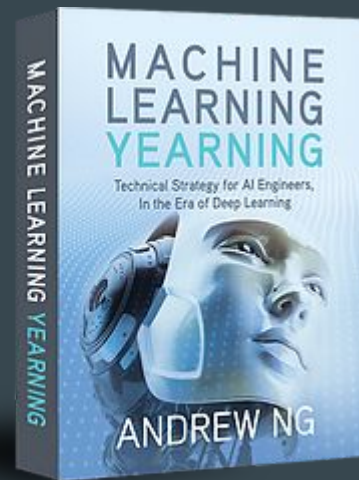


# Estrategias para Machine Learning

# Estrategias para Machine Learning

## Referencias:

- Andrew Ng. “Machine Learning Yearning”. Draft, 2018.  
<http://www.mlyearning.org/>
- Experiencia personal.
- Disclaimer



# Método iterativo

0. Setup



1. Idea



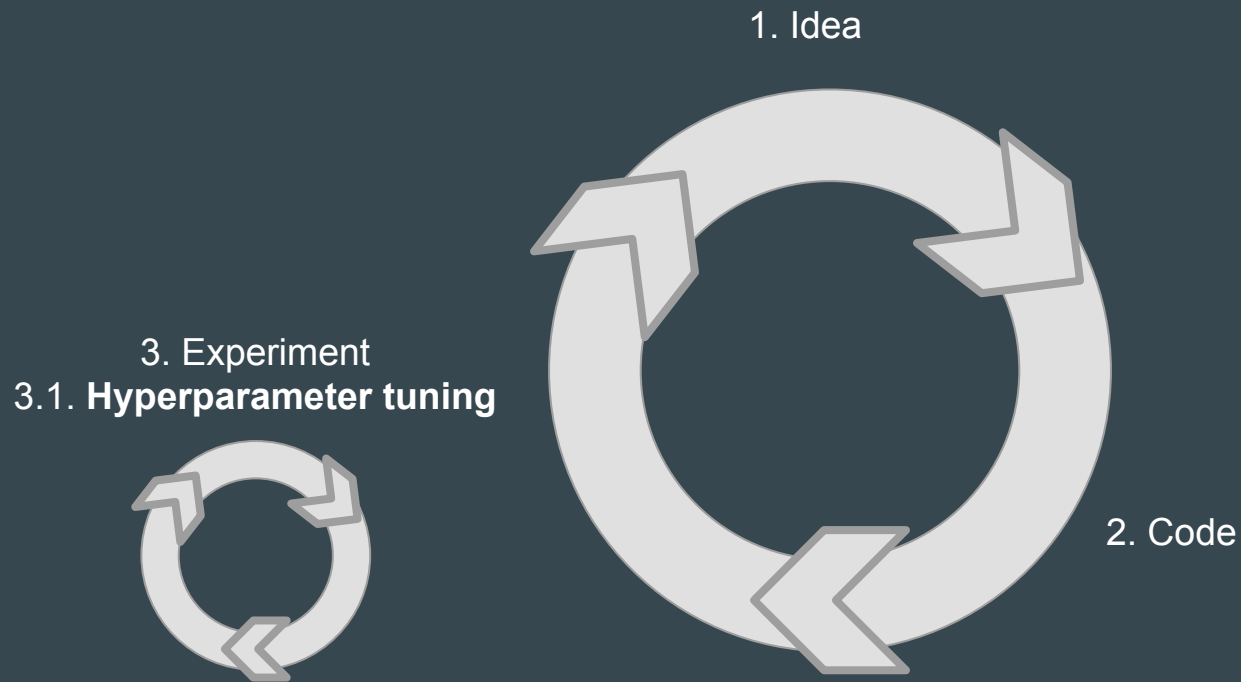
2. Code

3. Experiment

Vimos:

- Setup
- Primera iteración

# Ajuste de Hiperparámetros



# Ajuste de Hiperparámetros

Opciones:

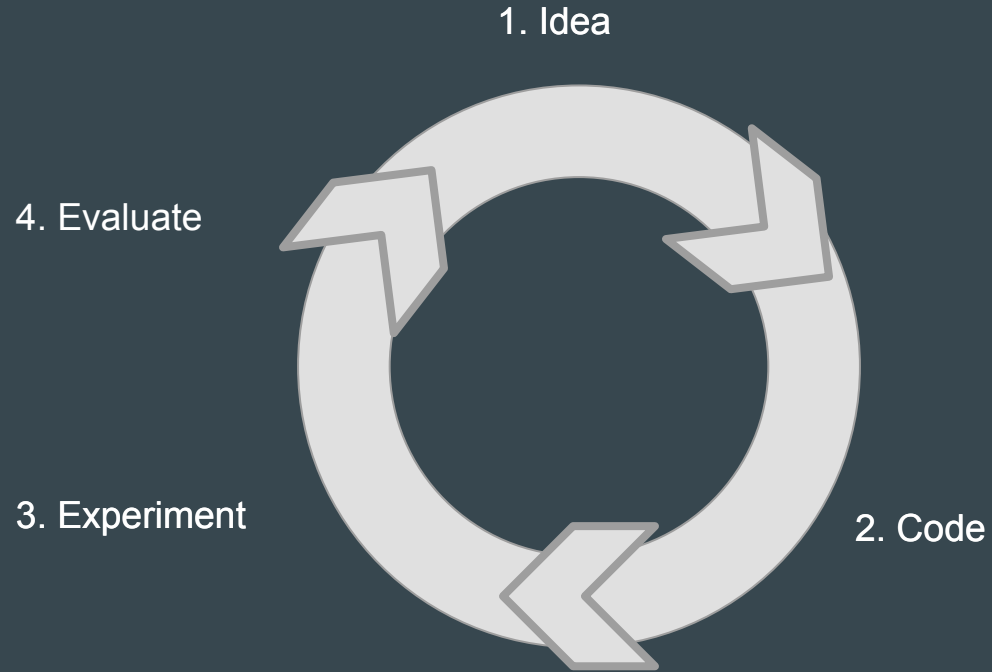
- Búsqueda manual.
- Búsqueda exhaustiva (grid-search): todas las combinaciones posibles de valores.
- Aleatoria (randomized): sampleando valores o combinaciones.
- Development vs. Cross Validation

# Ajuste de Hiperparámetros

Estrategias:

- ¡Leer documentación!
- Empezar con búsqueda manual. Elegir parámetros más relevantes.
- Seguir con búsqueda exhaustiva. Probar pocas combinaciones.
- Iterar.
- Guardar mejores configuraciones (no sólo la mejor).

# Evaluación



# Evaluación: Sesgo y Varianza

- **Sesgo (bias):** Error en el conjunto de entrenamiento.
- **Varianza (variance):** Error en el conjunto de development.
- **Error total:** bias + variance.
- Hacer Machine Learning = Bajar el error total.



# Evaluación: Sesgo

- **Sesgo alto:** el clasificador **ni siquiera** es capaz de aprender los datos de entrenamiento.
  - Anda peor que un sistema que memoriza los puntos de entrenamiento.
- ¿Cuánto quiere decir alto?
  - Depende del problema y de los valores a los que aspiramos.
  - Normalmente el sesgo **se puede reducir a cero. Se puede pero no necesariamente se quiere.**
- **PRIMER OBJETIVO DEL ML: CONTROLAR EL SESGO.**

# Evaluación: Reducción de Sesgo

- El sistema no logra aprender el conjunto de entrenamiento. No es lo suficientemente “expresivo” (underfitting).
- Soluciones:
  - Modelo más grande: agregar parámetros, capas, componentes, etc.
  - Modelo menos regularizado: salir del underfitting.
  - Features más expresivos: más dimensiones.
  - Modelo nuevo: clasificador diferente, otra arquitectura.

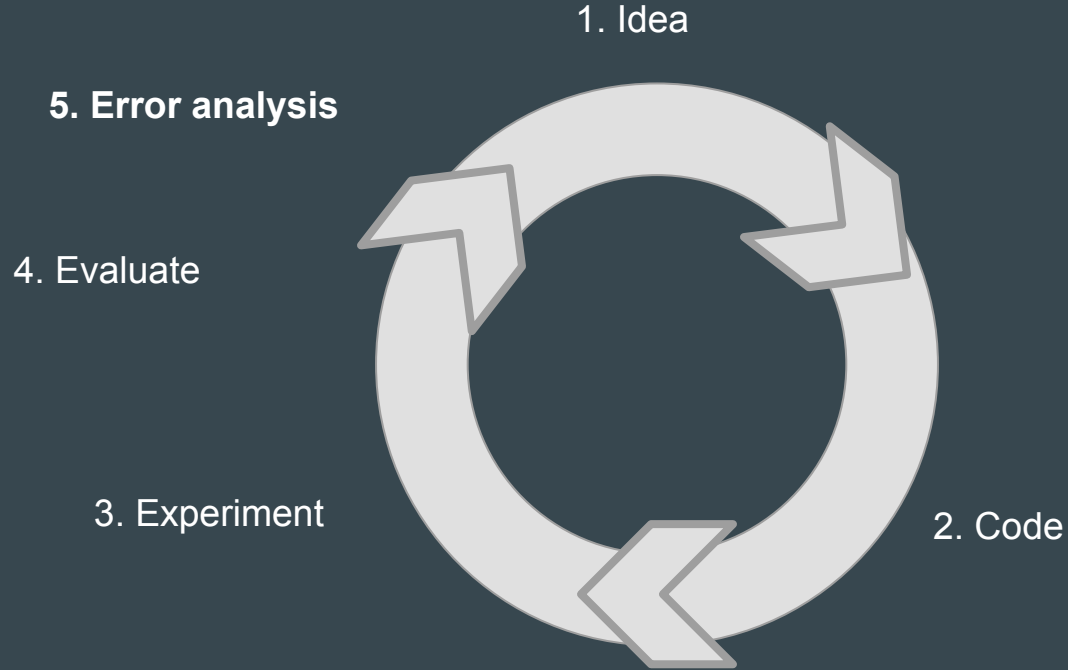
# Evaluación: Varianza

- **Sesgo bajo control:** Puedo hacerlo tan bajo como quiera.
- **Varianza alta:** No generaliza. No “aprende”. Memoriza. (overfitting)
- ¿Cuánto es varianza alta?
  - Nuevamente, depende del problema y de nuestros objetivos.
  - Con el sesgo controlado, la varianza es directamente proporcional al error total.
  - Con el sesgo controlado, **varianza cero = sistema perfecto.**
- **NUEVO OBJETIVO: Bajar la varianza tanto como se pueda. = HACER ML.**

# Evaluación: Reducción de Varianza

- El sistema no logra generalizar a partir del conjunto de entrenamiento.
- Posibles soluciones:
  - Más datos de entrenamiento. No hay de dónde aprender.
  - Mejores features: Facilitar al modelo el acceso a información valiosa.
  - Bajar expresividad: Regularización, early stopping, menos params., etc.
  - Modelo nuevo: clasificador diferente, otra arquitectura.

# Análisis de Error



# Análisis de Error (Error Analysis)

- ¿En qué se equivoca el modelo?
- Inspeccionar elementos mal clasificados.
- ¿Porqué se clasifica mal?
  - Ver la probabilidad / score de la clase correcta.
  - Ver features activos. Ver valores cercanos en instancias de entrenamiento.
  - Ver qué modificaciones del elemento hacen que se clasifique bien.
- Inspeccionar elementos **peor** clasificados (en base a prob/score)

# Análisis de Error (Error Analysis)

- Hacer una lista de ejemplos mal clasificados. (e.g., 50 de dev)
- Inspeccionar cada ejemplo. Identificar fuentes de error.
- Para cada fuente de error, identificar importancia y costo estimado.

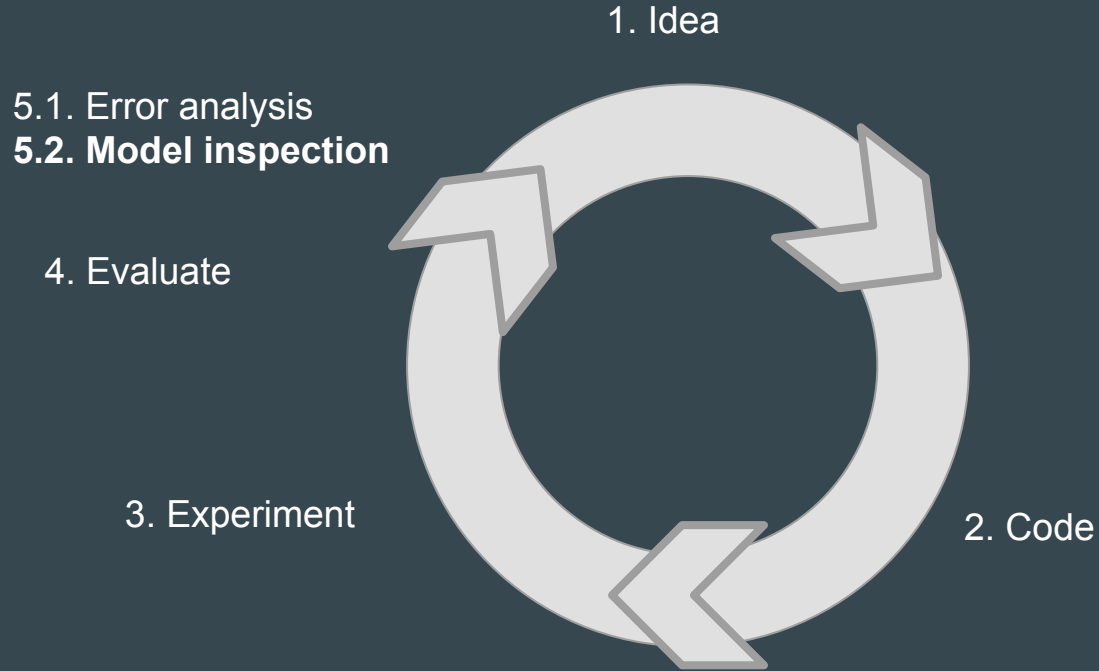
Audio clip	Loud background noise	User spoke quickly	Far from microphone	Comments
1	✓			Car noise
2	✓		✓	Restaurant noise
3		✓	✓	User shouting across living room?
4	✓			Coffeeshop
% of total	75%	25%	50%	

# Análisis de Error (Error Analysis)

- Subdivisión de development:
  - Eyeball dev set (~100 instancias)
  - Blackbox dev set (el resto)
  - Rotar cada tanto!
- Errores en el dataset:
  - Evaluar su impacto.
  - Si es importante, corregir en **todos** los datasets.



# Inspección del Modelo



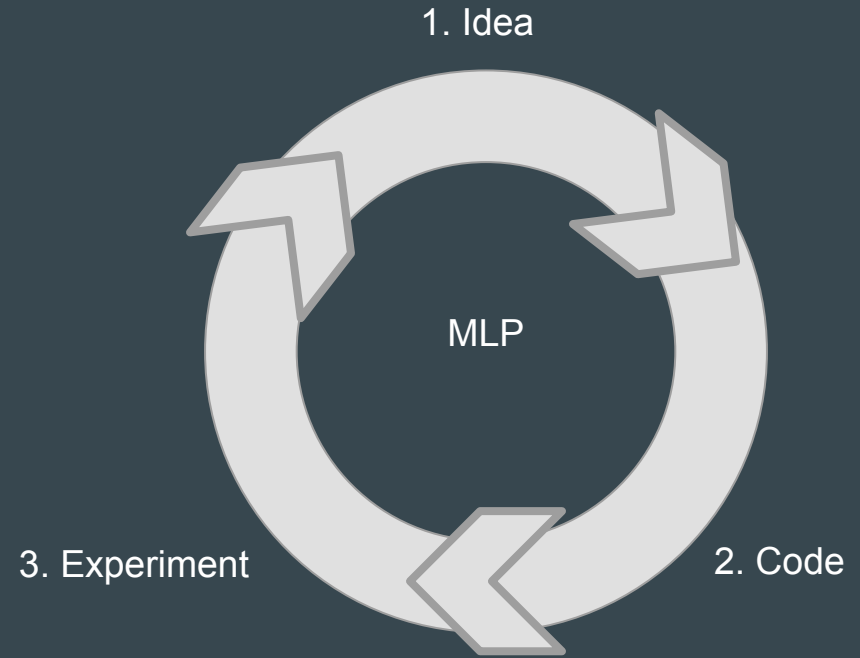
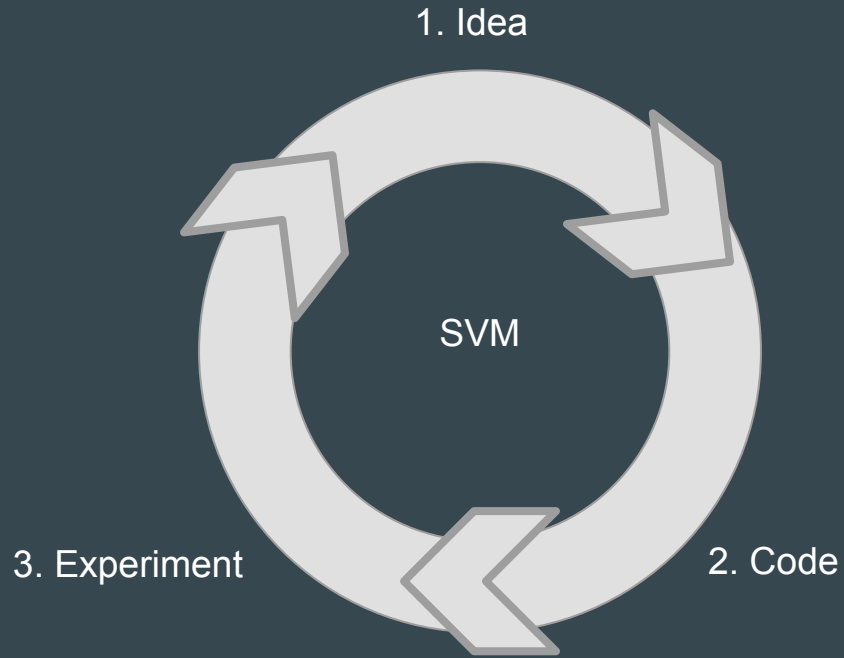
# Inspección del Modelo

- Estudiar los parámetros del modelo una vez aprendido.
- Features más influyentes para cada clase.
- Fronteras de decisión.

# Inspección del Modelo

- Modelos fácilmente inspeccionables:
  - **Decision Trees**
  - **Naive Bayes:** probabilidad de cada feature dada la clase (y prior de la clase)
  - **Logistic Regressions:** score de cada feature para cada clase (y bias o intercept)
- Más complicado:
  - **Random Forests:** son muchos árboles para ver!
  - **SVMs:** ver con qué features está más alineado el hiperplano.
  - **Redes Neuronales:** usar inputs para ver cómo reacciona la red.

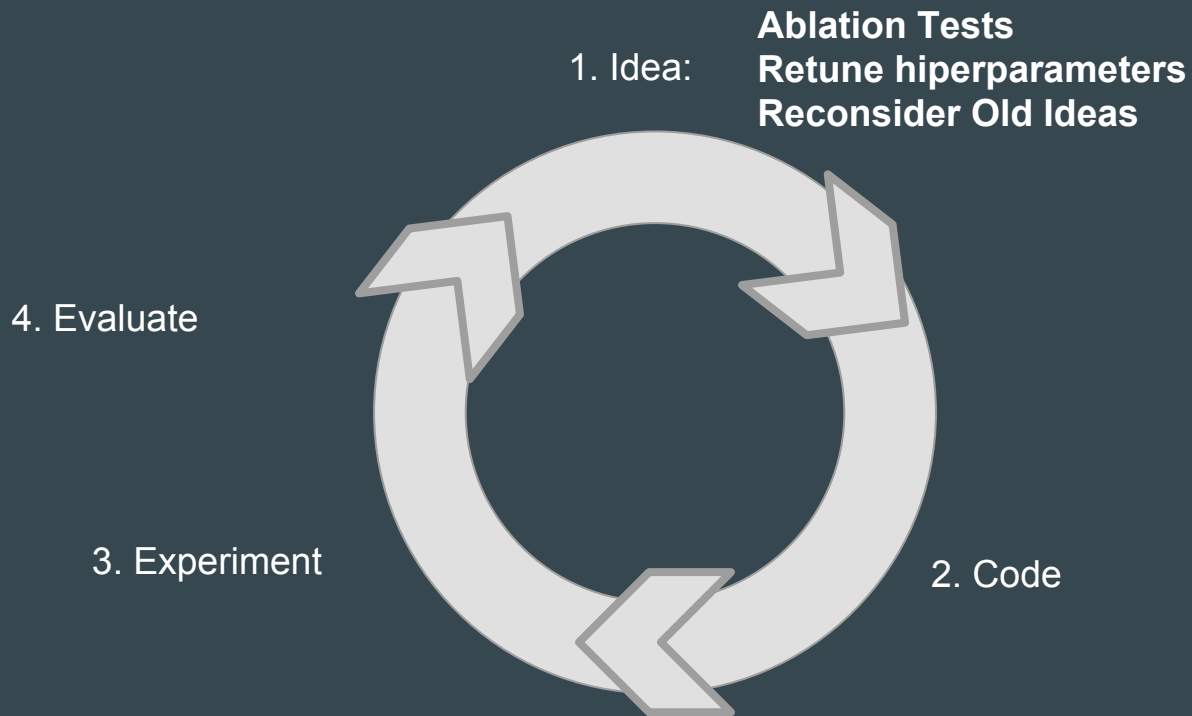
# Fork (Bifurcación)



# Bifurcación

- Empezar a mantener dos o más sistemas diferentes en simultáneo.
- Cada uno tiene su ciclo de experimentación.
- Con el tiempo, las configuraciones divergen.
- Ejemplo:
  - SVM / LR
  - Red neuronal: MLP / RNN / CNN

# Retrospectivas



# Retrospectivas

- Revisar ideas previas, tanto las aceptadas como las rechazadas.
- Ablation Tests: medir el impacto de cada componente del sistema actual.
- Hyperparameter retuning: Volver a hacer ajuste de hiperparámetros
- Reconsiderar viejas ideas

# Aumentación de Datos (Data Augmentation)

- Generar datos artificiales en base a los datos que tenemos.
- Las transformaciones deben preservar las etiquetas
- Imágenes: rotación, escala, espejado, cambio de color, etc.
- Texto: más difícil! sinónimos, traducción bidireccional, etc.