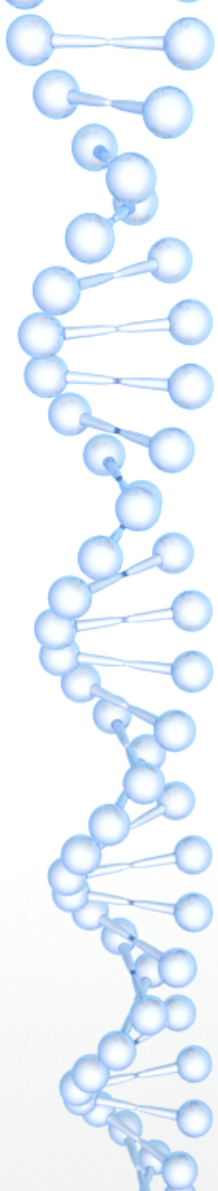
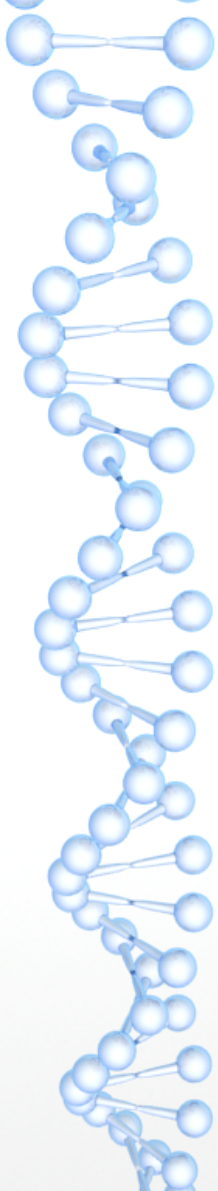


Trabajo Final - DSI

Luis Cabañero Gómez

KDDCup 2010

- Tutor inteligente
- Cada fila representa un paso
  - Incorpora campos de tiempo, duración, pistas, fallos, aciertos, conocimientos requeridos…
- Hay que predecir si se acierta a la primera

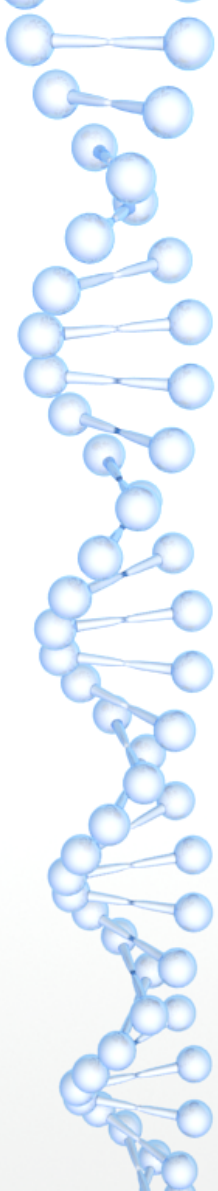


Problema propuesto

- Sobre los problemas en vez de los pasos
- Predecir la dificultad de un problema para un alumno
- Solo se ha usado el fichero de 2005-2006

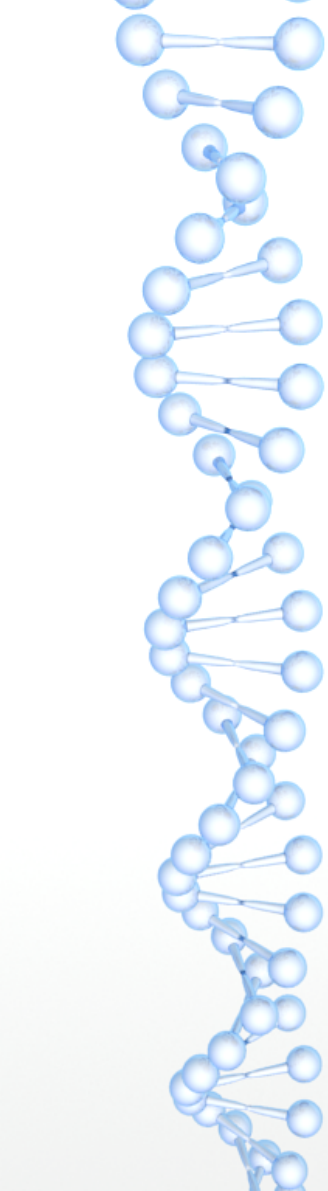
Herramientas

- Python
- pandas
- NumPy
- scikit-learn
- matplotlib
- seaborn



Preprocesamiento

- Cargado del fichero
- Agrupación de pasos en problemas
- Agrupación de problemas en esquemas
- Preparación de los datos del modelo



Cargar el fichero

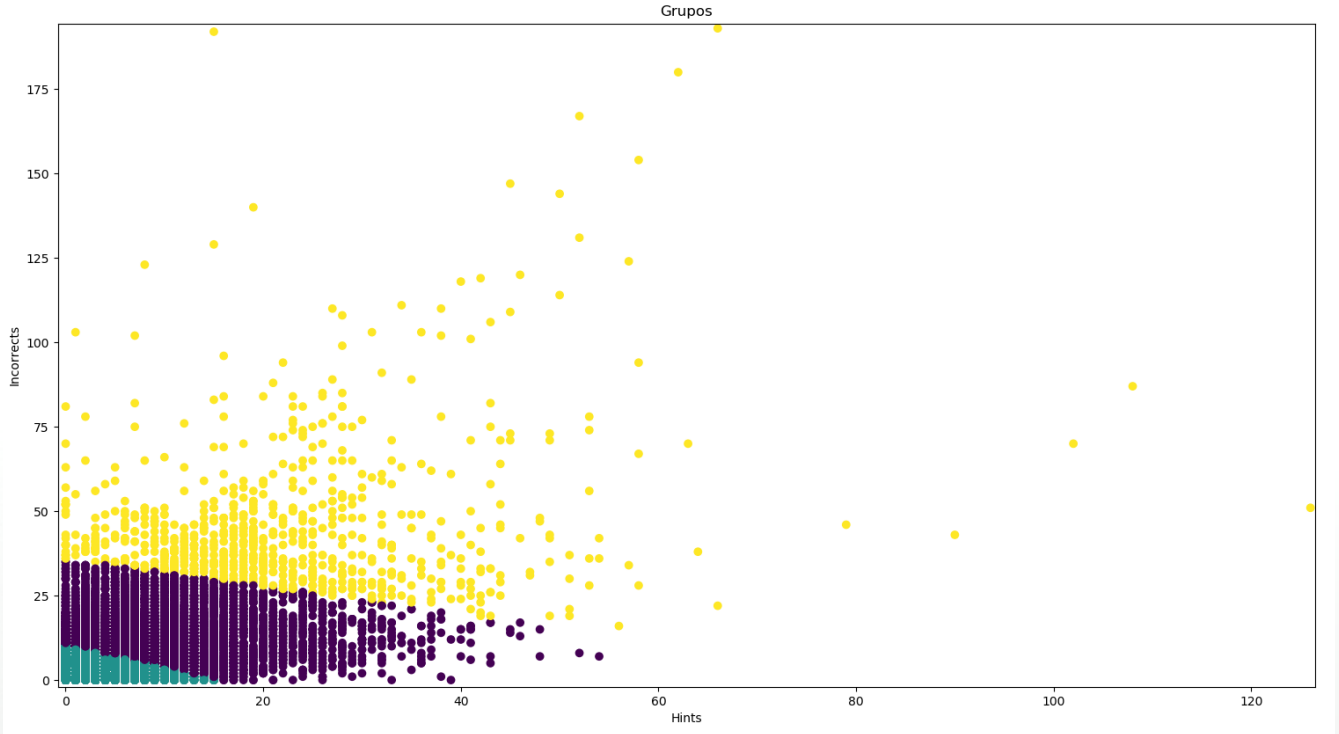
- Selección de una proporción de los datos
  - En función de los alumnos
- Transformación de fechas
- Transformación de KC y Opportunity en listas

Agrupación de pasos en problemas (1/2)

- Fallos, pistas y aciertos se suman
- CFA se promedia
- Fecha de inicio es la menor fecha de inicio
- Fecha de fin es la mayor fecha de fin
- Duración de los pasos se suma
  - Aunque también se crea otro campo nuevo de duración: fecha de fin-fecha de inicio
- KC se concatenan y luego se convierten en columnas
  - 1 si está presente y 0 si no

Agrupación de pasos en problemas (2/2)

- Se añade el grupo de dificultad mediante KMeans sobre las pistas y fallos.



Agrupación de problemas en esquemas

- Se calculan promedios
- Interesan los KC
  - 1: es imprescindible
  - 0: no es relevante

Preparación de los datos para el modelo

- Entra un problema y un alumno y sale la dificultad
- Features temporales(por alumno):
  - Número de problemas previos
  - Promedio de pistas, errores, aciertos, duración y CFA
  - Aprendizaje de los KC
- Features del problema:
  - Sustituir KC por los del esquema

Aprendizaje de los KC

- Penalización por pistas = $2^{-Hints/2}$
  - Se multiplica por los KC del problema
- Penalización por tiempo = $2^{-segundos/segundos\ por\ semana}$
  - Se multiplica por el aprendizaje previo
- Combinación de ambos mediante el máximo

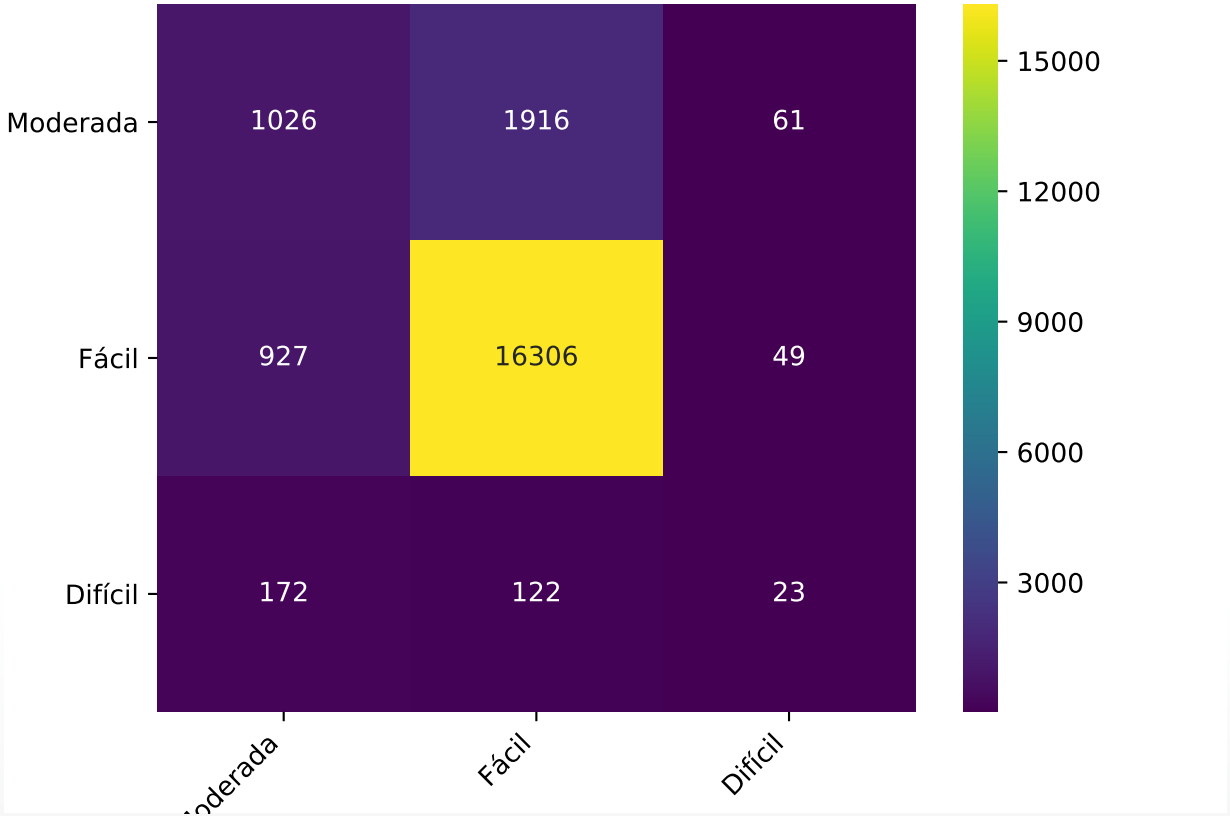
Modelo

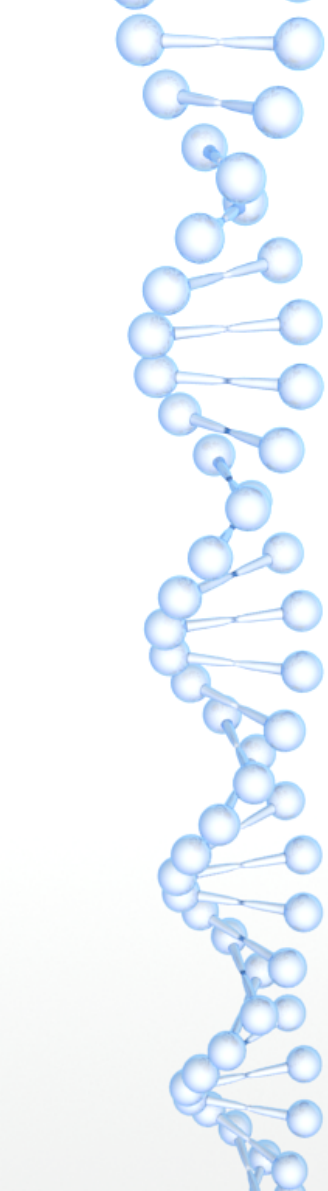
- Random Forest
  - Elegido probando varios algoritmos
- Número de estimadores = $2 \cdot \sqrt{nFeatures}$
- Los demás parámetros por defecto

Evaluación

- Validación cruzada: 3 splits
  - Precisión: 0.8479 (0.0124)
  - F1-Macro: 0.4412 (0.0055)
- Matriz de confusión
  - 60% de entrenamiento
  - 40% de test

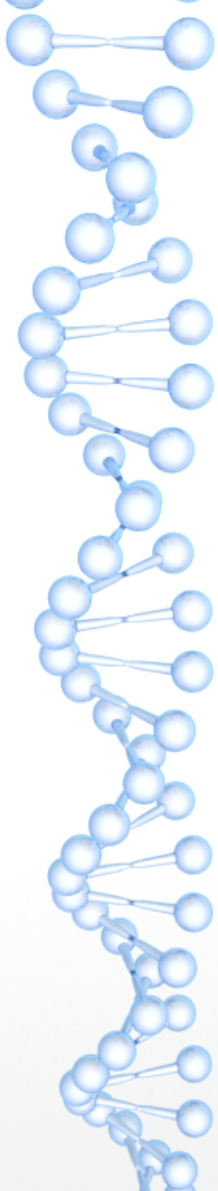
Matriz de confusión





Conclusiones

- Resultados no muy buenos
  - Aunque algo útiles
- Mejorable mediante:
  - Redes neuronales
  - Más información (los alumnos aprenden también fuera del tutor)
  - Menos ruido en los datos
  - Usar otras características
- Se han aprendido lecciones



Fin