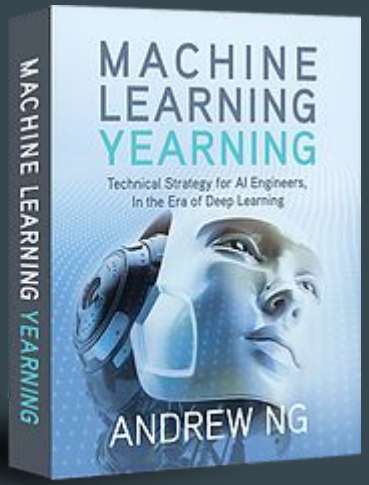
Estrategias para Machine Learning

Estrategias para Machine Learning

Referencias:

- Andrew Ng. “Machine Learning Yearning”. Draft, 2018. http://www.mlyearning.org/



- Experiencia personal.

- Disclaimer

Honest Machine Learning

Google:

- Cantidades astronómicas de datos
- Ejércitos de ingenieros
- Hectáreas de GPUs, memoria, etc.

Vos:

- 1500 datos ruidosos
- Una fracción de tu tiempo
- Una notebook del año 2009

Estrategias para Machine Learning

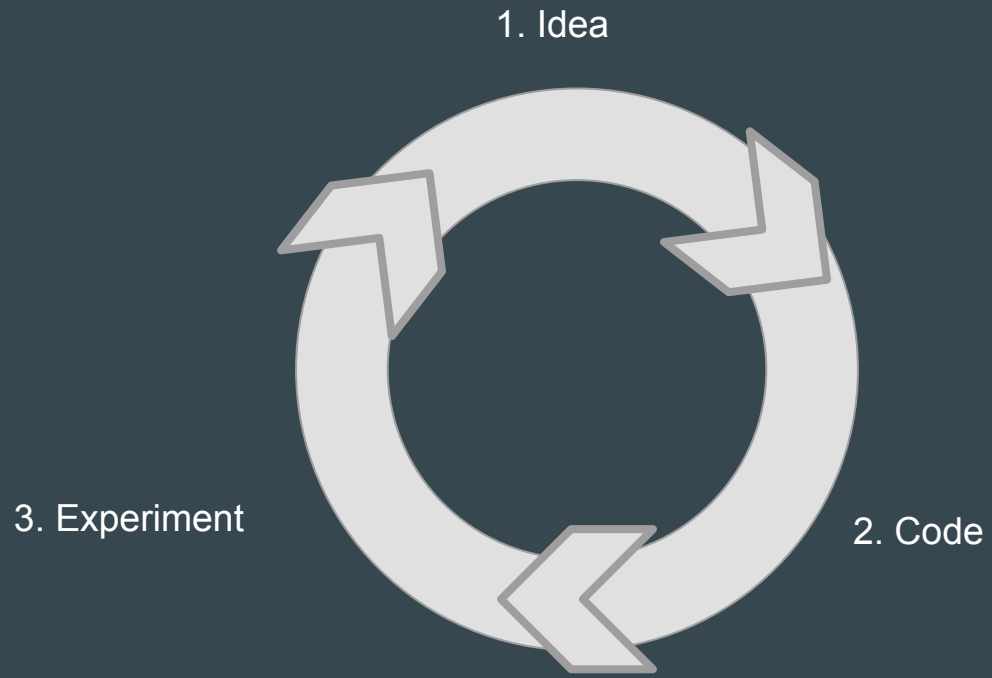
Queremos aplicar ML sobre un problema, de manera rápida y exitosa.

Lamentablemente, nuestro algoritmo anda mal. ¿Qué hacer? Opciones:

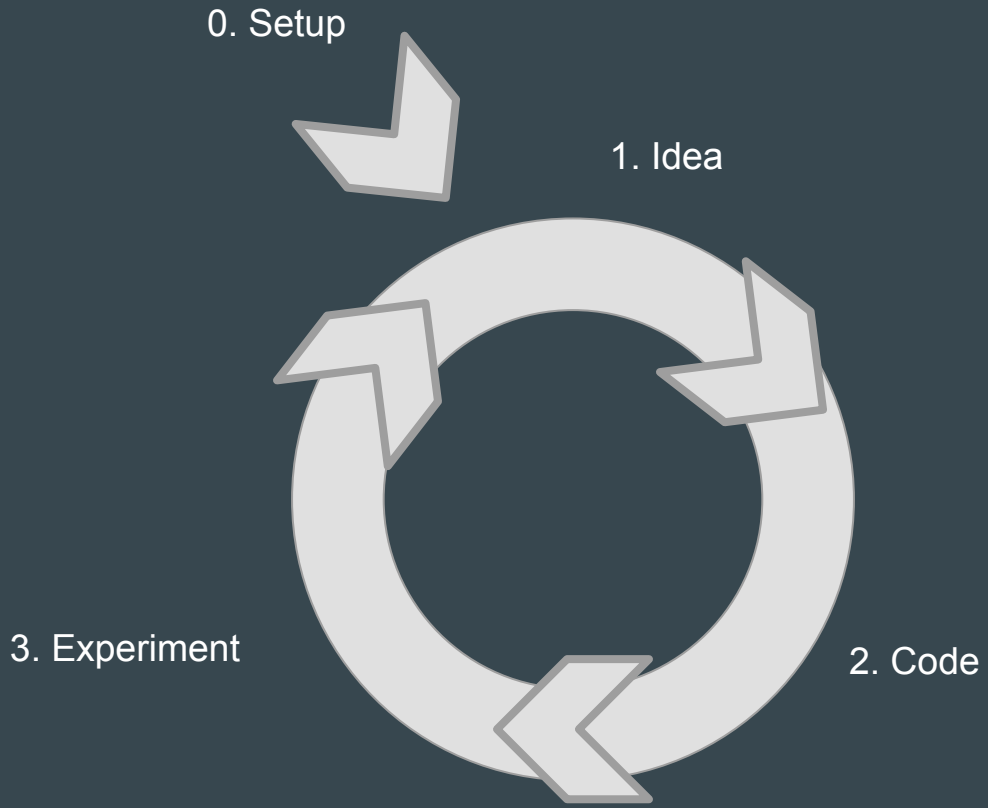
- Recolectar más datos, o datos más diversos
- Preprocesamiento: ingeniería de features, reducción de dimensionalidad, normalización, etc.
- Modelos de clasificación
- Parámetros / arquitectura: modelos más simples, o más complejos
- Entrenamiento

Hay que saber elegir!!

Método iterativo



Setup



Setup: Preparación de los Conjuntos de Datos

- Training: Entrenamiento
- Development: Para ajustar hiperparámetros, seleccionar features, analizar errores, etc.
- Test: Para obtener números finales de evaluación. **Nunca** para tomar decisiones.
- Dev y test **deben** responder a la misma distribución.
- Train no necesariamente.

Setup: Tamaño de los datasets

- Machine Learning clásico:
  - Split ~70/10/20
- Grandes cantidades de datos:
  - unos pocos miles para dev/test.
- Resolución: El tamaño del dataset indica la “resolución” de la accuracy
  - 100 elementos: 1%
  - 500: 0.2%
  - 1000: 0.1%
  - 10000: 0.01%

Setup: Métricas

- Accuracy:
  - Poco informativa para problemas desbalanceados
- Precision/recall/F1:
  - Binaria: Focalizar el problema en una de las dos clases.
  - Multiclase: Permite regular la importancia de cada clase (weighted macro-average).
- ROC AUC:
  - Más expresiva: evalúa probs/scores asignadas a todas las clases, no la predicción.

Setup: Métricas de optimización vs. satisfacción

- Establecer una **única** métrica numérica, cuyo objetivo es optimizar.

- Métricas secundarias:
  - Velocidad
  - Instancias sensibles que no pueden ser etiquetadas incorrectamente.
  - Valores mínimos de precision/recall para clases específicas.

- Definir criterios de “satisfacción” para las métricas secundarias.

Setup: Baselines

- Clasificadores “bobos” para calcular valores mínimos para las métricas.
  - Clase mayoritaria
  - Random uniforme
  - Random respetando distribución
- A veces también se pueden calcular upper bounds teóricas.

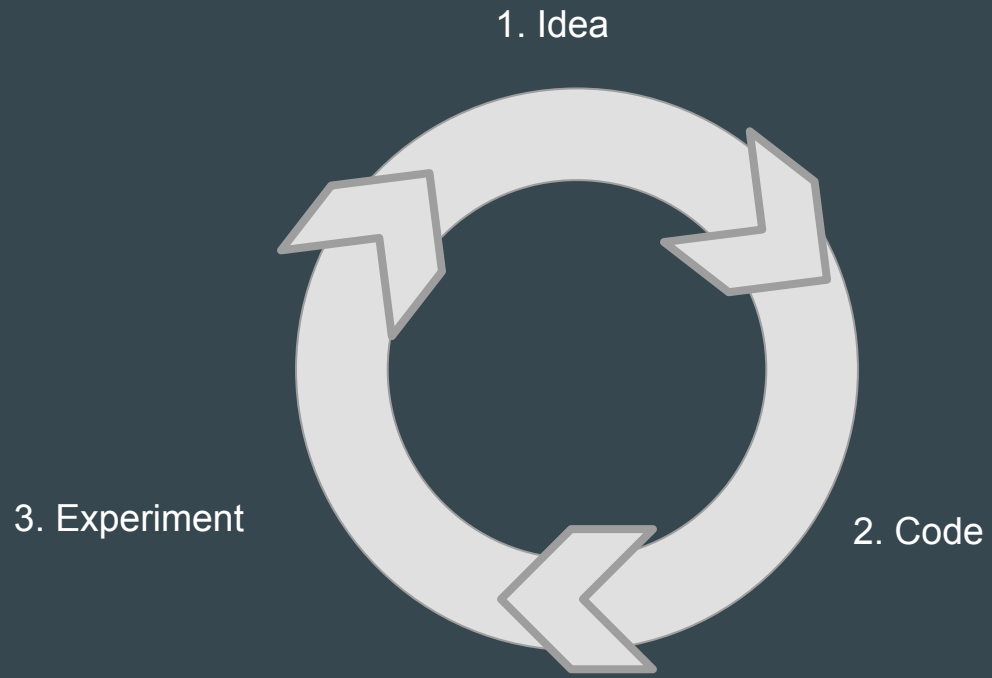
Setup: Rápido!

- Definir rápidamente los conjuntos de datos y las métricas objetivo.
- Permite iniciar el ciclo iterativo.
- Luego, los resultados y su análisis pueden indicar la necesidad de modificaciones:
    a. Los datos no reflejan la aplicación real: actualizar dev/test
    b. Overfitting en dev: se iteró muchas veces, actualizar dev.
    c. Las métricas no reflejan los objetivos.

Setup: Registro de Experimentos

- Historial de experimentos realizados.
- Registrar información necesaria para la reproducibilidad:
  - Fecha del experimento
  - Configuración del modelo
  - Resultado de las evaluación

Método iterativo



Primera Iteración: Sistema Básico

- No empezar tratando de construir el sistema perfecto.
- Construir y entrenar un sistema básico lo más rápido posible:
  - Vectorizador con opciones por omisión
  - Clasificador con opciones por omisión
- Evaluarlo y estudiarlo para decidir en qué direcciones avanzar.

Primera Iteración: Modelo de Clasificación

- Se pueden probar varios modelos de clasificación (DT, MNB, LR, SVM, etc.)
- Empezar eligiendo el que mejor ande sin ninguna configuración.
- **No** empezar con redes neuronales.
- No casarse con un único modelo.