1,\, p = 0)$$

Y vamos a querer que esta medida esté en cierto rango acotado

Definiciones de fairness

Ejemplo - *Equality of Opportunity*

Otra medida común es la "Igualdad de Oportunidades", que señala lo siguiente:

$$Pr(\hat{y} \mid y = 1, p) = Pr(\hat{y} \mid y = 1)$$

Esto significa que **la tasa de verdaderos positivos** debe ser la misma para cada población

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

Definiciones de fairness

Ejemplo - *Equality of Odds*

Una generalización de la medida anterior es *Equality of Odds*

$$Pr(\hat{y} \mid y, p) = Pr(\hat{y} \mid y)$$

Esto significa que la tasa de verdaderos positivos debe ser igual para ambos grupos, y la tasa de falsos positivos debe ser igual para ambos grupos

Definiciones de fairness

Ojo, se ha demostrado que en el contexto de fairness **no es posible** satisfacer todas las medidas a la vez (salvo en instancias triviales)

Definiciones de fairness

Definiciones individuales. Se busca cumplir propiedades entre pares de individuos; la premisa es "individuos similares deben ser tratados similarmente"

Se requieren supuestos fuertes, lo que hace que estas ideas sean muy difíciles de implementar en la práctica

Fairness: estado actual

Statistical fairness

Trabajos más recientes buscan establecer definiciones estadísticas sobre infinitas clases de grupos, definidas por ciertas funciones; además no se requieren supuestos sobre los datos

Fairness: estado actual

Statistical fairness

Preguntas de investigación:

- ¿Qué función de clases son razonables? (ej. conjunción de atributos protegidos)
- ¿Qué *features* deben ser protegidas?
- ¿Son atributos sensibles por si solos (género, raza) o en combinación con otro son sensibles (estilo de vestir)?

Fairness: estado actual

Individual fairness

Sobre nociones de *individual fairness* que exijan supuestos menos fuertes, se han desarrollado modelos que asumen acceso a un oráculo que puede entregar medidas de similaridad o bien encontrar violaciones de *fariness*

Dynamic of Fairness

Todas las medidas de *fairness* se han hecho sobre tareas de clasificación, ¿pero qué pasa en sistemas de *machine learning* más complejos?

Un problema sencillo aún no resuelto es entender cómo y cuando un modelo de hartas componentes de ML en que cada una se considera *fair* sigue siendo *fair* después de juntar estas componentes

Dynamic of Fairness

También es importante entender como los algoritmos van modificando el ambiente

Un ejemplo: algoritmo recomienda bajar la exigencia de ingreso a educación superior para grupos de riesgo

Habrán más familias con educación superior y se incentiva el esfuerzo a prepararse académicamente

Dynamic of Fairness

Se han buscado ideas del campo de la economía para entender como los efectos de los modelos en el ambiente mediante procesos de decisión de Markov

Corregir los sesgos

Pero las preocupaciones sobre fairness se dan en ocasiones cuando los datos ya vienen sesgados

Entender cómo se producen los sesgos en los datos y como corregirlos son los desafíos fundamentales en el área de *fairness*

Corregir los sesgos

Ejemplo

Se ha mostrado que los algoritmos de *machine learning* tienden a captar los sesgos de género

Los algoritmos tienden a asociar que "doctor" es más similar a masculino, o bien analogías como "man is to computer programmer as woman is to homemaker"

Corregir los sesgos

Ejemplo

Estos sesgos se pueden dar además cuando hay *feedback loops*: esto es, el modelo entrenado afecta en cómo se colectan datos en el futuro

Ejemplo. Supongamos un modelo que predice sectores con mayor delincuencia

La policía hará más esfuerzos en zonas con más arrestos, lo que inducirá más arrestos en esa zona en el futuro y perpetuará el sesgo

Corregir estos sesgos requiere entender cómo el proceso de obtención de datos es sesgado, además de emitir juicios sobre propiedades que los datos deberían tener en un mundo "justo"

Best paper SIGMOD 2019

Interventional Fairness : Causal Database Repair for Algorithmic Fairness

Babak Salimi[1] Luke Rodriguez[2] Bill Howe[2] Dan Suciu[1]

ABSTRACT

Fairness is increasingly recognized as a critical component of machine learning systems. However, it is the underlying data on which these systems are trained that often reflect discrimination, suggesting a database repair problem. Existing treatments of fairness rely on statistical correlations that can be fooled by statistical anomalies, such as Simpson's paradox. Proposals for causality-based definitions of fairness can correctly model some of these situations, but they require specification of the underlying causal models. In this paper, we formalize the situation as a database repair problem, proving sufficient conditions for fair classifiers in terms of admissible variables as opposed to a complete causal model. We show that these conditions correctly capture subtle fairness violations. We then use these conditions as the basis for database repair algorithms that provide provable fairness guarantees about classifiers trained on their training labels. We evaluate our algorithms on real data, demonstrating improvement over the state of the art on multiple fairness metrics proposed in the literature while retaining high utility.

Técnicas de Causalidad

Idea de este paper

Supongamos que tengo un *dataset* con distintas *features*, uno puede definir el grafo (DAG) causal de estas *features* (pensar en dependencias funcionales)

Se definen interacciones entre variables admisibles e inadmisibles

Definimos el grafo causal **que nos gustaría tener**

Técnicas de Causalidad

Idea de este paper

Usamos técnicas de reparación de bases de datos para lograr tener el grafo causal que queremos

Esto es eliminar/insertar un conjunto minimal de tuplas que satisfaga las dependencias entre atributos que buscamos

Representaciones justas

Algo que se busca en el área de *fairness* es tener herramientas para transformar los datos en que se mantenga la información relevante pero se protejan los atributos sensibles

¿Y si eliminamos las columnas (*features*) problemáticas?

En general, no es llegar y eliminar columnas:

- Podemos tener algún *proxy* a las variables protegidas.
- Queremos tratar de mantener en lo posible el desempeño de nuestros predictores

Model Remediation

Otra técnica para crear modelos justos, es forzar al modelo a "entrenarse" de manera justa

Esta técnica se conoce como *Model Remediation*, y en general consiste en cambiar el problema de optimización por una versión que toma en cuenta restricciones de *fairness*

Model Remediation

Ejemplo - MinDiff

Un ejemplo de esto es MinDiff, técnica propuesta por Google

Se basa en agregar regularización al modelo, castigando la correlación entre la respuesta y los grupos protegidos

Model Remediation

Ejemplo - MinDiff

$$Loss(X, y) = L_{\text{primary}}(f(X), y) + \lambda \cdot \text{correlation}(f(X), A \mid y = 0)$$

Función de optimización original

Se castiga por la correlación entre la predicción y el valor de la variable A del dataset (que nos dice a qué grupo pertenece) en todas las instancias negativas

Model Remediation

Ejemplo - MinDiff

En el ejemplo anterior se busca igualar la tasa de falsos positivos para ambos grupos (¿por qué?)

Hay formas mejores de medir la dependencia estadística entre la predicción y el grupo al que pertenece una instancia (ej. MMD)

Looking Forward

La idea es entender las nociones de *fairness* en entornos dinámicos (y más complejos, fuera de simplemente clasificación), como cuando el algoritmo influye en los datos que habrán en el futuro

Por ejemplo, no podremos saber si hubiese pagado alguien a quien no le damos un crédito

Looking Forward

Además hay que tener en mente el hecho de que para tener información mejor, a veces hay que tomar decisiones sub-óptimas (tratamientos médicos)

Esto se da en general para *settings* fuera de clasificación binaria, donde la literatura aún no es amplia

A Snapshot of the Frontiers of Fairness in Machine Learning++

Communications of the ACM, May 2020, Vol. 63, N°5.