**Lección 2** (resumen)

*Fuente: curso “Practical Deep Learning for Coders” (*[*Fast.ai*](https://www.fast.ai/)*)*

# Deep Learning en la práctica

Sirve para resolver multitud de problemas, pero podemos acercarnos con una de estas mentalidades erróneas:

* subestimar las limitaciones y sobreestimar la capacidades: lleva a resultados pobres al principio.
* sobreestimar las limitaciones y subestimar las capacidades: se aleja del uso de la herramienta.
* subestimar las limitaciones (lleva a errores en la ejecución) y capacidades (no se intentan soluciones que pueden ser beneficiosas)

Siempre será mejor trabajar de menos a más, averiguando los límites y capacidades del problema, asumiendo gradualmente los riesgos.

### **Comienzo del proyecto**

Pasos:

* Seleccionar el proyecto.
* Ver disponibilidad de datos

Consejos:

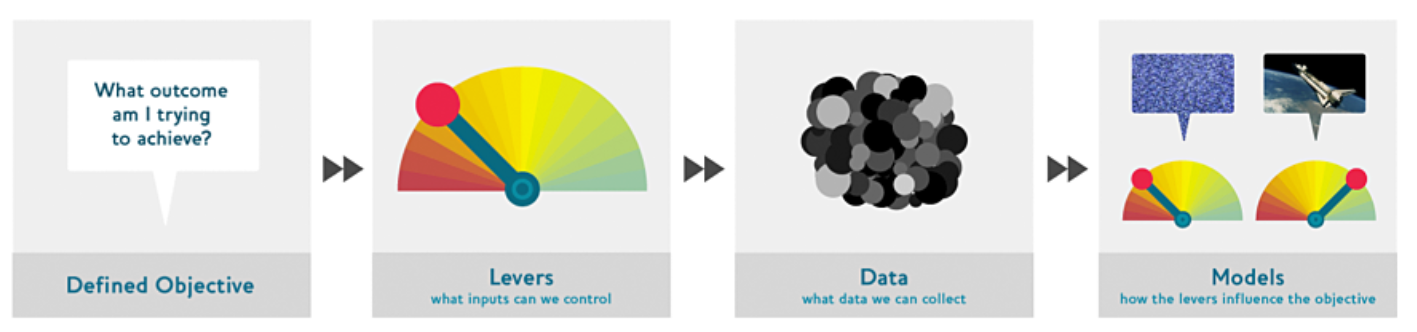
* trabajar de principio a fin en el proyecto, no esperar a tener el proyecto o datos perfectos. Esto ayuda a averiguar cuantos datos necesitamos.
* empezar con un prototipo que luego mejorará hasta que se ponga en producción
* si no encuentras proyecto, usa un proyecto que ya exista, como los que hay en las competiciones de [Kaggle](https://www.kaggle.com/).
* si estás empezando no escojas un proyecto demasiado novedoso, si algo falla no tendrás mucha ayuda.

### **Usos actuales**

#### Imagenes: reconocimiento de objetos, segmentación, otros usos diferentes de la imagen

* Texto (procesamiento del lenguaje natural:NLP): clasificar, resumir, generar textos (aunque no generan respuestas correctas), traducir.
* Combinando textos e imágenes (sin garantía de que resultados correctos, se aconseja supervisión humana)
* Datos en tablas y series temporales
* Sistemas recomendadores: aunque tienden a recomendar lo que gusta no lo que puede beneficiar (productos nuevos)
* otros tipos

### **Enfoque del tren motriz**



Define el **objetivo** que se busca -> averigua qué **acciones** puedes realizar y qué **datos** puedes conseguir -> construye el **modelo**

### **Datos**

En internet hay muchos, es necesario encontrar la forma de descargarlo.

Bing Image Search:

* registro en [Microsoft Azure](https://azure.microsoft.com/es-es/)
* usar la llave proporcionada (key) para realizar la búsqueda y descarga

key = os.environ.get('AZURE\_SEARCH\_KEY', 'XXXXX')

Nota: cuidado con los datos sesgados, ¿qué obtienes con tus búsquedas?

**Aumento de datos:** realizar variaciones aleatorias de los datos para que parezcan diferentes:

Rotar, deformación de perspectiva, cambios de brillo y cambios de contraste.

### **Procesar los datos para el modelo**

DataLoaders: almacenan los datos para entrenar y validar

* ¿qué tipos de datos tienes?
* ¿cómo tener un listado de los elementos?
* ¿cómo etiquetar los elementos?
* ¿cómo crear un conjunto para validación?

### **Entrenar modelo y limpiar datos**

Escoger un modelo para entrenar (Learner)

Obtener la matriz de confusión de las predicciones hechas por el modelo (ClassificationInterpretation)

Usar las herramientas de limpieza de fastai (ImageClassifierCleaner)

### **Uso del modelo en una aplicación online**

* Uso del modelo para inferir:

1. exportar el modelo (export)
2. cargar el modelo en la aplicacion (load\_learner)
3. usar modelo para hacer predicciones (predict)

* Crear una Notebook app con el modelo usando IPython widgets (ipywidgets) y Voilà
* Pasar el Notebook a aplicación real (Voila)
* Poner el modelo en producción: existen muchas opciones. Usar [Binder](https://mybinder.org/) por su sencillez

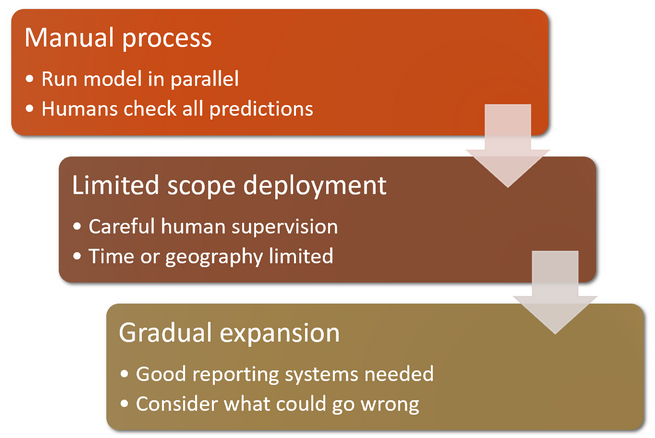
### **Uso del modelo en una aplicación móvil**

* integrar el modelo en la app móvil:
  + existen muchas opciones pero pueden no soportar toda la funcionalidad de fastai o Pytorch
  + será necesario crear una app por tipo de dispositivo (IOS, android,..)
* implementarlo en el servidor: (recomendado)
  + será necesaria conexión a internet
  + los tiempos de latencia pueden incrementarse
  + puede que use datos sensibles que no se quieran enviar al servidor

# Precauciones generales

* Considerar que el modelo forma parte de un sistema más grande
* Mayor dificultad de comprobar el funcionamiento que en una aplicación estándar:
  + datos fuera de dominio (no vistos en el entrenamiento)
  + cambio de dominio (cambia el entorno)

Solución para minimizar el riesgo:



* Consecuencias imprevistas, cuidado con los sesgos que retroalimentan el modelo. Considerar el caso en el que el modelo es altamente predictivo y la influencia en el comportamiento.