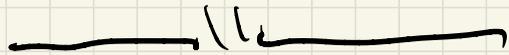


3. Estructuración Proyectos ML

Abril 2024

- Error stripping affects 1, 2 al tiempo w
are to have dificultad de usar.



- use single evaluation number
- satisfying metrics (running time)
- set up train / dev / test

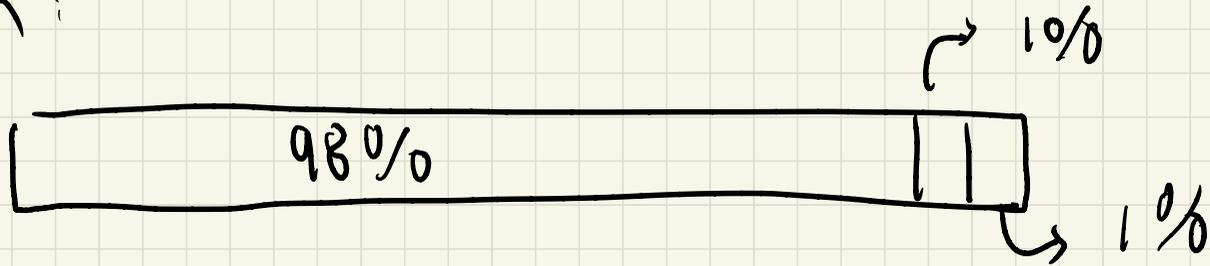
} choose one
to optimize
subject to
n satisfying
constraints.

TRAINING are running
LA MISMA DISTRIBUCION

y otras are seen de la misma distribución
con la que se va poner en producción.

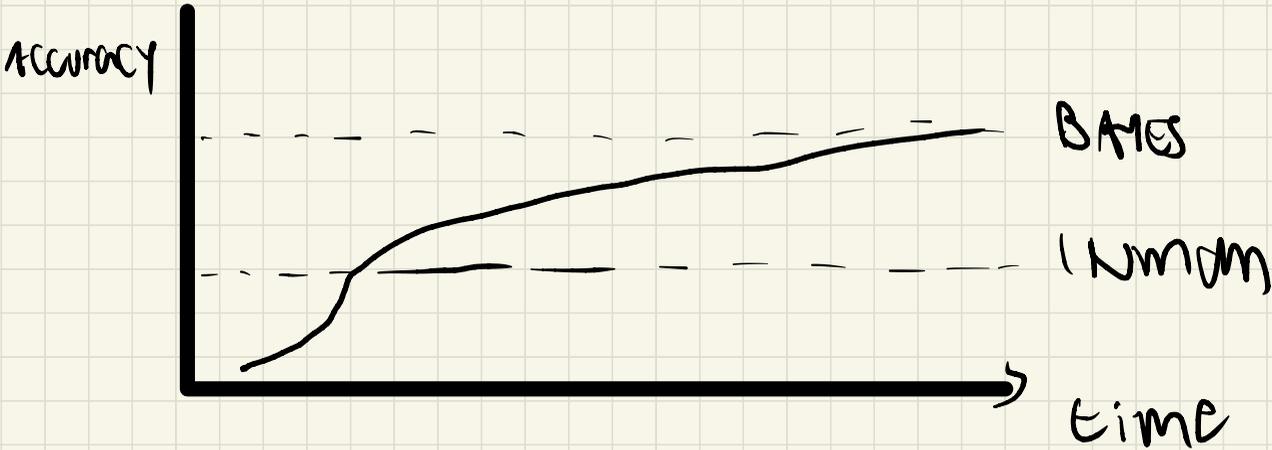
- en resumen elegir el dev set & métrica es elegir el objetivo del problema

- división en ent - dev / test. Actualmente es más común:



- se puede poner pesos en los errores de clasificación. Por ejemplo, cometer error en clasificar una imagen apropiada (cambio métrica). Parte de la definición correcta del problema, debe revelar: uso y preferencias

- Rendimiento humano



- Después de sobrepasar el nivel humano es difícil progresar con la ayuda de humanos.

- SESGO EVITABLE

- pensar en el error humano como una proxy de error bayesiano

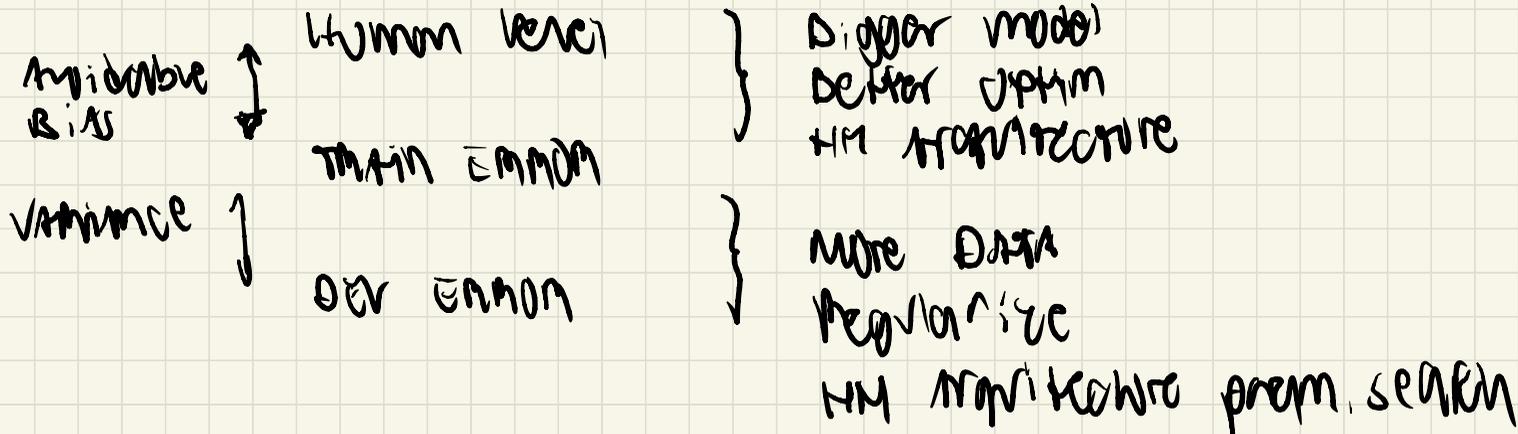
- LA DIFERENCIA ENTRE ERROR HUMANO Y ERROR DE ENTRENAMIENTO SE DEMONIHA SESGO EVITABLE: se puede o no evitar en reducir sesgo en entrenamiento.

AYUDA A ENFOCARSE ←

- human error ↑ avoids bias
- train error
- dev error ↑ variance

- si el error de entrenamiento es menor que el humano, enfocarse en la diferencia entre ambos IDE.

- RESUMEN



② ANÁLISIS DE RESULTADOS

- Análisis de errores: estudiar los errores en la base de desarrollo para diseñar estrategias PARA MEJORAR MODELO. HACER TABLA
- Errores en la validación de la base de entendimiento y desarrollo.
- en resumen: ① USAR ANÁLISIS SESGO/VARIANZA
② ANÁLISIS DE ERRORES

③ DATOS DE ENTRENAMIENTO DISTINTOS DE DEV TEST

DATOS PAGINAS WEB \approx 200.000

DATOS DE APP \approx 10.000

PROB. CLASIFICACION

OBJETIVO hacer que funciones bien en la App.

opcion 1 : mezclar todo y separar en
entrenam / des / prueba (todas misma dist.)

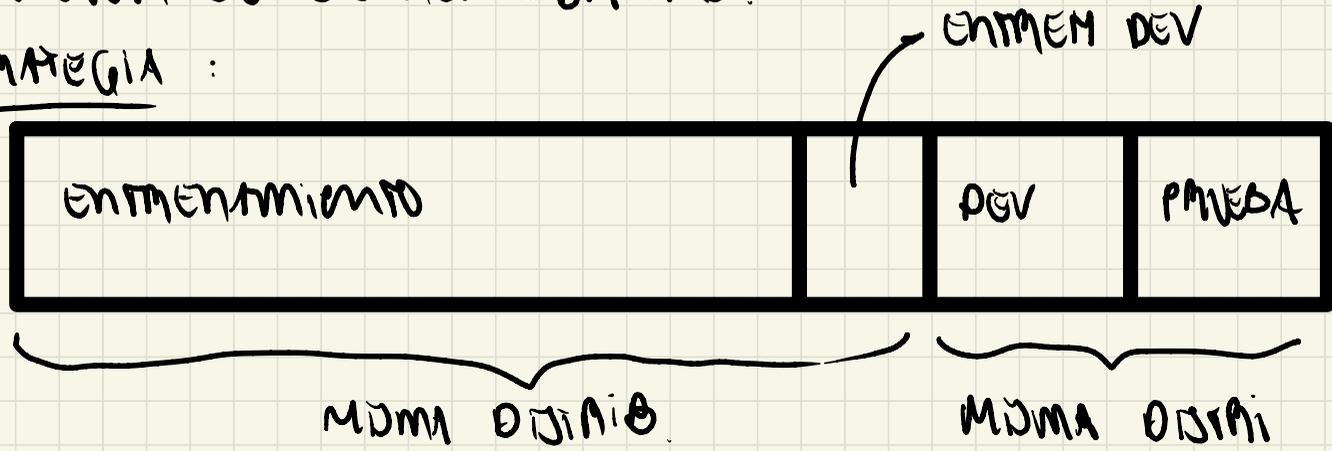
opcion 2 :



misma distri.
funcional

- Analisis sesgo / varianza cuando datos de entren / dev no vienen de distrib. distintas.

ESTRATEGIA :



- Hacem el analisis de sesgo varianza y errores en :

ENTRENAM
ENTRENAM DESARROLLO
DESARROLLO

y se puede comparar con nivel humano

Ejemplo

Humano

4 %

↓ sesgo evitable

Entren

7 %

↓ varianza

Entren prueba

10 %

↓ diferentes distrib. (DATA MISMATCH)

Desarrollo

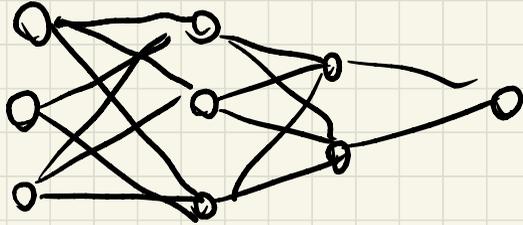
12 %

↓ overfit ?

Prueba

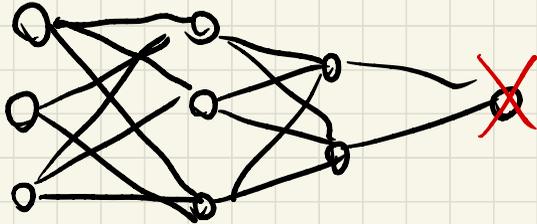
- PODRIA PASAR QUE LOS ERRORES DISMINUYAN!
- EL PROBLEMA DE DATA MISMATCH PUEDE MITIGARSE SIMULANDO DATOS ARTIFICIALES (DATA SYNTHESIS) TENIENDO CUIDADO DE NO SOBREPRESSENTAR SOLO UN TIPO DE DATOS PARA QUE EL MLGO NO HAGA OVERFITTING.

④ TRANSFER LEARNING



RED ENTRENADA (MUCHOS DATOS)

↓ Transferir



Reentrenar solo la última
CAPA

DEPENDIENDO DE LA CANTIDAD
DE DATOS SE PUEDE REENTRENAR
MAS CAPAS (POCOS DATOS)

- EN GENERAL ES UTIL CUANDO:

1. TAREA A y B tienen input similares (imágenes, sonido)
2. Se tiene mucho datos en A y pocos en B
3. Características de "bajo nivel" (en la mitad de la red de A) pueden ayudar a la tarea B.

⑤ MULTITASK LEARNING (cuando se tiene mas o menos la misma cantidad de datos para cada tarea, competencias o clases concurrentes)

- Entrenamos una sola red para clasificar ejemplos con multiples categorias incluso en ausencia de marcas:

$$y^{(i)} = \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix}$$

$$y^{(i)} = \begin{bmatrix} ? \\ \vdots \\ 0 \end{bmatrix} \quad \begin{matrix} 1 \\ \vdots \\ k \end{matrix}$$

- Estrategia: definir adecuadamente funcion perdida

$$L(\hat{y}^{(i)}, y) = \sum_{j=1}^K -y_j^{(i)} \ln(\hat{y}_j^{(i)}) - (1 - y_j^{(i)}) \ln(1 - \hat{y}_j^{(i)})$$

- A diferencia de softmax aqui un output puede tener varias clases.

- Ejemplo: objetos en una imagen.

⑥ END TO END LEARNING VS. SUBTASK LEARNING

- Depende fundamentalmente de los datos / CAMBIOS
- Problemas de endogeneidad en la mitad del proceso se genera los datos
- VALOR DE USAR CARACTERÍSTICAS BASADAS en EXPERIMENTAL / TEORÍA