

Curso 1 - Introducción al aprendizaje automático en la producción

En el primer curso de la especialización en ingeniería de aprendizaje automático para la producción, identificará los distintos componentes y diseñará un sistema de producción de ML de principio a fin: alcance del proyecto, necesidades de datos, estrategias de modelado y limitaciones y requisitos de despliegue; y aprenderá a establecer una línea de base del modelo, a abordar la deriva del concepto y a crear un prototipo del proceso para desarrollar, desplegar y mejorar continuamente una aplicación de ML en producción.

Entender los conceptos de aprendizaje automático y aprendizaje profundo es esencial, pero si quieres construir una carrera efectiva en el campo de la IA, también necesitas capacidades de ingeniería de producción. La ingeniería de aprendizaje automático para la producción combina los conceptos fundamentales del aprendizaje automático con la experiencia funcional de los roles modernos de desarrollo de software e ingeniería para ayudarlo a desarrollar habilidades listas para la producción. Semana 1: Visión general del ciclo de vida de ML y su implementación, Semana 2: Selección y entrenamiento de un modelo, Semana 3: Definición de datos y línea de base

Semana 3: Definición de datos y línea de base

Contenido

Semana 3: Definición de datos y línea de base	1
¿Por qué es difícil definir los datos?	1
Más ejemplos de ambigüedad de etiquetas	3
Principales tipos de problemas de datos	3
Datos pequeños y consistencia de las etiquetas	8
Mejorar la coherencia de las etiquetas	10
Rendimiento a nivel humano (HLP)	13
Aumentar la HLP	15
Obtención de datos	17
Canalización de datos	20
Metadatos, procedencia de los datos y linaje.....	22
Equilibrio entre el entrenamiento, el desarrollo y la prueba	23
Qué es el alcance	24
Proceso de evaluación	25
Diligencia sobre la viabilidad y el valor	27
Diligencia sobre el valor	30
Hitos y recursos	32
Referencias	32

¿Por qué es difícil definir los datos?

- Voy a utilizar el ejemplo de la detección de iguanas

Iguana detection example



Labeling instructions: "Use bounding boxes to indicate the position of iguanas"

- A tres etiquetadores se les ocurren estas tres formas diferentes de etiquetar las iguanas, y puede que cualquiera de ellas esté bien. Yo preferiría las dos primeras antes que la tercera.
- Pero lo que no está bien es que 1/3 de sus etiquetadores utilice la 1ª convención y 1/3 la 2ª, y 1/3 la 3ª convención de etiquetado
- Entonces sus etiquetas serán inconsistentes, y esto es confuso para el algoritmo de aprendizaje.

Phone defect detection



- Si le pides a un etiquetador que utilice recuadros delimitadores para indicar los defectos significativos, puede que un etiquetador mire y diga: "Bueno, está claro que el arañazo es el defecto más significativo".
- Un segundo etiquetador puede mirar el teléfono y decir: "En realidad hay dos defectos importantes. Hay un gran arañazo, y luego está esa pequeña marca de fosa". Pero entonces un tercer etiquetador puede mirar esto y decir, bueno, aquí hay un cuadro delimitador que muestra dónde están todos los defectos.
- Entre estas tres etiquetas, probablemente la del medio es la que mejor funciona. Pero este es un ejemplo muy típico de etiquetado incoherente que se obtiene de un proceso de etiquetado, con instrucciones de etiquetado ligeramente ambiguas
- Muchos investigadores e ingenieros de aprendizaje automático empezaron descargando datos de Internet para experimentar con modelos, es decir, utilizando datos preparados por otra persona. No hay nada malo en ello, pero para muchas aplicaciones prácticas, la forma en que prepares tus conjuntos de datos tendrá un enorme impacto en el éxito de tus proyectos de aprendizaje automático.

Más ejemplos de ambigüedad de etiquetas

Speech recognition example



"Um, nearest gas station"

"Umm, nearest gas station"

"Nearest gas station [unintelligible]"

- Teniendo en cuenta este clip de audio, parece que alguien estaba de pie en el borde de una carretera muy transitada preguntando por la gasolinera más cercana y entonces pasó un coche.
- Hay muchas formas combinatorias de transcribir esto. Con una M o con dos M, con coma o con elipsis, ya sea para escribir ininteligible al final de esto.
- Ser capaz de estandarizar una convención ayudará a su algoritmo de reconocimiento de voz.

Principales tipos de problemas de datos

User ID merge example

	Job Board (website)	Resume chat (app)
Email	nova@deeplearning.ai	nova@chatapp.com
First Name	Nova	Nova
Last Name	Ng	Ng
Address	1234 Jane Way	?
State	CA	?
Zip	94304	94304

Handwritten notes:

- is it a bot/spam account?
- fraudulent transaction?
- looking for job?

Legend:

- 1 if same
- 0 if different

- Una aplicación común en muchas grandes empresas es la fusión de **ID de usuario**. Es cuando se tienen varios registros de datos que se cree que corresponden a la misma persona, y se quiere fusionar estos registros de datos de usuario.
- Por ejemplo, digamos que usted dirige un sitio web que ofrece listados de empleos en línea. Puede tratarse de un registro de datos que tiene de uno de sus usuarios registrados con el correo electrónico, el nombre, los apellidos y la dirección.
- Ahora, digamos que su empresa adquiere una segunda empresa que tiene una aplicación móvil que permite a la gente conectarse, chatear y recibir consejos de los demás sobre sus currículos.
- Parece sinérgico para su negocio, y ahora tiene una base de datos diferente de usuarios.
- Teniendo en cuenta este registro de datos y este otro (mostrado en la diapositiva), ¿cree que estos dos son la misma persona?

- Un enfoque para el problema de la fusión de ID de usuario es el uso de un algoritmo de aprendizaje supervisado que toma los registros de datos de los usuarios como entrada y da como resultado uno o cero en función de si cree que estos dos son realmente el mismo ser humano.
- Si tienes una forma de obtener registros de verdad, como por ejemplo si un puñado de usuarios está dispuesto a vincular explícitamente las dos cuentas, entonces eso podría ser un buen conjunto de ejemplos etiquetados para entrenar un algoritmo.
- Pero si no se dispone de ese conjunto de datos de verdad, lo que muchas empresas han hecho es pedir a trabajadores humanos, a veces a un equipo de gestión de productos, que miren manualmente algunos pares de registros que han sido filtrados para tener quizás nombres similares o códigos postales similares, y luego usar el juicio humano para determinar si estos dos registros parecen ser la misma persona.
- Si estos dos registros se refieren realmente a la misma persona puede ser realmente ambiguo.
- Pueden y no pueden ser personas diferentes
- Si hay una forma de conseguir que etiqueten los datos de forma un poco más consistente, incluso cuando la verdad sobre el terreno es ambigua, puede ayudar al rendimiento de su algoritmo de aprendizaje.
- La fusión de ID de usuario es una función muy común en muchas empresas, así que permítanme pedirles que por favor hagan esto sólo de manera que sean respetuosos con los datos de los usuarios y su privacidad, y sólo si están usando los datos de una manera que sea consistente con lo que ellos les han dado permiso.
- Algunos otros ejemplos de datos estructurados. Si se trata de utilizar el algoritmo de aprendizaje para mirar la cuenta del usuario predecir si es un bot o una cuenta de spam. O si miras una compra online y compruebas si se trata de una transacción fraudulenta. A veces eso también es ambiguo.
- Ante tareas de predicción potencialmente muy importantes y valiosas como éstas, la verdad sobre el terreno puede ser ambigua.
- Si se pide a los usuarios que adivinen la etiqueta verdadera para tareas como éstas, dar instrucciones de etiquetado que den lugar a etiquetas más coherentes y menos ruidosas y aleatorias mejorará el rendimiento del algoritmo de aprendizaje.

Data definition questions

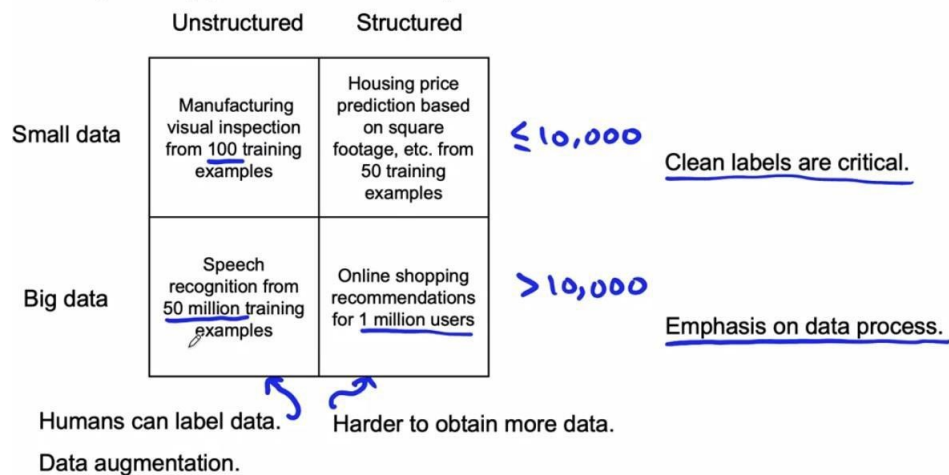
- What is the input x ?
 - Lighting? Contrast? Resolution?
 - What features need to be included?
- What is the target label y ?



- A la hora de definir los datos para su algoritmo de aprendizaje, hay algunas cuestiones importantes.
- En primer lugar, ¿cuál es la entrada x ?
- Por ejemplo, si está intentando detectar defectos en los teléfonos inteligentes, para las fotos que está tomando, ¿es la iluminación lo suficientemente buena? ¿Es el contraste de la cámara lo suficientemente bueno? ¿Es la resolución de la cámara lo suficientemente buena?
- Si te encuentras con que tienes un montón de fotos como éstas, que son tan oscuras que es difícil incluso para una persona ver lo que está pasando, puede que lo correcto **no** sea coger esta entrada x y simplemente etiquetarla.
- Tal vez sea mejor ir a la fábrica y pedir amablemente que mejoren la iluminación, porque sólo con esta mejor calidad de imagen el etiquetador puede ver mejor los arañazos como éste y etiquetarlos.
- Si tu sensor o tu solución de imagen o tu solución de grabación de audio no son lo suficientemente buenos, lo mejor que puedes hacer es reconocer que si una persona no puede mirar la entrada y decirnos qué está pasando, entonces mejorar la calidad de tu sensor o mejorar la calidad de la entrada x es el primer paso importante para asegurar que tu algoritmo de aprendizaje tenga un rendimiento razonable.
- En el caso de los problemas de datos estructurados, definir qué características incluir puede tener un gran impacto en el rendimiento de su algoritmo de aprendizaje.

- fundamental. Si tienes 100 ejemplos de entrenamiento, si sólo uno de los ejemplos está mal etiquetado, eso es el 1% de tu conjunto de datos (que tengas permiso del usuario para usarla), puede ser una herramienta muy útil para decidir si dos cuentas de usuario pertenecen realmente a la misma persona.
- Además de definir la entrada x , también hay que averiguar cuál debe ser la etiqueta de destino y .
- Como se ha visto en los ejemplos anteriores, una cuestión clave es cómo garantizar que las etiquetas sean coherentes. En el último vídeo y en este, has visto una serie de problemas de ambigüedad en las etiquetas o, en algunos casos, que la entrada x no es lo suficientemente informativa (por ejemplo, imágenes demasiado oscuras)

Major types of data problems



- Resulta que las mejores prácticas para organizar los datos de un tipo pueden ser bastante diferentes de las mejores prácticas para tipos totalmente diferentes. Veamos esta cuadrícula de 2×2 .
- Un eje será si su problema de aprendizaje automático utiliza datos no estructurados o datos estructurados.
- Descubrí que las mejores prácticas para estos son muy diferentes, principalmente porque los humanos son excelentes para procesar datos no estructurados, las imágenes y el audio y el texto, y no tan buenos para procesar datos estructurados como los registros de las bases de datos.
- El segundo eje es el tamaño del conjunto de datos. No hay una definición precisa de lo que es exactamente pequeño y lo que es grande. Pero voy a utilizar como un umbral ligeramente arbitrario, si usted tiene más de **10.000** ejemplos o no.
- He elegido el número **10.000** porque es el tamaño aproximado a partir del cual resulta bastante doloroso examinar cada uno de los ejemplos.
- Veamos algunos ejemplos. Si estás entrenando una inspección visual de fabricación a partir de sólo 100 ejemplos de teléfonos estirados, se trata de datos no estructurados con un conjunto de datos bastante pequeño. Si estás tratando de predecir los precios de las viviendas basándote en el tamaño de los salones y otras características, a partir de sólo 52 ejemplos, entonces hay un conjunto de datos estructurados con un conjunto de datos pequeño
- Si se realiza el reconocimiento del habla a partir de 50 millones de ejemplos de entrenamiento, se trata de datos no estructurados con un gran conjunto de datos
- Si está haciendo recomendaciones de compras en línea y tiene un millón de usuarios en su base de datos, entonces es un problema estructurado con una cantidad relativamente grande de datos.
- Para la inspección de la visión de fabricación, puede utilizar **el aumento de datos** para generar más imágenes de películas inteligentes o para el reconocimiento de voz. El aumento de datos también puede ayudarle a sintetizar clips de audio con diferentes ruidos de fondo.
- En cambio, para los problemas de datos estructurados, puede ser más difícil obtener más datos y también más difícil utilizar el aumento de datos. Es difícil sintetizar nuevos usuarios que no existen realmente, y también es más difícil (puede o no ser posible) conseguir que los humanos etiqueten los datos.
- Así que me parece que las mejores prácticas para los datos no estructurados frente a los estructurados son bastante diferentes
- Cuando se dispone de un conjunto de datos relativamente pequeño, tener etiquetas limpias es fundamental. Si tienes 100 ejemplos de entrenamiento, si sólo uno de los ejemplos está mal etiquetado, eso es el 1% de tu conjunto de datos.

- Y como el conjunto de datos es lo suficientemente pequeño como para que usted o un pequeño equipo lo revisen de forma eficiente, puede valer la pena revisar esos 100 ejemplos. Es importante asegurarse de que cada uno de esos ejemplos esté etiquetado de forma limpia y coherente, de acuerdo con una norma de etiquetado consistente
- En cambio, si tiene un millón de puntos de datos, puede ser más difícil (casi imposible) revisar manualmente cada ejemplo.
- Tener las etiquetas limpias sigue siendo muy útil, ya que las etiquetas limpias son mejores que las no limpias.
- Pero debido a la dificultad de hacer que el equipo pase manualmente por cada ejemplo, se hace hincapié en los **procesos de datos**, en términos de cómo se recogen los datos, se instalan los datos, las instrucciones de etiquetado que se escriben para un gran equipo de etiquetadores crowdsourced.

Unstructured vs. structured data

Unstructured data

- May or may not have huge collection of unlabeled examples x .
- Humans can label more data.
- Data augmentation more likely to be helpful.

Structured data

- May be more difficult to obtain more data.
- Human labeling may not be possible (with some exceptions).

- En el caso de los problemas de datos no estructurados, se puede tener o no una enorme colección de ejemplos sin etiquetar x .
- Puede que en su fábrica hayan tomado miles de imágenes de teléfonos inteligentes, pero no se han molestado en etiquetarlas todas todavía. Esto también es común en la industria de los coches autoconducidos, donde muchas empresas de coches autoconducidos han recogido toneladas de imágenes de coches circulando, pero simplemente no han cogido todavía esos datos etiquetados.
- Para estos problemas de datos no estructurados, se pueden obtener más datos pidiendo a los humanos que simplemente etiqueten más datos sin etiquetar
- Y como ya hemos mencionado, el aumento de datos también puede ser útil.
- En el caso de los problemas de datos estructurados, suele ser más difícil obtener más datos, ya que sólo hay un número determinado de usuarios de los que se pueden recoger datos.
- Y el etiquetado humano, en promedio, también es más difícil, aunque hay algunas excepciones, como cuando se vio que podíamos pedir a la gente que etiquetara ejemplos para la fusión de ID de usuario.
- Pero en muchos casos en los que pedimos a los humanos que etiqueten los datos de la estructura, incluso cuando vale la pena pedir a la gente que intente etiquetar si dos registros son la misma persona, es probable que haya una mayor ambigüedad en la que incluso al etiquetador humano le resulte difícil estar seguro de cuál es la etiqueta correcta.

Small data vs. big data

Small data

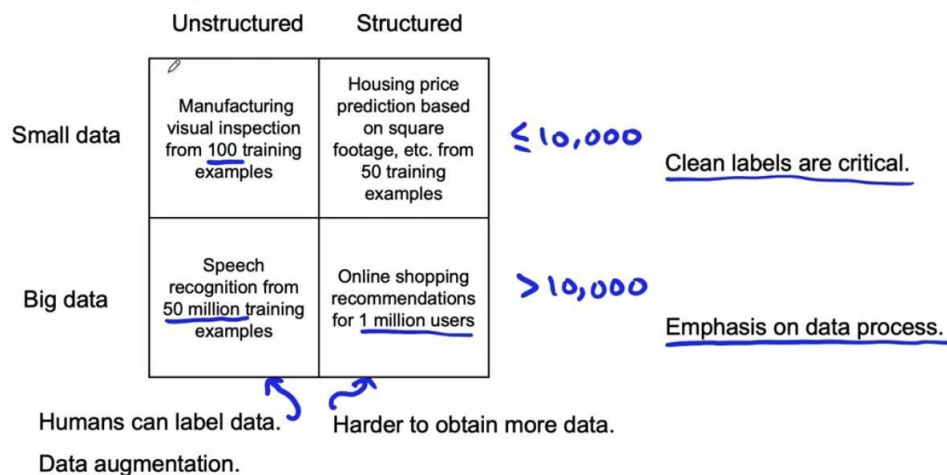
- Clean labels are critical.
- Can manually look through dataset and fix labels.
- Can get all the labelers to talk to each other.

Big data

- Emphasis data process.

- Analicemos los datos pequeños frente a los grandes, en los que he utilizado un umbral ligeramente arbitrario de si tienes más o menos de 10.000 ejemplos de entrenamiento.
- En el caso de conjuntos de datos pequeños, la limpieza de las etiquetas es fundamental y el conjunto de datos puede ser lo suficientemente pequeño como para que usted pueda revisar manualmente todo el conjunto de datos y corregir cualquier etiqueta incoherente.
- Además, el equipo de etiquetado probablemente no sea tan numeroso, por lo que es más fácil que hablen entre ellos y se pongan de acuerdo en una convención de etiquetado.
- Para los conjuntos de datos muy grandes, hay que hacer hincapié en el proceso de datos. Y si tienes 100 etiquetadores o incluso más, es más difícil reunir a 100 personas en una sala para que hablen entre sí y discutan el proceso.
- Por ello, es posible que tenga que recurrir a un equipo más pequeño para establecer una definición coherente de la etiqueta y luego compartir esa definición con el equipo más grande de etiquetadores y pedirles que todos apliquen el mismo proceso.

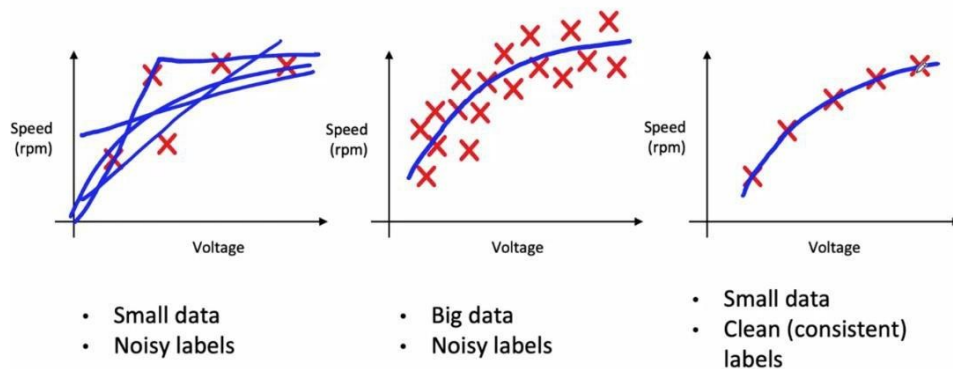
Major types of data problems



- Quiero dejaros con una última reflexión, y es que esta categorización de los problemas en no estructurados frente a estructurados, pequeños frente a grandes datos, me ha parecido útil
- Si estás trabajando en un problema de uno de estos cuatro cuadrantes, aceptar el consejo de alguien que ha trabajado en problemas en el mismo cuadrante será probablemente más útil que el consejo de alguien que ha trabajado en un cuadrante diferente.
- También he descubierto que, al contratar ingenieros de aprendizaje automático, alguien que ha trabajado en el mismo cuadrante que el problema que estoy tratando de resolver suele ser capaz de adaptarse más rápidamente a trabajar en otros problemas de ese cuadrante.
- Porque los instintos y las decisiones son más similares dentro de un cuadrante que si se cambia a otro totalmente diferente.
- A veces he oído consejos como: si estás construyendo un sistema de visión por ordenador, consigue siempre al menos 1000 ejemplos etiquetados.
- Creo que la gente que da esos consejos tiene buenas intenciones y aprecio que intenten dar buenos consejos, pero me parece que esos consejos no son realmente útiles para todos los problemas.
- El aprendizaje automático es muy diverso y es difícil encontrar un consejo único para todos.
- He visto problemas de visión por ordenador contruidos con 100 ejemplos o incluso sistemas contruidos con 100 millones de ejemplos.
- Así que si buscas consejo para un proyecto de aprendizaje automático, intenta encontrar a alguien que haya trabajado en el mismo cuadrante que el problema que intentas resolver.

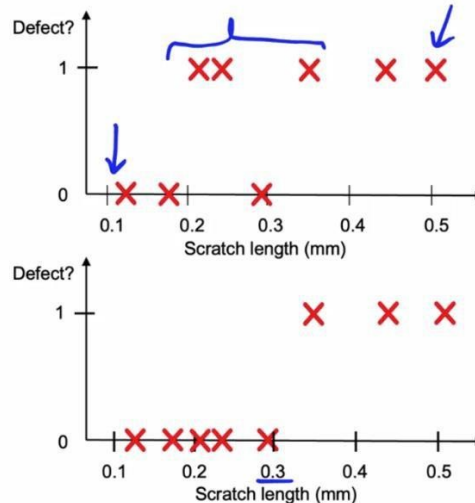
Datos pequeños y coherencia de las etiquetas

Why label consistency is important



- En un conjunto de datos pequeño, tener etiquetas limpias y coherentes es especialmente importante.
- Una de las cosas que solía hacer es utilizar el aprendizaje automático para volar helicópteros. Lo que podría hacer es introducir el voltaje aplicado al motor o al rotor del helicóptero y predecir cuál es la velocidad del rotor.
- Digamos que tienes un conjunto de datos como este, donde tienes cinco ejemplos.
- A partir del pequeño conjunto de datos, la salida Y es bastante ruidosa, por lo que es difícil saber la función que hay que utilizar para asignar la tensión a la velocidad del rotor en rpm.
- Tal vez debería ser una línea recta, o tal vez una curva como esa. Es realmente difícil de decir cuando se tiene un pequeño conjunto de datos, especialmente con etiquetas ruidosas
- Ahora bien, si tuviéramos datos igual de ruidosos que los de la izquierda, pero tuviéramos muchos más datos. El algoritmo de aprendizaje puede entonces promediar los datos ruidosos y ahora se puede ajustar una función con bastante confianza
- Últimamente se ha desarrollado mucha IA en grandes empresas de Internet de consumo que pueden tener 100 millones de usuarios o mil millones de usuarios y, por tanto, con conjuntos de datos muy grandes.
- Por ello, creo que no se ha hecho tanto hincapié en algunas de las prácticas para tratar conjuntos de datos pequeños como sería necesario para abordar problemas en los que no se tiene 100 millones de ejemplos, pero sólo 1000 o incluso menos.
- Así que para mí, el caso interesante es ¿qué pasa si todavía tienes un conjunto de datos pequeño?
- A la derecha, vemos cinco ejemplos con etiquetas limpias y coherentes. En este caso se puede ajustar con bastante seguridad una función a través de los datos y con sólo cinco ejemplos.
- Se puede construir un modelo bastante bueno para predecir la velocidad en función de la tensión de entrada de los sistemas de visión por ordenador entrenados con sólo 30 imágenes y puede funcionar bien.
- La clave suele ser asegurarse de que las etiquetas estén limpias y sean coherentes.

Phone defect example



- Veamos otro ejemplo de inspección de defectos telefónicos,
- Si las instrucciones de etiquetado son poco claras al principio, los etiquetadores etiquetarán las imágenes de forma incoherente.
- Puede haber una zona de ambigüedad en la que diferentes inspectores etiqueten diferentes arañazos con una longitud de entre 0,2 mm y 0,4 mm de forma ligeramente incoherente.
- Así que una solución a esto sería obtener muchas más fotos de teléfonos y arañazos, ver lo que hacen los inspectores, y entonces tal vez podamos entrenar una red neuronal que pueda averiguar a partir de las imágenes qué es y qué no es un arañazo en **promedio**.
- Tal vez ese enfoque podría funcionar, pero sería mucho trabajo y requeriría recopilar muchas imágenes.
- He descubierto que puede ser más fructífero pedir a los inspectores que se sienten y traten de llegar a un acuerdo sobre cuál es el tamaño de la raya que hay que definir como defecto
- Así que, en este ejemplo, si los etiquetadores se ponen de acuerdo en que el punto de transición en el que los pequeños arañazos se convierten en un defecto es una longitud de 0,3 mm, entonces la forma en que etiquetan las imágenes (es decir, el establecimiento de cuadros delimitadores) resulta mucho más coherente.
- Y a los algoritmos de aprendizaje les resulta mucho más fácil tomar imágenes de entrada como ésta y decidir sistemáticamente si algo es un rasguño o un defecto.
- Una mayor consistencia en el etiquetado de imágenes permite a sus algoritmos de aprendizaje alcanzar una mayor precisión, incluso cuando su conjunto de datos no es tan grande.

Big data problems can have small data challenges too

Problems with a large dataset but where there's a long tail of rare events in the input will have small data challenges too.

- Web search
- Self-driving cars ←
- Product recommendation systems ←

- Los problemas de big data también pueden tener desafíos de small data, específicamente los problemas del gran conjunto de datos donde hay una larga cola de eventos raros en la entrada
- Por ejemplo, las grandes empresas de motores de búsqueda en la web tienen todos conjuntos de datos muy grandes de consultas de búsqueda en la web, pero muchas consultas son en realidad muy raras. Por lo tanto, la cantidad de datos de flujo de clics para las consultas poco frecuentes es realmente pequeña
- Por ejemplo, los coches de autoconducción. Las empresas de coches autoconducidos suelen tener conjuntos de datos muy grandes, recopilados a partir de la conducción de cientos de miles o millones de horas o más

- Pero hay sucesos raros que son críticos para asegurarse de que un coche de autoconducción es seguro, como el caso muy raro de un niño pequeño corriendo a través de la carretera, o el caso muy raro de un camión estacionado a través de la carretera.
- Por tanto, aunque un coche de autoconducción tenga un conjunto de datos muy grande, el número de ejemplos que puede tener de estos eventos raros es en realidad muy pequeño. Garantizar la coherencia de las etiquetas en cuanto a la forma de etiquetar estos sucesos raros sigue siendo muy útil para mejorar los coches autoconducidos.
- En el caso de los sistemas de recomendación de productos, si tiene un catálogo de cientos de miles, o millones o más de artículos, tendrá muchos productos en los que el número de ventas de ese artículo es bastante pequeño.
- La cantidad de datos que tienes de los usuarios que interactúan con los artículos de la cola larga es realmente pequeña, por lo que conseguir los datos limpios y consistentes ayudará a tus algoritmos de aprendizaje.
- Por eso, cuando se tiene un conjunto de datos pequeño, la coherencia de las etiquetas es fundamental. Incluso cuando se tiene un gran conjunto de datos, la consistencia de las etiquetas puede ser muy importante.
- Lo que ocurre es que, por término medio, es más fácil conseguir la coherencia de las etiquetas en conjuntos de datos pequeños que en los muy grandes.

Mejorar la coherencia de las etiquetas

Improving label consistency

- Have multiple labelers label same example.
 - When there is disagreement, have MLE, subject matter expert (SME) and/or labelers discuss definition of y to reach agreement.
 - If labelers believe that x doesn't contain enough information, consider changing x .
 - Iterate until it is hard to significantly increase agreement.
- Veamos algunas formas de mejorar la coherencia de sus etiquetas
 - Si le preocupa que las etiquetas sean incoherentes, busque algunos ejemplos y haga que varios etiquetadores etiqueten esos mismos ejemplos.
 - También puedes hacer que el mismo etiquetador etiquete un ejemplo, dejar que se tome un descanso y volver a etiquetarlo para ver si es coherente consigo mismo.
 - Si hay desacuerdos, haz que los responsables del etiquetado discutan juntos lo que creen que debería ser una definición más coherente de la etiqueta y ,
 - Haz que intenten llegar a un acuerdo y , a ser posible, que también lo documenten y lo escriban
 - Esta definición de y puede convertirse en un conjunto actualizado de instrucciones de etiquetado, que pueden utilizar para etiquetar nuevos datos o para reetiquetar datos antiguos.
 - Durante esta discusión, en algunos casos los etiquetadores volverán a decir que no creen que la entrada x tenga suficiente información. Si ese es el caso, considere la posibilidad de cambiar la entrada x . Por ejemplo, cuando algunas fotos de teléfonos pueden ser tan oscuras que ni siquiera podríamos distinguir lo que está pasando,
 - Eso fue una señal de que deberíamos considerar aumentar la iluminación bajo la que se tomaron las fotos. Pero, por supuesto, sé que esto no siempre es posible, pero a veces esto puede ser una gran ayuda.
 - Todo esto es un proceso iterativo, por lo que después de mejorar las instrucciones de etiquetado, se pedirá al equipo que etiquete más datos.
 - Si crees que sigue habiendo desacuerdos, repite todo el proceso de que varios etiquetadores etiqueten el mismo ejemplo, resuelve el desacuerdo y así sucesivamente.

Examples

- Standardize labels

"Um, nearest gas station"

"Umm, nearest gas station"

"Nearest gas station [unintelligible]"



"Um, nearest gas station"

- Merge classes



Deep scratch



Shallow scratch



Scratch

- Un resultado común de este tipo de ejercicio es la estandarización de la definición de las etiquetas.
- Entre estas formas de etiquetar el clip de audio que escuchaste en el video anterior, quizás los etiquetadores se estandaricen en esto como la convención. Así los datos serán más coherentes.
- Otra decisión común que he visto salir de un proceso como este es la fusión de clases.
- Si la definición entre lo que constituye un arañazo profundo y un arañazo superficial no está clara, entonces los etiquetadores terminan etiquetando de forma inconsistente las cosas como arañazos profundos o superficiales.
- A veces, la fábrica tiene que distinguir entre los arañazos profundos y los superficiales.
- Pero a veces he descubierto que no es necesario distinguir entre estas dos clases. Si es así, puede fusionar las dos clases en una sola clase (por ejemplo, toda la etiqueta como la clase de rascado), y esto se deshace de todas las inconsistencias
- La fusión de clases no siempre es aplicable, pero cuando lo es, simplifica la tarea del algoritmo de aprendizaje.

Have a class/label to capture uncertainty

- Defect: 0 or 1



Alternative: 0, Borderline, 1

- Unintelligible audio



"nearest go"

"nearest grocery"

"nearest [unintelligible]"

- Una de las técnicas que he utilizado es crear una nueva clase/etiqueta para capturar la incertidumbre.
- Por ejemplo, digamos que se pide a los etiquetadores que etiqueten los teléfonos como defectuosos o no en función de la longitud del arañazo.
- Tal vez todo el mundo esté de acuerdo en que el arañazo gigante es un defecto, un arañazo minúsculo no es un defecto, pero no se ponen de acuerdo en lo que hay en medio.

- Aquí hay otra opción, que es crear una nueva clase en la que ahora tiene tres etiquetas, en la que los ejemplos ambiguos (genuinamente fronterizos) se colocan en una nueva clase fronteriza.
- Esto ayuda a mejorar potencialmente la coherencia del etiquetado.
- Permítanme usar una ilustración del discurso para ilustrar esto. Dado este clip de audio, realmente no puedo decir lo que dijeron. Si se les obliga a etiquetar, algunos etiquetadores lo transcribirían como "Cerca de ir", mientras que otros quizá lo transcriban como "Tienda de comestibles más cercana".
- Es muy difícil llegar a la coherencia porque el clip de audio es realmente ambiguo.
- Para mejorar la consistencia del etiquetado, puede ser mejor crear una nueva etiqueta llamada etiqueta ininteligible, y simplemente pedir a todos que etiqueten el clip ambiguo como ininteligible.
- Esto puede dar lugar a etiquetas más coherentes que si pidiéramos a todos que adivinaran lo que han oído cuando realmente es ininteligible.

Small data vs. big data (unstructured data)

Small data

- Usually small number of labelers.
- Can ask labelers to discuss specific labels.

Big data

- Get to consistent definition with a small group.
- Then send labeling instructions to labelers.
- Can consider having multiple labelers label every example and using voting or consensus labels to increase accuracy. }

- Permítanme concluir con algunas sugerencias para trabajar con conjuntos de datos pequeños frente a los grandes para mejorar la coherencia de las etiquetas.
- Para los conjuntos de datos pequeños suele haber un número reducido de etiquetadores.
- Por eso, cuando encuentres una incoherencia, puedes pedir a los etiquetadores que se sienten a discutir una imagen o un clip de audio concretos, y tratar de llegar a un acuerdo.
- En el caso de los grandes conjuntos de datos, sería más habitual intentar llegar a definiciones coherentes con un pequeño grupo, y luego enviar las instrucciones de etiquetado a un grupo mayor de etiquetadores.
- Otra técnica que se utiliza comúnmente, pero que en mi opinión se usa en exceso, es que puedes hacer que varios etiquetadores etiqueten cada ejemplo, y luego dejar que voten.
- La votación se denomina a veces **etiquetado de consenso** y se utiliza para aumentar la precisión.
- Me parece que este tipo de técnica de mecanismo de votación, puede funcionar, pero probablemente se utiliza demasiado en el aprendizaje automático hoy en día.
- Lo que he visto hacer a muchos equipos es tener instrucciones de etiquetado incoherentes, y luego tratar de tener muchos etiquetadores y luego realizar votaciones para tratar de hacerlo más coherente.
- Pero antes de recurrir a esto, que sí utilizo pero más bien como último recurso, yo intentaría primero llegar a definiciones de etiquetas más coherentes, para intentar que las elecciones de los etiquetadores individuales sean menos ruidosas en primer lugar. Esto se opone a tomar un montón de datos ruidosos y luego tratar de usar la votación para reducir el ruido.
- Una de las lagunas que veo en el mundo del aprendizaje automático hoy en día es que todavía faltan herramientas para llevar a cabo este tipo de procesos de forma más consistente y repetitiva.

Rendimiento a nivel humano (HLP)

Why measure HLP?

Estimate Bayes error / irreducible error to help with error analysis and prioritization.

Ground Truth Label	Inspector
1	1 ✓
1	0 ✗
1	1 ✓
0	0 ✓
0	0 ✓
0	1 ✗

Human?

99.0%

66.7% accuracy

- Uno de los usos más importantes de la medición del Rendimiento a Nivel Humano o HLP es estimar el error basado o el error irreducible, especialmente en las tareas de datos no estructurados para ayudar a su análisis y priorización y establecer lo que podría ser posible.
- Tomemos como ejemplo las tareas de inspección visual. He recibido solicitudes de propietarios de empresas que me piden que construya un sistema con una precisión del 99% o incluso del 99,9%.
- Una forma de establecer lo que podría ser posible sería tomar un conjunto de datos y mirar los datos de la verdad sobre el terreno.
- Supongamos que tiene seis ejemplos con las etiquetas de la verdad básica (véase la diapositiva), y pide a un inspector humano que etiquete los mismos datos sin conocer las etiquetas de la verdad básica, y ve lo que obtiene
- Si el resultado es el que se ve arriba en la diapositiva, donde el inspector estuvo de acuerdo con la verdad sobre el terreno sólo en cuatro de los seis ejemplos, entonces el HLP es del 66,7%.
- Y esto te permitiría volver al propietario de la empresa y decir: mira, incluso tu inspector sólo obtuvo un 66,7% de precisión. ¿Cómo puede esperar que obtenga un 99% de precisión?
- Así que la HLP es útil para establecer una línea de base en términos de lo que podría ser posible.
- Hay una pregunta que no se hace a menudo, y es ¿qué es exactamente esta etiqueta de la verdad sobre el terreno?
- Deberíamos pensar si realmente estamos midiendo lo que es posible, o sólo estamos midiendo lo bien que dos personas diferentes están de acuerdo entre sí (ya que la propia etiqueta de la verdad básica está determinada por un humano también)

Other uses of HLP

- In academia, establish and beat a respectable benchmark to support publication.
- Business or product owner asks for 99% accuracy. HLP helps establish a more reasonable target.
- "Prove" the ML system is superior to humans doing the job and thus the business or product owner should adopt it.

✗ Use with caution

- En el mundo académico, el HLP se utiliza a menudo como un punto de referencia respetable. Y así, cuando se establece que las personas sólo tienen un 92% de precisión (por ejemplo, en un conjunto de datos de reconocimiento del habla), y si se puede superar el rendimiento a nivel humano, eso ayuda a demostrar que el algoritmo de aprendizaje está haciendo algo difícil
- No digo que esto sea un gran uso de la HLP, pero en el mundo académico, demostrar que se puede superar la HLP tal vez por primera vez ha sido una fórmula probada para establecer la importancia académica de un trabajo, y ayuda a conseguir que se publique algo.
- En la última diapositiva hemos hablado brevemente de lo que hay que hacer si el propietario del producto pide un 99% de precisión y, si cree que eso no es realista, la medición de la HLP puede ayudarle a establecer un objetivo más razonable.
- Tenga cuidado al tratar de demostrar que el sistema de aprendizaje automático es superior al humano en la realización de un determinado trabajo. Aunque puede ser tentador demostrar la superioridad de su algoritmo de aprendizaje, en la práctica, este enfoque rara vez funciona.
- La semana pasada también se vio que las empresas necesitan sistemas que hagan algo más que obtener una buena precisión media en el conjunto de pruebas.
- Le insto a utilizar este tipo de lógica con precaución, o tal vez incluso más preferiblemente, simplemente no utilice estos argumentos. Por lo general, he encontrado otros argumentos que son más eficaces que esto cuando se trabaja con el negocio para ver si deben adoptar un sistema de aprendizaje automático.

The problem with beating HLP as a “proof” of ML “superiority”

"Um... nearest gas station" ← 70% of labels
 "Um, nearest gas station" ← 30%
 Two random labelers agree: $0.7^2 + 0.3^2 = 0.58$
 ML agrees with humans: 0.70 ← +12%

The 12% better performance is not important for anything! This can also mask more significant errors ML may be making.

- El problema de superar el rendimiento a nivel humano como prueba de la superioridad del aprendizaje automático es múltiple. Uno de los problemas de esta métrica es que a veces da a un algoritmo de aprendizaje una ventaja injusta cuando las instrucciones de etiquetado son inconsistentes.
- Por ejemplo, tenemos un clip de audio que dice "gasolinera más cercana". Digamos que el 70% de los etiquetadores utiliza la convención de etiqueta de "Um... gasolinera más cercana" (utilizando la elipsis) y el 30% de los etiquetadores utiliza la convención de etiqueta de "Um, gasolinera más cercana" (utilizando la coma).
- De hecho, ambas transcripciones son completamente válidas, pero por suerte, el 70% de los etiquetadores tienden a elegir la primera y el 30% la segunda.
- Como resultado, la probabilidad de que dos etiquetadores aleatorios coincidan será de $0.7^2 + 0.3^2$, lo que supone un 0,58.
- Significa que, dada la primera convención, hay una probabilidad de 0,7 de que dos etiquetadores etiqueten ambos de la misma manera. Si se utiliza la segunda convención, hay una probabilidad de 0,3 de que ambos etiquetadores aleatorios lo etiqueten de la misma manera.
- En general, la probabilidad de que los etiquetadores estén de acuerdo es de 0,58.
- Y en la forma habitual de medir el Rendimiento de Nivel Humano, concluirá que el Rendimiento de Nivel Humano es 0,58.
- Pero, en realidad, lo que estás midiendo es la posibilidad de que dos etiquetadores aleatorios se pongan de acuerdo.
- Aquí es donde el aprendizaje automático de nuestra sala tiene una ventaja injusta.
- Tras el entrenamiento, el algoritmo de aprendizaje puede utilizar siempre la primera convención de etiquetado (con elipsis) porque sabe que, estadísticamente, tiene un 70% de posibilidades de acertar si utiliza la elipsis en lugar de la coma.

- Así, un algoritmo de aprendizaje coincidirá con los humanos el 70% de las veces, simplemente eligiendo la primera convención de etiquetado.
- Pero esta mejora del 12% en el rendimiento (0,70 frente a 0,58) no es realmente tan importante como para elegir entre dos opciones igualmente buenas y ligeramente arbitrarias.
- El algoritmo de aprendizaje elige sistemáticamente la primera, por lo que obtiene lo que parece una ventaja del 12% en este tipo de consultas, pero en realidad no supera a ningún humano en ningún aspecto que le interese a un usuario.
- Y un efecto secundario de esto es que, si el algoritmo de aprendizaje comete algunos errores más significativos en otros tipos de audio de entrada, entonces las partes en las que realmente se desempeña peor podrían ser promediadas por consultas como estas
- Y esto, por lo tanto, enmascarará u ocultará el hecho de que su algoritmo de aprendizaje está creando en realidad peores transcripciones que las humanas, y creando una falsa impresión de que el sistema de aprendizaje automático lo está haciendo mejor que el HLP.
- En general, podría estar produciendo peores transcripciones que la transcripción humana porque sólo lo hace mejor en este tipo de problema, en el que no es importante hacerlo mejor. En realidad, podría estar haciéndolo peor en otros tipos de audio de entrada.

Aumentar la HLP

Raising HLP

When the ground truth label is externally defined, HLP gives an estimate for Bayes error / irreducible error. *E.g. biopsy*
 But often ground truth is just another human label.

Scratch length (mm)	Ground Truth Label	Inspector
0.7	1	1
0.2	1 0	0
0.5	1	1
0.2	0	0
0.1	0	0
0.1	0	1 0

0.3 mm

66.7%

↓

100%

- He trabajado mucho en imágenes médicas, trabajando en la IA para recuperar el diagnóstico de los rayos X
- Dada la tarea de predecir un diagnóstico a partir de una imagen de rayos X, si el diagnóstico de la verdad básica se define de acuerdo con las pruebas biológicas o médicas (por ejemplo, la biopsia), entonces HLP le ayuda a medir la eficacia de un médico en comparación con un algoritmo de aprendizaje
- Pero cuando la verdad de base la define un ser humano (por ejemplo, una imagen de rayos X etiquetada por un médico), entonces la HLP sólo mide lo bien que un médico puede predecir la etiqueta de otro médico
- Eso también es útil, pero es diferente a si se mide lo bien que se compara su algoritmo con un médico en la predicción de un resultado real de una biopsia médica.
- En resumen, cuando la etiqueta de la verdad básica está definida externamente (por ejemplo, una biopsia médica), la HLP proporciona una estimación del error de base y del error irreducible en términos de predicción del resultado
- Si la verdad de base es otra etiqueta humana (lo que es más frecuente), habrá problemas
- Tomemos por ejemplo el ejemplo de la inspección visual de la pantalla del teléfono que tuvimos antes.
- En lugar de aspirar a superar al inspector humano, puede ser más útil ver por qué la verdad sobre el terreno y el inspector no coinciden.

- Si observamos las longitudes de los diferentes arañazos que etiquetaron (véase la tabla de la diapositiva anterior), si hablamos de los inspectores y hacemos que estén de acuerdo en que 0,3 mm es el umbral a partir del cual un tramo se convierte en un defecto, entonces lo que nos damos cuenta es que para la primera fila, las dos etiquetas de 1 son totalmente adecuadas.
- Para la segunda fila, la verdad de fondo debería ser ahora 0 en lugar de 1 (ya que 0,2mm es menos que el umbral de 0,3mm),
- Si hacemos este ejercicio de conseguir que la etiqueta de la verdad sobre el terreno y este inspector coincidan, entonces acabamos de elevar el rendimiento a nivel humano del 66,7% al 100%, al menos según lo medido en estos seis ejemplos.
- Pero fíjate en lo que hemos hecho, al aumentar la HLP al 100% hemos hecho prácticamente imposible que el algoritmo de aprendizaje supere la HLP,
- Puede parecer terrible que ya no puedas decirle al empresario que tu sistema puede superar a HLP, pero el beneficio de esto es que ahora tienes datos mucho más limpios y consistentes, y eso, en última instancia, permitirá a tu algoritmo de aprendizaje hacerlo mejor.
- El objetivo es conseguir un algoritmo de aprendizaje que realmente genere predicciones precisas, en lugar de limitarse a demostrar que se puede superar la HLP

Raising HLP

- When the label y comes from a human label, $HLP \ll 100\%$ may indicate ambiguous labeling instructions. *Um, Um...*
 - Improving label consistency will raise HLP.
 - This makes it harder for ML to beat HLP. But the more consistent labels will raise ML performance, which is ultimately likely to benefit the actual application performance.
- En resumen, cuando la etiqueta de verdad y procede de un humano, que la HLP sea bastante inferior al 100% puede indicar simplemente que las instrucciones o la convención de etiquetado son ambiguas.
 - Anteriormente vimos un ejemplo de esto en la inspección visual.
 - Este tipo de convención de etiquetado ambiguo también hará que el HLP sea inferior al 100%.
 - La mejora de la consistencia de las etiquetas ayudará entonces a aumentar el rendimiento a nivel humano, haciendo más difícil (por desgracia) que su algoritmo de aprendizaje supere a HLP
 - Pero, en general, un etiquetado más coherente aumentará el rendimiento de su algoritmo de aprendizaje automático, lo que en última instancia probablemente beneficiará a la aplicación real.

HLP on structured data

Structured data problems are less likely to involve human labelers, thus HLP is less frequently used.

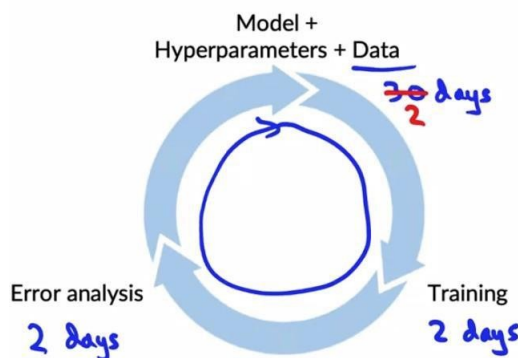
Some exceptions:

- User ID merging: Same person?
- Based on network traffic, is the computer hacked?
- Is the transaction fraudulent?
- Spam account? Bot?
- From GPS, what is the mode of transportation – on foot, bike, car, bus?

- Hasta ahora hemos hablado de HLP en los datos no estructurados, pero algunas de estas cuestiones se aplican también a los datos estructurados.
- Ya se sabe que los problemas de datos estructurados tienen menos probabilidades de implicar a los etiquetadores humanos y, por tanto, la HLP se utiliza con menos frecuencia.
- Pero hay excepciones. Ya has visto el ejemplo de la identificación de usuario emergente, en el que podrías tener un etiquetador humano para identificar si dos registros pertenecen a la misma persona.
- He trabajado en proyectos en los que examinamos el tráfico de red de un ordenador para intentar averiguar si ha sido pirateado, y tenemos expertos en informática que nos proporcionan etiquetas.
- A veces es difícil saber si una transacción es fraudulenta y sólo hay que pedir a un humano que lo etiquete. ¿O se trata de una cuenta de spam o de una cuenta generada por un bot?
- O a partir del GPS, cuál es el modo de transporte, si esta persona va a pie, o en bicicleta, o en el coche, o en el autobús. Resulta que los autobuses paran en las paradas, así que se puede saber si alguien está en un autobús o en un coche basándose en el rastro del GPS.
- Para problemas como estos, sería bastante razonable pedir a un humano que etiquete los datos, al menos en la primera pasada para que un algoritmo de aprendizaje haga esas predicciones.
- Cuando la etiqueta de la verdad sobre el terreno que se intenta predecir proviene de un humano, se aplicarán las mismas preguntas sobre el significado de la HLP.
- Es una línea de base útil para averiguar lo que es posible, pero a veces cuando se mide la HLP, uno se da cuenta de que la baja HLP proviene de etiquetas inconsistentes, y trabajar para mejorar la HLP mediante la creación de un estándar de etiquetado más consistente aumentará la HLP y le dará datos más limpios con los que mejorar el rendimiento de su algoritmo de aprendizaje.
- Esto es lo que espero que saques de este vídeo. En primer lugar, la HLP es importante para los problemas en los que el rendimiento a nivel humano puede proporcionar una referencia útil. Yo lo mido y lo utilizo como referencia de lo que podría ser posible y para impulsar el análisis y la priorización.
- Dicho esto, cuando mida el HLP, si encuentra que el HLP es mucho menor que el 100%, pregúntese también si algunas de las lagunas en el HLP se deben a instrucciones de etiquetado incoherentes.
- Porque si resulta ser así, la mejora de la coherencia del etiquetado aumentará
- HLP y también proporciona datos más limpios para su algoritmo de aprendizaje, lo que en última instancia resultará en un mejor rendimiento del algoritmo de aprendizaje automático.

Obtención de datos

How long should you spend obtaining data?



- Get into this iteration loop as quickly as possible.
- Instead of asking: How long it would take to obtain m examples?
Ask: How much data can we obtain in k days.
- Exception: If you have worked on the problem before and from experience you know you need m examples.

- Una pregunta clave que hay que plantearse es cuánto tiempo hay que dedicar a la obtención de datos.
- Usted sabe que el aprendizaje automático es un proceso altamente iterativo en el que hay que elegir un modelo, unos hiperparámetros, tener un conjunto de datos, realizar un entrenamiento, llevar a cabo un análisis de errores y, a continuación, dar varias vueltas a este bucle para llegar a un buen modelo.

- Digamos que entrenar el modelo por primera vez lleva un par de días, y realizar el análisis de errores por primera vez puede llevar un par de días.
- Si este es el caso, le instaría a no pasar **30** días recopilando datos, porque eso retrasará un mes entero antes de entrar en esta iteración.
- En su lugar, le insto a entrar en este bucle de iteración lo antes posible.
- Le insto a que se pregunte qué pasaría si se diera sólo dos días para recopilar datos. ¿Te ayudaría eso a entrar en este bucle mucho más rápidamente?
- Tal vez dos días sea demasiado poco, pero he visto a demasiados equipos que tardan demasiado en recopilar su conjunto de datos iniciales antes de entrenar el modelo inicial.
- Una vez entrenado el modelo inicial y realizado el análisis de errores, hay mucho tiempo para volver a recoger más datos.
- En muchos de los proyectos que he dirigido, me he dado cuenta de que cuando me dirijo al equipo y digo: "Oíd todos, vamos a pasar como máximo siete días recogiendo datos, ¿qué podemos hacer?". Descubrí que plantear la pregunta de esa manera a menudo conduce a formas mucho más creativas de obtener datos
- Eso le permite entrar en este bucle de iteración más rápidamente y dejar que el proyecto avance más rápido
- Una excepción a esta directriz es si ha trabajado en este problema antes, y por experiencia sabe que necesita al menos un cierto tamaño de conjunto de entrenamiento. Si es así, puede estar bien invertir más esfuerzo por adelantado para recoger esa cantidad de datos.
- Como he trabajado en el reconocimiento de voz, tengo una buena idea de cuántos datos necesitaré para hacer ciertas cosas y sé que no vale la pena intentar entrenar ciertos modelos si tengo menos de cierto número de horas de datos.
- Pero muchas veces, si estás trabajando en un problema nuevo y si no estás seguro (y a menudo es difícil saberlo incluso por la literatura), es mucho mejor recoger rápidamente una pequeña cantidad de datos, entrenar un modelo y utilizar el análisis de errores para decirte si vale la pena recoger más datos.

Inventory data

Brainstorm list of data sources ( speech recognition)





Source	Amount	Cost	Time	
Owned	100h	\$0	0	✓
Crowdsourced – Reading	1000h	\$10000	14 days	
Pay for labels	100h	\$6000	7 days	
Purchase data	1000h	\$10000	1 day	✓

Other factors: Data quality, privacy, regulatory constraints

- En cuanto a la obtención de los datos, un paso que suelo llevar a cabo es hacer un inventario de las posibles fuentes de datos.
- Sigamos utilizando el reconocimiento de voz como ejemplo. Si tuvieras que hacer una lluvia de ideas con una lista de fuentes de datos, esto es lo que se te ocurriría.
- Tal vez ya posea 100 horas de datos de habla transcritos, y como ya los posee, el coste de eso es cero.
- También puedes utilizar una plataforma de crowdsourcing y pagar a personas para que lean un texto. Les proporcionas un texto y les pides que lo lean en voz alta. Esto crea los datos de texto que necesitas, ya que tienes la transcripción y ellos leen a partir de ella.
- O puede decidir tomar el audio que tiene y que aún no ha sido etiquetado, y pagar para que lo transcriban. Resulta que esto es más caro por hora que pagar a personas para que lean textos, pero el resultado es un audio que suena más natural porque la gente no lo está leyendo

- O puede encontrar algunas organizaciones comerciales que podrían venderle datos.
- A través de un ejercicio como éste, puedes hacer una lluvia de ideas sobre los diferentes tipos de datos que podrías utilizar, así como sus costes asociados.
- Una columna que falta, y que me parece muy importante, es la de los costes de tiempo, es decir, cuánto tiempo se tarda en obtener estos diferentes tipos de datos.
- Para los datos en propiedad se podría conseguir de forma instantánea. En el caso de la lectura colectiva, es posible que tengas que implementar un montón de software, encontrar la plataforma de crowdsourcing adecuada y llevar a cabo la integración del software, por lo que podrías estimar que eso son dos semanas de trabajo de ingeniería.
- Pagar para que los datos sean etiquetados es más sencillo, pero sigue siendo necesario organizarlos y gestionarlos, mientras que comprarlos puede ser mucho más rápido.
- Me parece que algunos equipos no pasan por un proceso de inventario como este y se limitan a elegir una idea al azar. Pero si te sientas, escribes todas las fuentes de datos y piensas en las ventajas y desventajas, incluidos los costes y el tiempo, eso te ayudará a tomar mejores decisiones.
- Si está especialmente presionado por el tiempo, basándose en este análisis, puede decidir
- utilizar los datos que ya posee y tal vez comprar algunos de ellos
- Además de la cantidad de datos que se pueden adquirir, y de los costes financieros y de tiempo, otros factores importantes que dependen de la aplicación serán la calidad de los datos, la privacidad y las restricciones normativas.

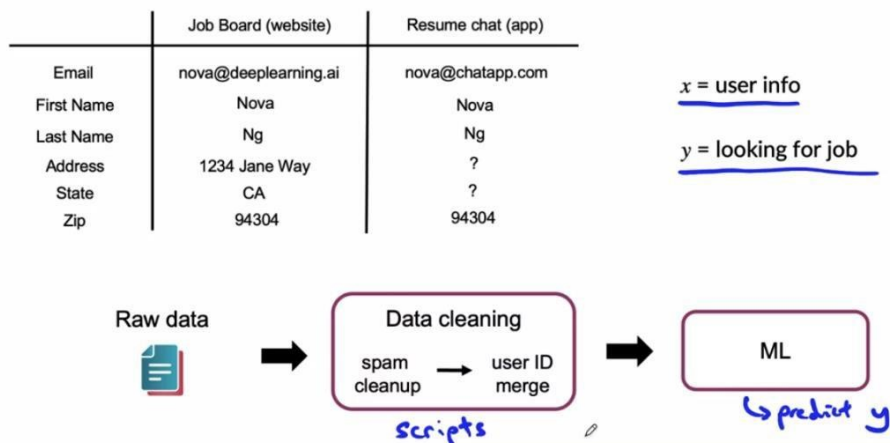
Labeling data

- Options: In-house vs. outsourced vs. crowdsourced
 - Having MLEs label data is expensive. But doing this for just a few days is usually fine.
 - Who is qualified to label? 
 -  Speech recognition – any reasonably fluent speaker
 -  Factory inspection, medical image diagnosis – SME (subject matter expert)
 -  Recommender systems – maybe impossible to label well
 - Don't increase data by more than 10x at a time
-
- Las tres formas más comunes de etiquetar los datos son: **en la empresa**, donde tu propio equipo etiqueta los datos, frente a la **subcontratación**, donde puedes encontrar alguna empresa que etiquete los datos y hacer que lo hagan por ti, frente al **crowdsourcing**, donde puedes utilizar una plataforma de crowdsourcing para que un gran grupo etiquete los datos de forma colectiva.
 - La diferencia entre la externalización y el crowdsourced es que, dependiendo del tipo de datos que tengas, puede haber empresas especializadas que podrían ayudarte a conseguir la etiqueta de forma bastante eficiente.
 - Tener ingenieros de aprendizaje automático que etiqueten los datos suele ser caro, pero creo que para poner en marcha un proyecto rápidamente, tener ingenieros de aprendizaje automático que etiqueten sólo durante unos días suele estar bien
 - De hecho, esto puede ayudar a construir la intuición de los ingenieros de aprendizaje automático sobre los datos.
 - Cuando estoy trabajando en un nuevo proyecto, **a menudo no me importa pasar unas horas, o tal vez uno o dos días, etiquetando datos yo mismo, si eso me ayuda a construir mi intuición sobre el proyecto.**
 - Pero más allá de cierto punto, es posible que no quiera dedicar todo su tiempo como ingeniero de aprendizaje automático a etiquetar datos, y que quiera cambiar a un proceso de etiquetado más escalable.
 - Dependiendo de su aplicación, también puede haber diferentes grupos o subgrupos de individuos que van a estar más cualificados para proporcionar las etiquetas
 - Si estás trabajando en el reconocimiento de voz, quizá casi cualquier hablante razonablemente fluido pueda escuchar el audio y transcribirlo.
 - En el caso de aplicaciones más especializadas, como la inspección de fábricas o el diagnóstico por imágenes médicas, es probable que una persona normal y corriente no pueda mirar una imagen médica de rayos X y diagnosticar a partir de ella, o mirar un smartphone y determinar qué es un defecto. Tareas más especializadas como estas suelen requerir la intervención de una PYME o un experto en la materia, a fin de proporcionar etiquetas precisas.

- Por último, hay algunas aplicaciones que son muy difíciles de conseguir que alguien dé buenas etiquetas. Por ejemplo, las recomendaciones de productos, probablemente hay sistemas de recomendación de productos que te dan mejores recomendaciones que incluso tus mejores amigos o quizás tu pareja.
- Para ello, es posible que tenga que basarse en los datos de compra por parte del usuario como etiqueta en lugar de hacer que los humanos lo etiqueten.
- Cuando trabaje en una aplicación, averiguar en cuál de estas categorías de aplicación está trabajando e identificar el tipo de persona adecuada para ayudarlo a etiquetar, será un paso importante para asegurarse de que sus etiquetas son de alta calidad.
- Un último consejo. Supongamos que tienes 1.000 ejemplos y has decidido que necesitas un conjunto de datos más grande. ¿Cuánto más grande debe hacer su conjunto de datos?
- Un consejo que he dado a muchos equipos es que **no aumenten sus datos más de 10 veces a la vez**.
- Si ha entrenado su modelo con 1.000 ejemplos, quizá merezca la pena invertir para intentar aumentar su conjunto de datos a 3.000 ejemplos o, como máximo, a 10.000.
- Pero yo primero haría un aumento de menos de 10 veces, entrenaría otro modelo, llevaría a cabo un análisis de errores y sólo entonces averiguaría si merece la pena aumentar sustancialmente más allá de eso.
- Porque una vez que se aumenta el tamaño del conjunto de datos en 10 veces, muchas cosas cambian, y se hace realmente difícil predecir lo que sucederá cuando el tamaño del conjunto de datos aumente incluso más allá de eso.
- Se espera que esta guía ayude a los equipos a evitar invertir demasiado en toneladas de datos, sólo para darse cuenta de que recoger tantos datos no era lo más útil

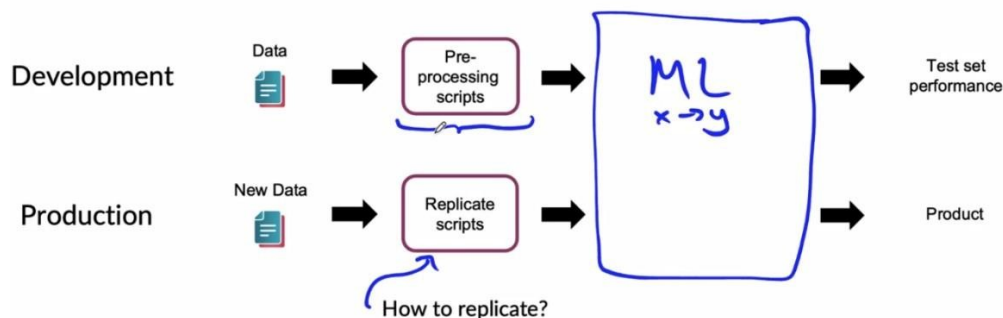
Canalización de datos

Data pipeline example



- Supongamos que, dada la información de los usuarios, se quiere predecir si un usuario determinado está buscando trabajo y, a continuación, mostrarle anuncios de empleo u otra información útil.
- Cuando los datos están en bruto, como los de arriba, a menudo se realiza un preprocesamiento o una limpieza de los datos antes de introducirlos en el algoritmo de aprendizaje.
- La limpieza de datos puede incluir cosas como la limpieza de spam, como la eliminación de las cuentas de spam, y tal vez también la fusión de ID de usuario, de la que hablamos en un vídeo anterior.
- Digamos que estas limpiezas de datos se hacen sólo con scripts (secuencias de instrucciones explícitas). También se podría hacer con algoritmos de aprendizaje automático, pero hace que su gestión sea un poco más compleja.

Data pipeline example



- Cuando se tienen scripts para la limpieza de datos, uno de los problemas que se plantean es la replicabilidad cuando se llevan estos sistemas a la implantación en producción.
- Durante la fase de desarrollo, habrás visto que los scripts de preprocesamiento pueden ser bastante desordenados. Puede ser que hackees algo, que proceses datos, que envíes un archivo a otro miembro de tu equipo, que tengas unos cuantos conjuros en Python, etc.
- La pregunta clave es: si el preprocesamiento se hizo con un montón de scripts, repartidos en un montón de ordenadores de diferentes personas, ¿cómo replicar los scripts para asegurarse de que la distribución de entrada a un algoritmo de aprendizaje automático era la misma para los datos de desarrollo y los datos de producción?
- Creo que la cantidad de esfuerzo que se debe invertir para asegurarse de que los scripts de preprocesamiento son altamente replicables depende de la fase del proyecto.

POC and Production phases

POC (proof-of-concept):

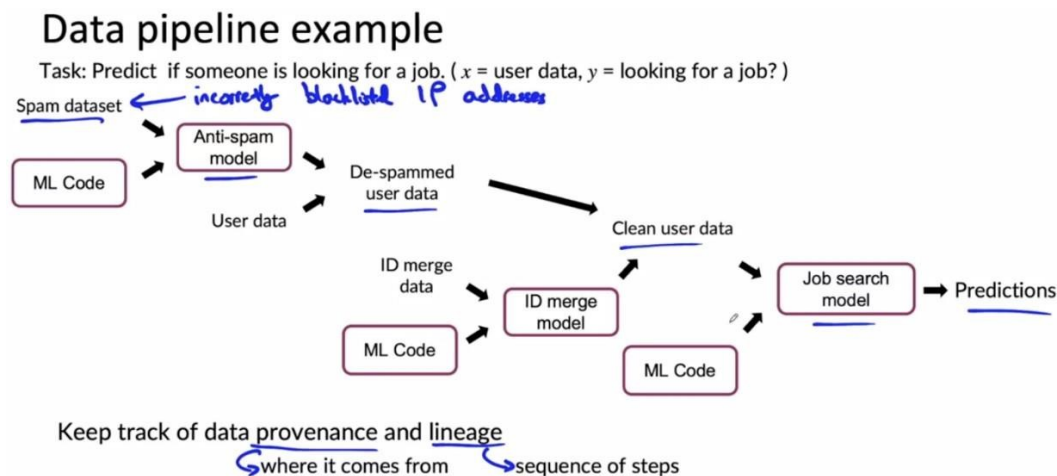
- Goal is to decide if the application is workable and worth deploying.
- Focus on getting the prototype to work!
- It's ok if data pre-processing is manual. But take extensive notes/comments.

Production phase:

- After project utility is established, use more sophisticated tools to make sure the data pipeline is replicable.
- E.g., TensorFlow Transform, Apache Beam, Airflow,....

- Muchos proyectos pasan por una fase de prueba de concepto o POC, y luego por una fase de producción
- Durante la fase de prueba de concepto, el objetivo principal es simplemente decidir si la aplicación es viable y merece la pena construirla y desplegarla.
- Mi consejo para la mayoría de los equipos es que durante la fase POC se centren en hacer funcionar el prototipo.
- No pasa nada si parte del preprocesamiento de datos es manual. Si el proyecto tiene éxito, puedes replicar todo este preprocesamiento más adelante.
- Mi consejo es que tomes muchas notas y escribas muchos comentarios para aumentar las probabilidades de que puedas replicar todo este preprocesamiento más adelante, pero tampoco es el momento de enfrascarse en toneladas de procesos sólo para asegurar la replicabilidad
- El objetivo es decidir si la solicitud es viable y merece la pena pasar a la siguiente fase.
- Una vez que hayas decidido que vale la pena llevar este proyecto a producción, entonces sabes que va a ser muy importante hacer la réplica de los scripts de preprocesamiento.
- En esta fase, es cuando utilizaría herramientas más sofisticadas para asegurarme de que todo el pipeline de datos es replicable. Herramientas como TensorFlow Transform, Apache Beam y Airflow se vuelven así muy valiosas.

Metadatos, procedencia de los datos y linaje



- He aquí un ejemplo más complejo de canalización de datos. Supongamos que se empieza con un conjunto de datos de spam. Esto puede incluir una lista de cuentas de spam conocidas, así como características de las direcciones IP de la lista negra
- A continuación, se implementa un algoritmo para la detección de spam.
- A continuación, se toman los datos de los usuarios y se aplica el modelo antispam para obtener los datos de los usuarios desespamados.
- Ahora, tomando los datos de tu usuario desespamado, puedes llevar a cabo la fusión de ID de usuario.
- Para ello, podría empezar con algunos datos de identificación combinados. Se trataría de datos etiquetados que le indican algunos pares de cuentas que realmente corresponden a la misma persona
- A continuación, tenemos una implementación del algoritmo de aprendizaje automático, entrenamos el modelo en eso, y esto le da un modelo de fusión de ID aprendido
- Lo aplicamos a los datos de los usuarios desespamados, y eso nos da los datos de los usuarios limpios.
- Por último, basándose en los datos limpios de los usuarios, tendrá otro modelo de aprendizaje automático para predecir si un determinado usuario está buscando trabajo o no.
- He visto tuberías de datos o cascadas de datos que son incluso mucho más complicadas que esto.
- Uno de los retos de trabajar con tuberías de datos como esta es, qué pasa si después de ejecutar este sistema durante meses, descubres que las listas negras de direcciones IP que estás utilizando tienen algunos errores (por ejemplo, direcciones IP incorrectamente incluidas en la lista negra porque varios usuarios la utilizan, como en un campus universitario)
- La cuestión es que, una vez construido este gran y complejo sistema, si se actualiza el conjunto de datos sobre el spam y todas las diferentes partes de la cadena de producción, ¿cómo se soluciona el problema? ¿Cómo se puede volver atrás y solucionar el problema?
- Esto es especialmente difícil si cada uno de estos sistemas fue desarrollado por diferentes ingenieros, y usted tiene archivos repartidos entre los ordenadores portátiles de su equipo de desarrollo de ingeniería de aprendizaje automático.
- Para asegurarse de que su sistema es mantenible, puede ser muy útil hacer un seguimiento de la procedencia de los datos, así como del linaje.
- La procedencia de los datos se refiere a su origen, por ejemplo, ¿a quién le compró la dirección IP de spam?
- El linaje se refiere a la secuencia de pasos necesarios para llegar al final de la tubería.
- Como mínimo, disponer de una amplia documentación podría ayudarle a reconstruir la procedencia y el linaje de los datos, lo que ayudaría a construir sistemas robustos y mantenibles en la fase de producción.
- Para ser honesto, las herramientas para hacer un seguimiento de la procedencia y el linaje de los datos son todavía inmaduras en este mundo del aprendizaje automático. Me parece que una amplia documentación puede ayudar y algunas herramientas formales como TensorFlow Transform también pueden ayudar, pero resolver este tipo de problemas todavía no es algo en lo que seamos grandes como comunidad.

Meta-data

Examples:



Manufacturing visual inspection: Time, factory, line #, camera settings, phone model, inspector ID,....



Speech recognition: Device type, labeler ID, VAD model ID,....

Useful for:

- Error analysis. Spotting unexpected effects.
- Keeping track of data provenance.

- Los metadatos son datos sobre datos.
- Por ejemplo, en la inspección visual de la fabricación, los datos serían fotos de teléfonos y sus etiquetas, y los metadatos te dicen a qué hora se tomó la foto, de qué fábrica era esta foto, cuál es el número de línea, cuáles fueron los ajustes de la cámara, como el tiempo de exposición y la apertura de la cámara, cuál es el ID del inspector que proporcionó esta etiqueta, etc.
- Este tipo de metadatos puede resultar muy útil porque si durante el desarrollo del aprendizaje automático descubres que, por alguna extraña razón, algunas muestras están produciendo muchos más errores, esto te permite volver a la fuente para investigarla durante el análisis de errores
- Muchas veces, cuando almaceno los metadatos adecuados, esos metadatos ayudan a generar una idea clave que ayuda a que el proyecto avance.
- Mi consejo es que si tienes un framework o un conjunto de herramientas MLOps para almacenar metadatos, definitivamente te hará la vida más fácil, al igual que rara vez te arrepientes de comentar tu código
- Del mismo modo, si no se almacenan los metadatos en el momento oportuno, puede ser mucho más difícil volver a recuperar y organizar esos datos.
- Un ejemplo más de reconocimiento de voz.
- Si tienes grabaciones de audio de diferentes marcas de smartphones, el modelo de detección de actividad vocal utilizado puede variar. Si hay algún error en alguno de estos modelos de DVA, disponer de metadatos aumenta significativamente las probabilidades de que descubras los errores y los utilices para mejorar el rendimiento de tu algoritmo.
- En resumen, los metadatos pueden ser muy útiles para el análisis de errores y la detección de efectos inesperados, categorías de datos que tienen un rendimiento inusualmente bajo
- Como parte de la creación de los sistemas, considere la posibilidad de realizar un seguimiento de los metadatos, que puede ayudarle a rastrear la procedencia de los datos, pero también a analizar los errores.

Equilibrio entre el entrenamiento, el desarrollo y la prueba

Balanced train/dev/test splits in small data problems



Visual inspection example: 100 examples, 30 positive (defective)

Train/dev/test: 60%/20%/20%

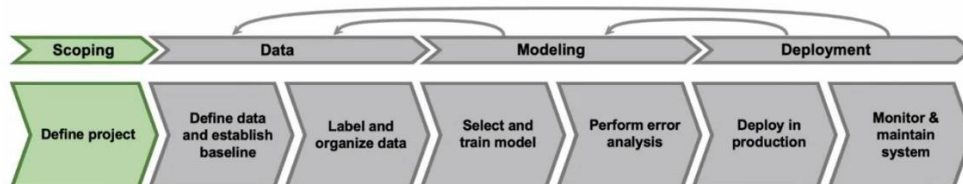
Random split: 21/2/7 positive example
35% 10% 35%

Want: 18/6/6
30%/30%/30% } balanced split

No need to worry about this with large datasets - a random split will be representative.

- Muchos de nosotros estamos acostumbrados a tomar un conjunto de datos y dividirlo aleatoriamente en conjuntos de entrenamiento, desarrollo y prueba.
- Resulta que cuando el conjunto de datos es pequeño, tener un conjunto equilibrado de entrenamiento, desarrollo y prueba puede mejorar significativamente el proceso de desarrollo del aprendizaje automático.
- Utilicemos nuestro ejemplo de inspección visual de fabricación. Digamos que su conjunto de entrenamiento tiene 100 imágenes (un conjunto de datos bastante pequeño) y con 30 ejemplos positivos, es decir, 30 teléfonos defectuosos y 70 no defectuosos.
- Si se utilizara una división del 60% de los datos en el conjunto de entrenamiento, el 20% en el conjunto de desarrollo (o validación) y el 20% en el conjunto de prueba, entonces, por casualidad, en la división aleatoria, podría terminar con 21 ejemplos positivos en el entrenamiento, 2 en el desarrollo y 7 en la prueba.
- Esto sería bastante probable sólo por el azar.
- Y esto significa que el conjunto de entrenamiento es 35% positivo, no tan lejos del 30% positivo en el conjunto de datos, pero su conjunto de desarrollo es sólo 10% positivo y su conjunto de prueba es 35% positivo.
- Y esto hace que su conjunto de desarrollo no sea representativo, porque sólo tiene 2 (o 10%) ejemplos positivos en lugar de un 30% de ejemplos positivos.
- Así que lo que realmente queremos es que el conjunto de entrenamiento tenga exactamente 18 ejemplos positivos, que el conjunto de desarrollo tenga exactamente 6 ejemplos positivos y que el conjunto de prueba tenga exactamente 6 ejemplos positivos. Y esto sería 30%, 30% 30%.
- Y si pudieras conseguir este tipo de división, esto se llamaría una **división equilibrada**, donde cada uno de tus entrenamientos, desarrollos y pruebas tiene exactamente un 30% de ejemplos positivos, y esto hace que tu conjunto de datos sea más representativo de la verdadera distribución de datos.
- No hay que preocuparse por este efecto cuando se tiene un gran conjunto de datos, ya que una división aleatoria será muy probablemente representativa, lo que significa que el porcentaje de ejemplos positivos será bastante cercano a su conjunto de datos.
- Esta es una de esas pequeñas técnicas que resulta marcar una gran diferencia en el rendimiento cuando se trabaja en un pequeño problema de datos

Qué es el alcance



Scoping example: Ecommerce retailer looking to increase sales

- Better recommender system
- Better search
- Improve catalog data
- Inventory management
- Price optimization

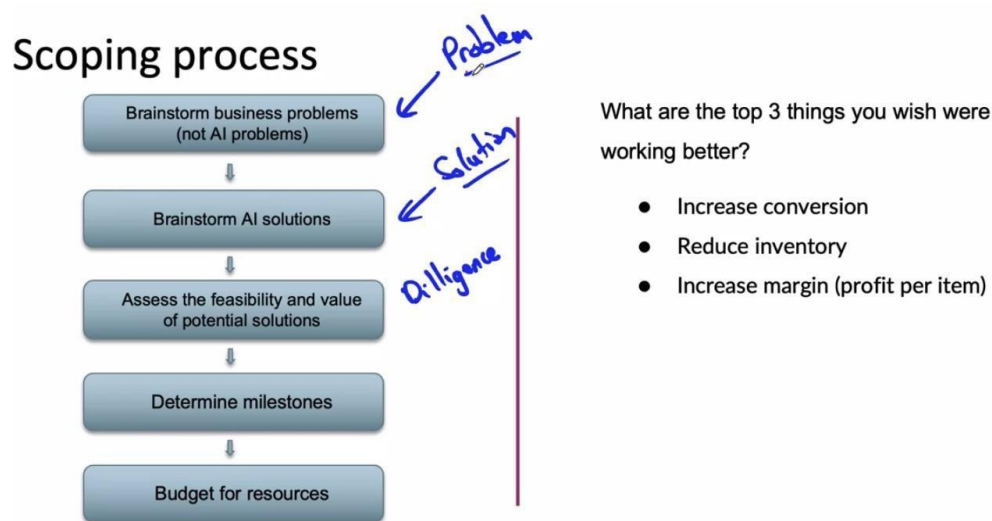
Questions:

- What project should we work on?
- What are the metrics for success?
- What are the resources (data, time, people) needed?

- Elegir el proyecto adecuado en el que trabajar es una de las habilidades más raras y valiosas de la IA hoy en día.
- Recuerdo que cuando era más joven, tendía a lanzarme al primer proyecto que me entusiasmaba y a veces tenía suerte y salía bien.
- Ahora que tengo un poco más de experiencia, creo que merece la pena dedicar mucho tiempo a pensar en algunas opciones y seleccionar el proyecto más prometedor en el que trabajar antes de dedicarle tanto esfuerzo. Así que vamos a sumergirnos en el alcance.
- Utilicemos el ejemplo de un minorista de comercio electrónico que quiere aumentar sus ventas. Si se sentara a hacer una lluvia de ideas, se le ocurrirían muchas, como por ejemplo un mejor sistema de recomendación de productos, o una mejor búsqueda para que la gente pueda encontrar lo que busca, o tal vez quiera mejorar la calidad de los datos del catálogo

- Incluso puede ayudarles con la gestión del inventario, como decidir cuántas camisetas comprar, dónde enviarlas o qué precio optimizar.
- Con una rápida sesión de brainstorming, es posible que se le ocurran docenas de ideas para ayudar a este minorista de comercio electrónico.
- Las preguntas a las que nos gusta responder con el alcance son qué proyecto o proyectos debemos trabajar.
- También queremos saber cuáles son las métricas de éxito y cuáles son los recursos necesarios para ejecutar este proyecto.
- Lo que he visto en muchos negocios es que de todas las ideas que puedes trabajar, algunas van a ser mucho más valiosas que otras.
- Ser capaz de elegir el proyecto más valioso aumentará significativamente el impacto de su trabajo



Proceso de evaluación



- Lo primero que hago es reunirme con el propietario de una empresa o particular (alguien que entienda el negocio) y hacer una lluvia de ideas con ellos. ¿Cuáles son sus problemas de negocio o de aplicación?
- Y en esta etapa estoy tratando de identificar un problema de negocios, no un problema de IA.
- Podría preguntar por las tres cosas principales que desearía que funcionaran mejor. Podría tratarse de problemas empresariales como el aumento de las conversiones, la reducción del inventario, el aumento de los beneficios por artículo vendido, etc.
- En este punto del proceso, no trato de identificar un problema de IA. De hecho, a menudo les digo a mis socios que no quiero oír hablar de sus problemas de IA. Quiero oír hablar de sus problemas de negocio.
- Mi trabajo es trabajar con usted para ver si hay una solución de IA. A veces no la hay y eso está bien. Siéntase libre de utilizar las mismas palabras cuando haga una lluvia de ideas con sus socios que no son de la IA.
- Una vez que se han identificado algunos problemas empresariales, sólo entonces empiezo a hacer una lluvia de ideas sobre si hay posibles soluciones de IA. No todos los problemas pueden ser resueltos por la IA y eso está bien.
- Es útil separar la identificación del problema, de la identificación de la solución
- Como ingenieros, somos bastante buenos en la búsqueda de soluciones, pero tener una articulación clara de cuál es el problema primero a menudo nos ayuda a encontrar mejores soluciones.
- Tras una lluvia de ideas sobre diferentes soluciones, evaluaría la viabilidad y el valor de las mismas.
- A veces me oyes utilizar la palabra diligencia. Diligencia es un término que en realidad procede del ámbito jurídico, pero básicamente significa volver a comprobar si una solución de IA es realmente viable y valiosa desde el punto de vista técnico
- Si sigue pareciendo prometedor, entonces concretamos los hitos del proyecto y presupuestamos los recursos

Separating problem identification from solution

Problem	Solution
Increase conversion	Search, recommendations
Reduce inventory	Demand prediction, marketing
Increase margin (profit per item)	Optimizing what to sell (e.g., merchandising), recommend bundles


 What to achieve
 
 How to achieve

- Profundicemos en este proceso de identificación de problemas y soluciones
- Así que el primero es el aumento de la conversión
- Puede tener diferentes ideas al respecto. Por ejemplo, puede querer mejorar la calidad de los resultados de búsqueda del sitio web, para que la gente encuentre productos más relevantes cuando los busque. O tal vez decidas intentar ofrecer mejores recomendaciones de productos en función de su historial de compras.
- Es bastante común que un problema pueda dar lugar a múltiples ideas de solución.
- También puede aportar otras ideas, como por ejemplo un rediseño de la forma en que se muestran los productos en la página.
- O puede encontrar formas interesantes de sacar a la luz las reseñas de productos más relevantes, para ayudar a los usuarios a entender el producto y, con suerte, a comprarlo.
- Tomemos el siguiente problema de reducción de inventario. Puedes imaginar que haces un proyecto de predicción de la demanda para estimar mejor cuántas personas te comprarán algo.
- Esto le ayudará a mantener un inventario más preciso en sus almacenes.
- O bien, puede decidir realizar una campaña de marketing para impulsar las ventas de los productos de los que ha comprado demasiados, con el fin de reducir el inventario
- Podría haber numerosas ideas para solucionar el problema del aumento del margen.
- Puede que se le ocurran algunas formas de utilizar el aprendizaje automático para optimizar lo que se vende.
- En el comercio minorista, a veces esto se llama merchandising, es decir, decidir qué vender.
- Puedes recomendar paquetes en los que si alguien compra una cámara, o puedes recomendarle una funda protectora para la cámara para aumentar el margen.
- La identificación del problema es un paso en el que hay que pensar qué cosas se quieren conseguir.
- La identificación de la solución es un proceso de reflexión sobre cómo alcanzar esos objetivos.
- Una cosa que veo que hacen demasiados equipos hoy en día es lanzarse al primer proyecto que les entusiasma
- Creo que vale la pena empezar con el pensamiento divergente, en el que se hace una lluvia de ideas sobre muchas posibilidades. A continuación, hay que seguir con el pensamiento convergente, en el que se reduce a uno o un pequeño puñado de los proyectos más prometedores en los que centrarse.

Diligencia sobre la viabilidad y el valor

Feasibility: Is this project technically feasible?

Use external benchmark (literature, other company, competitor)

	Unstructured (e.g., speech, images)	Structured (e.g., transaction records)
New	HLP	Predictive features available?
Existing	HLP History of project	New predictive features? History of project

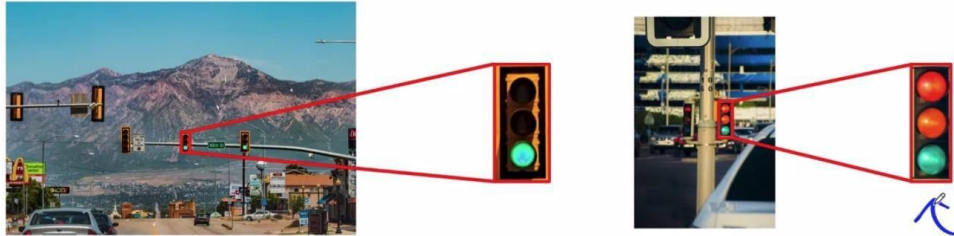
HLP: Can a human, given the same data, perform the task?

- Empecemos por averiguar si la idea del proyecto es técnicamente viable
- Una manera de tener una idea rápida de la viabilidad es utilizar un punto de referencia externo, como la investigación existente, la literatura u otras formas de publicaciones
- Puede tratarse incluso de información de otros competidores o de otras empresas
- Eso podría ayudarte a tener una idea de que este proyecto puede ser técnicamente factible, porque alguien más ha logrado hacer algo similar.
- Aquí hay otras formas de evaluar la viabilidad también, utilizando una matriz de dos por dos. En un eje, vemos si nuestro problema tiene datos no estructurados, como imágenes, o datos estructurados, como registros de transacciones.
- En el otro eje, voy a poner nuevo frente a existente. Nuevo significa que estás entregando una capacidad totalmente nueva, mientras que existente significa que estás preparando el proyecto para mejorar una capacidad existente.
- En el cuadrante superior izquierdo (nuevo no estructurado), HLP para ser muy útil para darle una idea inicial de si un proyecto es factible.
- A la hora de evaluar la HLP, daría a un humano para que evaluara los mismos datos que se le darían a un algoritmo de aprendizaje
- Por ejemplo, dadas las imágenes de smartphones rayados, ¿puede un humano realizar la tarea de detectar los arañazos de forma fiable?
- Si un humano puede hacerlo, aumenta la esperanza de que también podamos conseguir que un algoritmo de aprendizaje lo haga.
- Para los proyectos existentes, yo también utilizaría HLP como referencia. Si los humanos pueden alcanzar el nivel al que esperas llegar, eso podría darte más esperanzas de que es técnicamente factible.
- Mientras que si se espera aumentar el rendimiento mucho más allá del nivel humano, entonces sugiere que el proyecto podría ser más difícil, o podría no ser posible.
- Además de la HLP, a menudo utilizo también el historial del proyecto (por ejemplo, el ritmo de avance anterior) como indicador del progreso futuro.
- Pasando a la columna de la derecha. Si estás trabajando en un proyecto nuevo con datos estructurados, la pregunta que me haría es si las funciones predictivas están disponibles
- ¿Tiene razones para pensar que los datos que tiene (entradas X) son fuertemente predictivos o suficientemente predictivos de las salidas objetivo Y?
- En este recuadro del cuadrante inferior derecho, si se trata de mejorar un sistema existente con datos estructurados, algo que ayudará mucho es si se pueden identificar nuevas características predictivas.
- ¿Hay funciones que aún no utiliza pero que podrían ayudar a predecir?
- También veremos la historia del proyecto

Why use HLP to benchmark?






People are very good on unstructured data tasks

Criteria: Can a human, given the same data, perform the task?



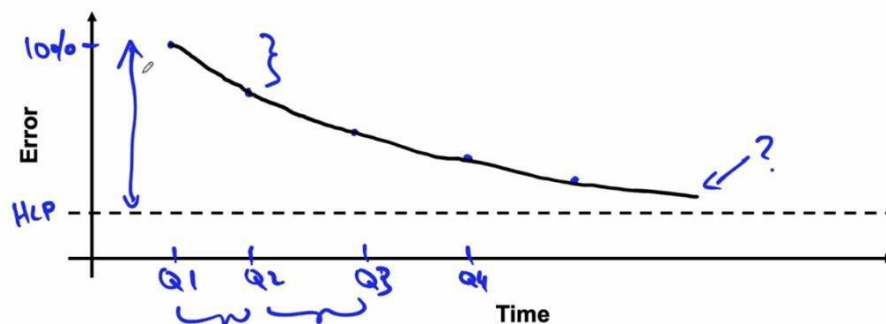
- Utilizo HLP para evaluar lo que podría ser factible para los datos no estructurados porque la gente es muy buena en las tareas de datos no estructurados.
- El criterio clave para evaluar la viabilidad de un proyecto es, dados los mismos datos exactos que se darían a un algoritmo de aprendizaje, ¿puede el ser humano realizar la tarea?
- Supongamos que estás construyendo un coche autónomo y quieres que un algoritmo clasifique si un semáforo está en rojo, amarillo o verde.
- Tomaré fotos del coche autónomo y pediré a una persona que mire una imagen como ésta (a la izquierda de la diapositiva), y veré si la persona que sólo mira la imagen puede decir qué lámpara está iluminada. En este ejemplo, está bastante claro que es verde.
- Pero si también tiene imágenes como éstas (a la derecha de la diapositiva), en las que no podemos saber qué lámpara está iluminada, se convertirá en un problema.
- Por eso es importante para este punto de referencia HLP asegurarse de que el humano recibe sólo los mismos datos que su algoritmo de aprendizaje.
- La prueba útil no es si el ojo humano puede reconocer qué lámpara está iluminada mientras está en el coche. La prueba útil es si la persona puede seguir haciendo la tarea si estuviera sentada en la oficina y sólo pudiera ver la imagen de la cámara
- En concreto, le ayuda a adivinar si un algoritmo de aprendizaje, que sólo tendrá acceso a esta imagen, también puede detectar con precisión qué lámpara del semáforo está iluminada.
- Es muy importante asegurarse de que un humano vea sólo los mismos datos que un algoritmo de aprendizaje.
- He visto muchos proyectos en los que durante mucho tiempo un equipo ha trabajado en un sistema de visión por ordenador, para luego descubrir que ni siquiera un humano que mirara la imagen podría entender lo que estaba pasando.
- Si te das cuenta de eso antes, entonces puedes darte cuenta antes de que simplemente no era factible con la configuración actual de la cámara
- Lo más eficiente habría sido invertir desde el principio en una mejor cámara o en una mejor configuración de la iluminación, en lugar de seguir trabajando en el algoritmo de aprendizaje automático

Do we have features that are predictive? ^x ^y

-  Given past purchases, predict future purchases ✓
-  Given weather, predict shopping mall foot traffic ✓
-  Given DNA info, predict heart disease ?
-  Given social media chatter, predict demand for a clothing style ?
-  Given history of a stock's price, predict future price of that stock ✗

- Para los problemas de datos estructurados, uno de los criterios clave para evaluar la viabilidad técnica es si tenemos características de entrada X que parecen predecir la salida Y.
- En el comercio electrónico, si se dispone de funciones que muestren cuáles son las compras anteriores de un usuario, es posible predecir las compras futuras, porque las compras anteriores son predictivas de las compras futuras.
- Si trabajas con una tienda física y quieres predecir la afluencia de público a los centros comerciales, una característica predictiva puede ser el tiempo. Sabemos que cuando llueve mucho, hay menos gente que sale de casa, así que el tiempo predice el tráfico de personas en los centros comerciales
- Veamos algunos ejemplos más. Dado el ADN de un individuo, intentemos predecir si este individuo tendrá una enfermedad cardíaca. Esto no lo sé, porque el mapeo del genotipo al fenotipo, o su genética a su condición de salud, es un mapeo muy ruidoso.
- Yo tendría sentimientos encontrados respecto a este proyecto, porque resulta que tu secuencia genética sólo predice ligeramente si padeces una enfermedad cardíaca.
- En otro ejemplo, teniendo en cuenta las conversaciones en las redes sociales, ¿se puede predecir la demanda de un estilo de ropa?
- Esta es otra pregunta dudosa. Si se puede predecir la demanda de ropa en este momento, ¿se puede predecir cuál será la tendencia de moda dentro de seis meses? La verdad es que eso parece muy difícil.
- A veces, los datos no son tan predictivos y se acaba con un algoritmo de aprendizaje que apenas supera a las conjeturas al azar.
- Por eso, analizar si tiene características que cree que son predictivas, es un paso importante de diligencia para evaluar la viabilidad técnica de un proyecto.
- Un último ejemplo que puede ser aún más claro, es dado un historial del precio de una acción bursátil en particular, puede usted predecir el precio futuro de esa acción.
- Toda la evidencia que he visto es que esto no es factible o realizable, a menos que se consiga algún otro conjunto inteligente de características que miren el precio histórico de una sola acción.
- La historia pasada no es predictiva del precio futuro de esa acción según las pruebas que he visto.
- Incluso dejando de lado la cuestión de si tales predicciones de valores tienen algún valor social, también creo que este proyecto es simplemente inviable desde el punto de vista técnico

History of project



- Un último criterio que he mencionado es la historia de un proyecto.
- Cuando he trabajado en una aplicación de aprendizaje automático durante muchos meses, he descubierto que los índices de mejoras anteriores pueden ser un predictor sorprendentemente bueno del índice de mejoras futuras.
- Veamos un reconocimiento de voz como ejemplo, y digamos que se trata de un rendimiento de nivel humano.
- Voy a utilizar HLP como nuestra estimación de Bayes o nivel de error irreducible al que esperamos llegar.
- Supongamos que al iniciar un proyecto, en el primer trimestre el sistema presentaba una tasa de error del 10%. Con el tiempo, en los trimestres siguientes, el error se redujo aún más en el segundo, tercer y cuarto trimestre.
- Resulta que no es un modelo terrible para estimar esta curva.
- Si quiere estimar lo bien que podría hacerlo el equipo en el futuro, un modelo sencillo que he utilizado es estimar la tasa de reducción de la tasa de error para cada período de tiempo fijo (por ejemplo, cada trimestre) en relación con el rendimiento de nivel humano.
- En este caso, parece que esta brecha entre el nivel actual de rendimiento y el nivel de rendimiento humano se está reduciendo tal vez un 30% cada trimestre, por lo que se obtiene esta curva que decae exponencialmente a lo que es HLP
- La estimación de este ritmo de avance le dará una idea de lo que podría ser razonable para el futuro ritmo de avance de este proyecto.

Diligencia sobre el valor

Diligence on value



Have technical and business teams try to agree on metrics that both are comfortable with.

- ¿Cómo se estima el valor de un proyecto de aprendizaje automático? A veces no es fácil de estimar, pero permítame compartir con usted algunas de las mejores prácticas.
- Digamos que estás trabajando en la construcción de un sistema de reconocimiento de voz más preciso para la búsqueda por voz
- Resulta que en la mayoría de las empresas, habrá algunas métricas que los ingenieros de aprendizaje automático están acostumbrados a optimizar, y algunas métricas que los propietarios del producto o negocio querrán maximizar. A menudo hay una brecha entre estas dos.
- Pero muchos equipos de aprendizaje automático se sentirían cómodos optimizando la precisión a nivel de palabra.
- Pero en un contexto empresarial, otra métrica clave es la precisión a nivel de consulta, es decir, la frecuencia con la que se acierta en todas las palabras de una consulta.
- Para algunas empresas, la precisión a nivel de palabra es importante, pero la precisión a nivel de consulta puede ser incluso más importante.
- Ahora nos hemos alejado del objetivo que el algoritmo de aprendizaje está optimizando directamente.
- Incluso después de acertar con la consulta, lo que más importa a los usuarios es la calidad del resultado de la búsqueda. La razón por la que la empresa puede querer asegurar la calidad de los resultados de búsqueda, es que da a los usuarios una mejor experiencia y por lo tanto aumenta el compromiso del usuario. Esto, a su vez, hará que vuelvan al motor de búsqueda más a menudo.
- Una brecha que he visto a menudo entre los equipos de aprendizaje automático y los equipos de negocio es que el equipo de ingeniería normalmente querrá trabajar en esto (es decir, la precisión a nivel de palabra), mientras que el líder de negocio puede querer trabajar en esto (es decir, los

- Para que un proyecto avance, suelo intentar que los equipos técnico y empresarial se pongan de acuerdo en las métricas con las que ambos se sienten cómodos.
- Esto a menudo requiere un poco de compromiso, en el que el equipo de aprendizaje automático puede estirarse un poco más a la derecha y los equipos de negocios se estiran un poco más a la izquierda.
- **Cuanto más nos acerquemos a la derecha, más difícil será para un equipo de aprendizaje automático dar realmente una garantía en términos de resultados**
- Muchos problemas prácticos requieren que hagamos algo más que optimizar la precisión de las pruebas.
- Que el equipo técnico y los equipos de negocio salgan un poco de su zona de confort es a menudo importante para llegar a un compromiso sobre un conjunto de métricas que el equipo técnico puede entregar y el equipo de negocio siente que puede crear suficiente valor para el negocio
- Otra práctica que he encontrado útil es hacer cálculos aproximados para relacionar el nivel de precisión con las métricas.
- Por ejemplo, si la precisión de las palabras mejora en un uno por ciento, basándose en una estimación aproximada, se puede suponer que, en consecuencia, mejorará la precisión del nivel de consulta en tal vez un 0,7 por ciento o un 0,8 por ciento.
- A continuación, se puede seguir calculando en qué medida esto mejorará la calidad de los resultados de búsqueda y la participación de los usuarios y, en última instancia, los ingresos,
- Se puede crear este tipo de cálculos brutos utilizando conceptos como las estimaciones de Fermi.
- Estos cálculos de vuelta a la realidad pueden ser una forma de ayudar a unir las métricas del equipo de aprendizaje automático y las métricas del negocio.

Ethical considerations

- Is this project creating net positive societal value?
 - Is this project reasonably fair and free from bias?
 - Have any ethical concerns been openly aired and debated?
- Como parte de la estimación del valor de un proyecto, le animo a que piense también en consideraciones éticas, como por ejemplo si este proyecto está creando un valor social positivo neto. Si no es así, espero que no lo haga.
 - También les animo a que piensen si este proyecto es razonablemente justo y libre de prejuicios
 - Las cuestiones de valores y ética dependen en gran medida del ámbito, y pueden ser muy diferentes cuando se trata de conceder préstamos o de prestar asistencia sanitaria o de recomendar productos en línea.
 - Le animo a que busque los marcos éticos que se han desarrollado para su industria y su aplicación.
 - En última instancia, si no crees que el proyecto en el que estás trabajando vaya a ayudar a otras personas o a que la humanidad avance, espero que sigas buscando otros proyectos más significativos a los que lanzarte.
 - En mi trabajo, me he enfrentado a decisiones difíciles en las que no estaba seguro de si un proyecto concreto era algo en lo que debía trabajar.
 - He descubierto que debatir en equipo y hacerlo abiertamente suele ayudarnos a responder mejor y a sentirnos más cómodos con cualquier decisión que tomemos.
 - He cancelado varios proyectos, a pesar de que el proyecto era económicamente sólido, porque no creía que fuera a ayudar a la gente

Hitos y recursos

Milestones & Resourcing

Key specifications:

- ML metrics (accuracy, precision/recall, etc.)
- Software metrics (latency, throughput, etc. given compute resources)
- Business metrics (revenue, etc.)
- Resources needed (data, personnel, help from other teams)
- Timeline

If unsure, consider benchmarking to other projects, or building a POC (Proof of Concept) first.

- La determinación de los hitos y la dotación de recursos implica la redacción de las especificaciones clave de su proyecto.
- Esto incluirá métricas de aprendizaje automático como la exactitud o la precisión-recuerdo. Para algunas aplicaciones, esto también puede incluir tipos de métricas de equidad.
- Las especificaciones también suelen incluir métricas de software relativas al sistema de software, como la latencia, el rendimiento, las consultas por segundo, los recursos informáticos disponibles
- También puede redactar estimaciones de los parámetros de negocio que espera conseguir con el proyecto que está definiendo, como por ejemplo el incremento de los ingresos.
- Además, escriba los recursos necesarios, por ejemplo, ¿cuántos datos? ¿De qué equipos? ¿Involucrar a qué personal o necesitar ayuda de equipos interfuncionales?
- Por último, el calendario en el que espera alcanzar determinados hitos o resultados.
- Si te resulta muy difícil redactar algunas de estas especificaciones clave, también puedes considerar la posibilidad de llevar a cabo un ejercicio de evaluación comparativa con otros proyectos similares en los que otros hayan trabajado antes, o construir primero una prueba de concepto para tener una mejor idea de las métricas que podrían ser factibles,
- Sólo después de haber realizado ese POC podrá utilizar esa información para determinar con mayor seguridad los hitos y los recursos necesarios para una ejecución a mayor escala del proyecto que tiene en mente.

Referencias

- [Ambigüedad de las etiquetas](#)
- [Tuberías de datos](#)
- [Línea de datos](#)
- [MLops](#)
- Geirhos, R., Janssen, D. H. J., Schutt, H. H., Rauber, J., Bethge, M., & Wichmann, F. A. (s.f.). Comparación de las redes neuronales profundas contra los humanos: reconocimiento de objetos cuando la señal se debilita*. Recuperado el 7 de mayo de 2021, del sitio web Arxiv.org: <https://arxiv.org/pdf/1706.06969.pdf>