



# Análisis de las fases del sueño

---

**Presentación y trabajo creados por:**

Arturo Sirvent Fresneda y  
Ángel Guevara Ros

*Máster Ciencia de Datos (UV)*

2022

# Índice

1. Objetivo.
2. Preprocesado de datos.
3. Modelo 3 pacientes.
4. Modelo 10 pacientes.
5. Aproximación alternativa.
6. Conclusiones.



# 01

## Objetivo

---

¿Qué resultado queremos obtener?



# Motivación

---

- El análisis de las fases del sueño es una ardua tarea que conlleva una gran cantidad de tiempo.
- Esto puede provocar colas de espera y atrasos en la realización de dicha prueba.
- Por tanto, sería interesante el desarrollo de una herramienta que agilizara este proceso y sirviera de ayuda a los médicos.

## ¿Qué buscamos?

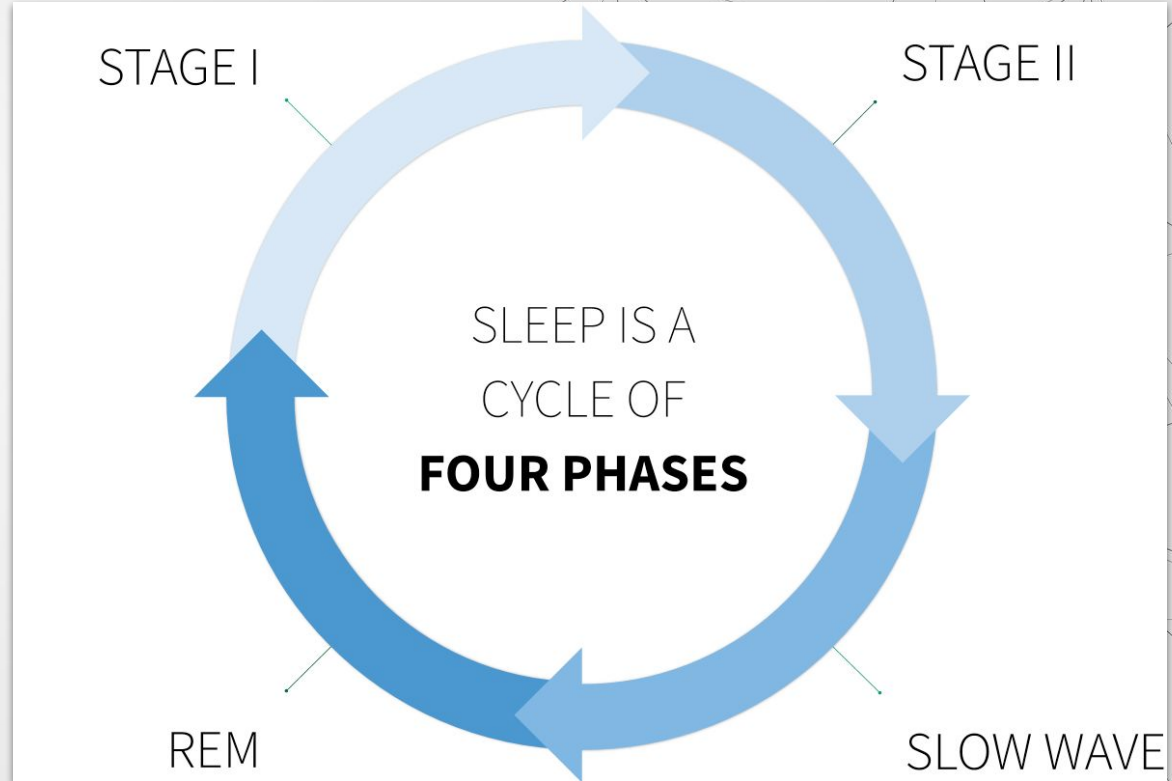
---

- El objetivo del trabajo es predecir las fases del sueño en las que se encuentra un paciente a lo largo de una noche.
- Esto lo haremos a través del polisomnograma correspondiente.
- Dispondremos de un total de 10 pacientes con los que poder trabajar. De cada uno tenemos su hipnograma y el etiquetado por parte de dos expertos.



Consideramos 5 fases posibles:

1. Despierto.
2. Fase N1.
3. Fase N2.
4. Fase N3.
5. Fase REM.



Stages	EEG					EOG	EMG
	<i>Delta</i> ( $< 4$ Hz)	<i>Theta</i> (4 - 7 Hz)	<i>Alpha</i> (8 - 13 Hz)	<i>Beta</i> ( $> 13$ Hz)	<i>Other</i> EEG patterns		
<b>AWAKE</b>			x	x		0.5-2 Hz	Variable amplitude but usually higher than during sleep stages
<b>N1</b>		x	x		Vertex waves	Slow eye movement	Lower amplitude than in stage awake
<b>N2</b>		x			K-complexes; Sleep spindles	Usually no eye movement, but slow eye movements may persist	Lower amplitude than in stage awake and may be as low as in stage REM
<b>N3</b>	x				Sleep spindles may persist	Eye movements are not typically seen	Lower amplitude than in stage N2 and sometimes as low as in stage REM
<b>REM</b>		x	x		Saw tooth waves	Rapid eye movement	Low chin EMG tone; usually the lowest level of entire recording



# 02

## Preprocesado de datos

---

¿Cómo lo hacemos?



## Procesado

- Nos hemos quedado solo con 3 canales : C3-A2, LOC-A2 y X1.
- Hemos hecho un downsampling de la señal a 100 tomas por segundo.
- Se ha aplicado un filtro de pasa alta de hasta 50 Hz.
- Hemos dividido la señal en ventanas de 30 segundos, que es lo habitual en los softwares de análisis del sueño.
- Se han eliminado las últimas 30 épocas de cada paciente porque no se consideran válidas.
- Además, no hemos utilizado todas las muestras posibles. Hemos eliminado aquellas que tenían alguna anotación en la columna de eventos.





# 03

## Modelo con 3 pacientes

---

Primera aproximación al problema.  
¿Cómo nos ha ido?

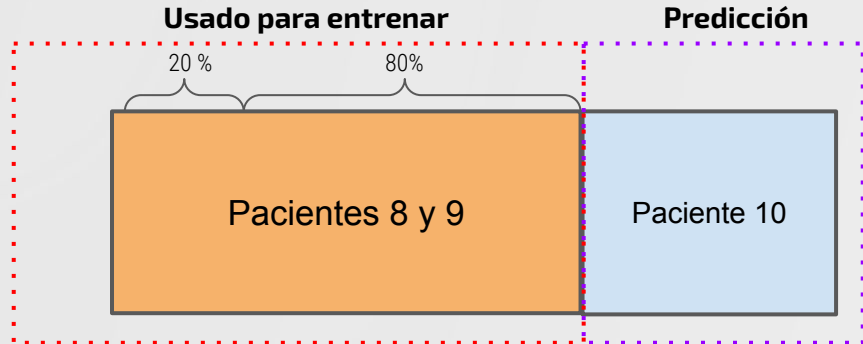
# Extracción de características

- Para la búsqueda de variables para el modelo, hemos usado el paquete **“yasa”** especializado en análisis del sueño.
- Tiene una función que automáticamente nos extrae 149 características de la señal que le pasemos.
- Entre esas características se encuentran por ejemplo las ondas alpha, beta, delta y theta.
- Otra de las funcionalidades del paquete, es la del etiquetado automático del polisomnograma, que es justamente lo que vamos persiguiendo nosotros.



# Modelo

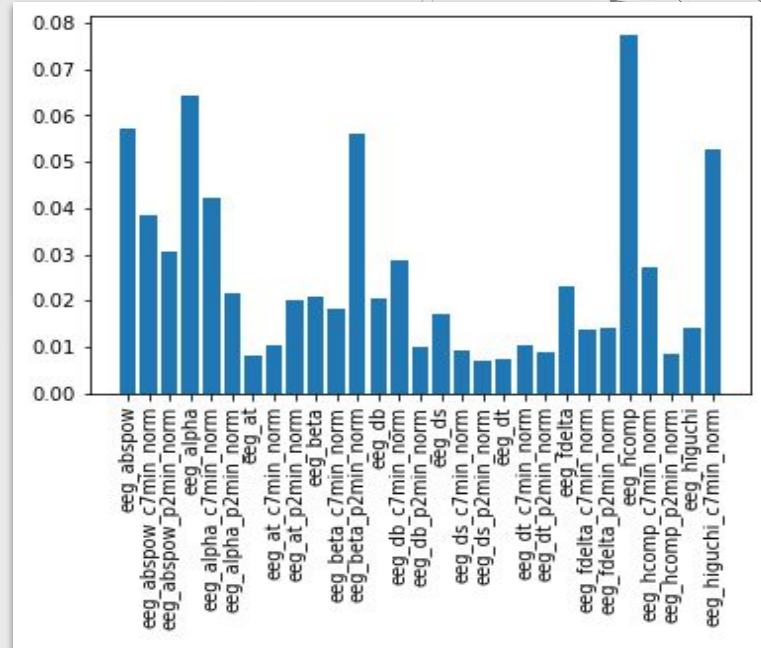
- Hemos usado un Random Forest con una aproximación OneVsRest.
- Esto quiere decir, que transformamos el problema multiclase en 5 problemas en los que comparamos 1 de las clases contra el resto.
- Hemos utilizado un 80% de los datos de los pacientes 8 y 9 para entrenar el modelo y un 20% para validarlo.
- Después hemos predicho sobre el paciente 10.



# Reducción de características

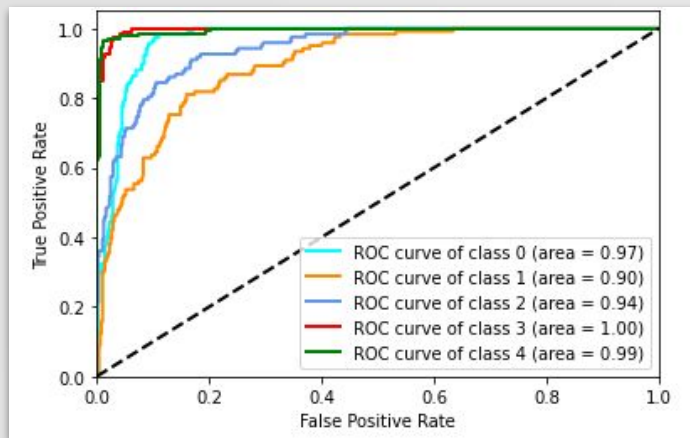
- Una vez que tenemos el modelo entrenado, podemos calcular la importancia que éste le da a cada una de las variables que usa.
- Así, nos hemos quedado solo con las más importantes para el modelo.
- También hemos añadido algunas que tenían sentido clínico pero que para el modelo no eran importantes.
- Finalmente nos quedamos con un total de 52 características (de 149 que teníamos).

**Importancia de características**

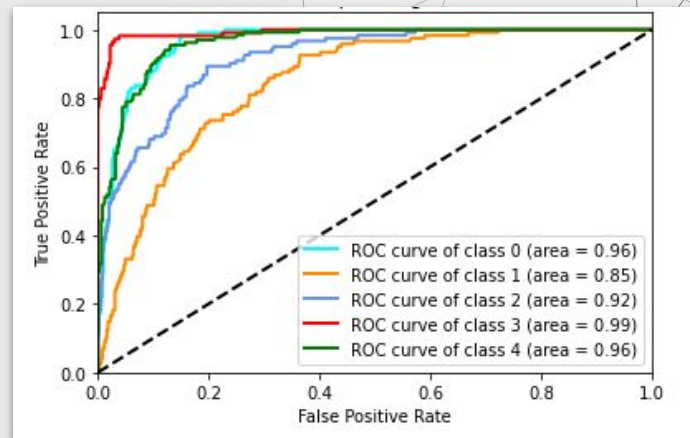


# Resultados en test

## Modelo con todas las características



## Modelo con las características importantes



## Después de modificar umbrales

	Accuracy_train	AUC_train	Accuracy_test	AUC_test
Modelo				
RF	0.894231	0.984781	0.741722	0.959263
RF_carac_reducidas	0.881410	0.973186	0.759934	0.936173

# 04

## Modelo con 10 pacientes

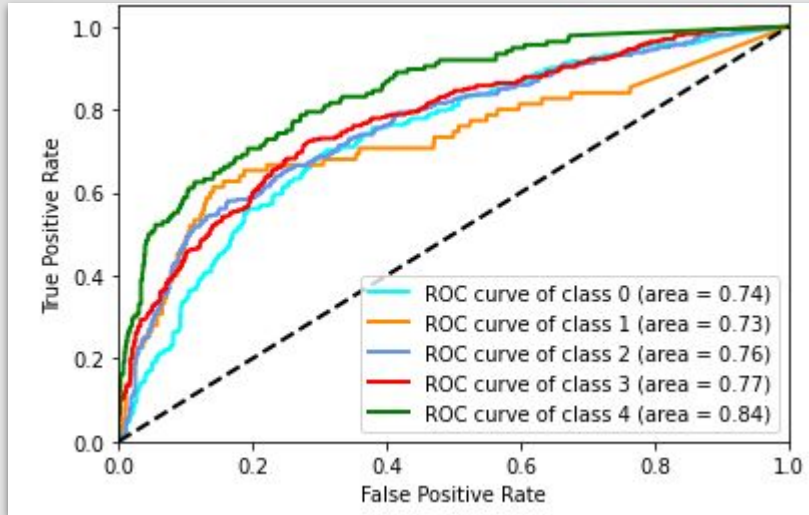
---

Entrenamos con 8 pacientes y comprobamos con dos.  
¿Qué ocurre al generalizar con más pacientes?

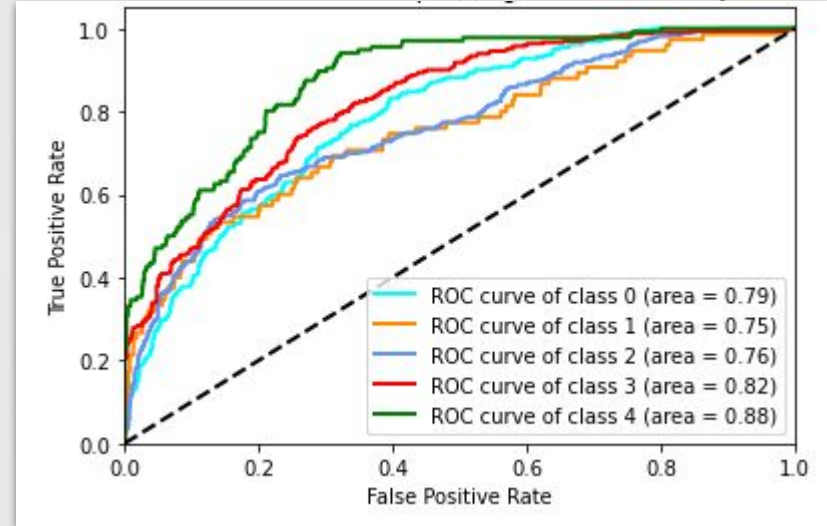


# Resultados en entrenamiento

**Modelo con todas las características**

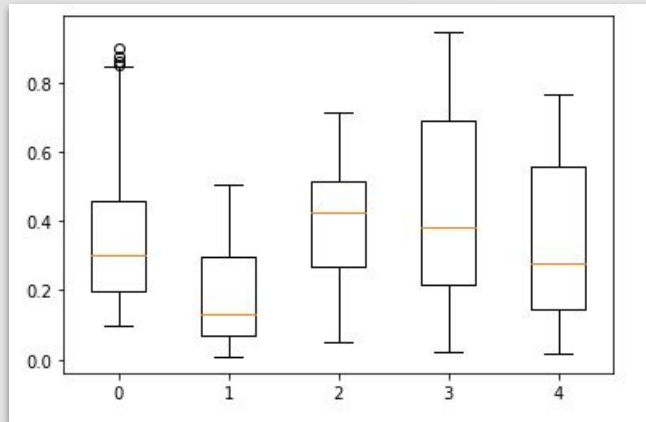


**Modelo con las características importantes**



# Ajuste de umbrales

- Ajustando los umbrales de decisión del modelo es posible mejorar los resultados.
- Sin embargo, esto puede ser complicado cuando las salidas del modelo para cada clase se solapan, ya que no pueden estar muy claros dichos umbrales.



**Distribución salidas probabilísticas**

**Confusion Matrix**

	0	1	2	3	4
0	1.2e+02	5	83	40	17
1	11	15	30	11	8
2	35	8	2.1e+02	48	19
3	49	0	97	1.8e+02	9
4	16	1	41	12	66
	0	1	2	3	4

True Class

Predicted Class



# 05

## Aproximación alternativa

---

¿Cómo podemos abordar el problema desde otra perspectiva?



# Planteamiento

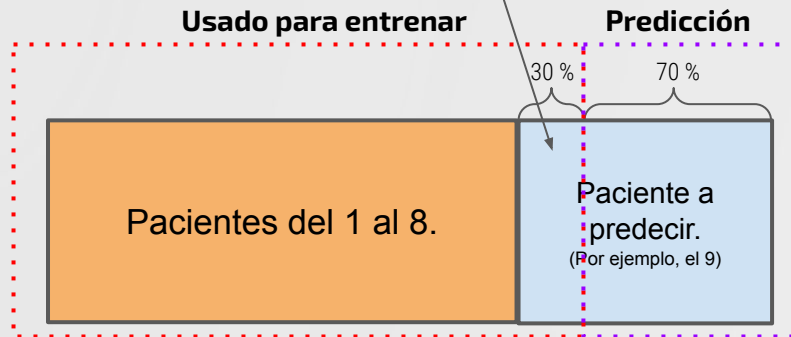
## Problema

- Si entrenamos para un paciente, el modelo sí es capaz de clasificar muy bien sus fases de sueño. El problema viene cuando pasamos de un paciente a otro, pequeñas variaciones hacen la generalización inter-paciente muy complicada.

## Solución

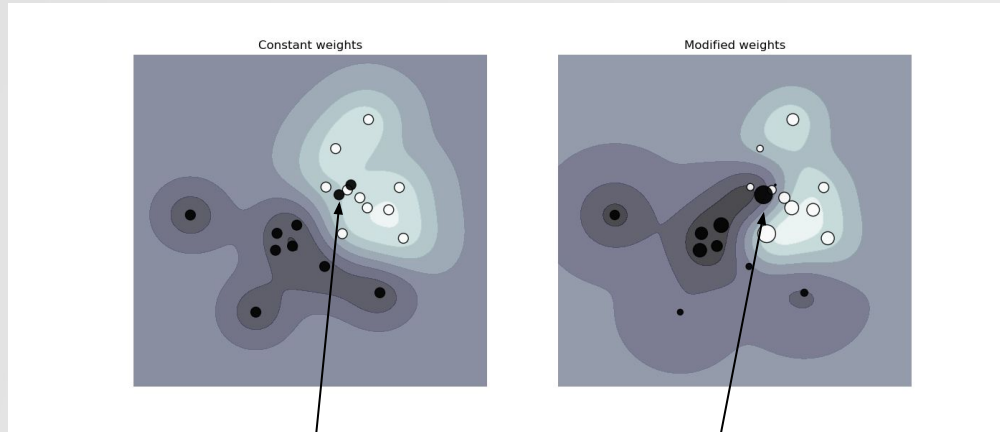
- Vamos a entrenar un modelo que use un conjunto limitado de datos etiquetados del paciente, para hacer **fine-tuning** sobre el modelo.

Este 30% **NO** está etiquetado, por lo que el profesional lo debería etiquetar manualmente. A cambio, el modelo le proporciona el **70%** restante etiquetado con un alto porcentaje de precisión (~92%).



# Modelo usado

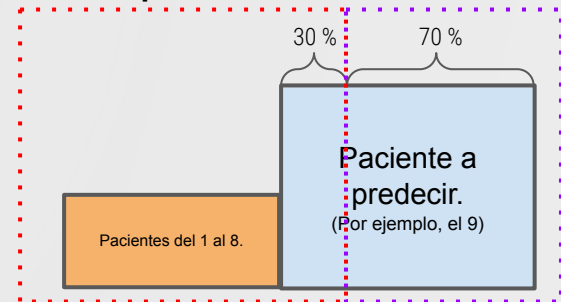
- Usamos una SVM para clasificar 5 clases, en donde los puntos tendrán diferentes pesos. Con esto conseguimos ese *fine-tuning* del modelo, con respecto al 30% de los datos del paciente a predecir.



Los "ignora"

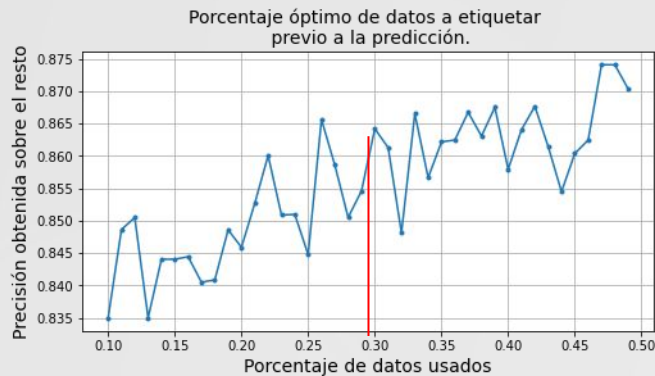
"Sobreajusta" sobre  
esos puntos

Usado para entrenar      Predicción

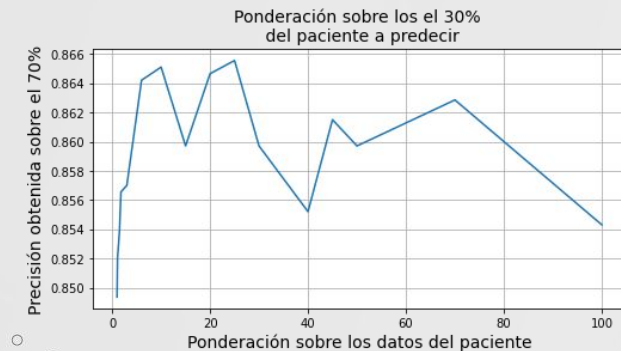


# Resultados

**¿Por qué usar el 30%?**



**¿Qué importancia dar al 30% del paciente a predecir?**



# Resultados

**Primera aproximación:** ~75% de acierto.

**Segunda aproximación:** ~87% de acierto.

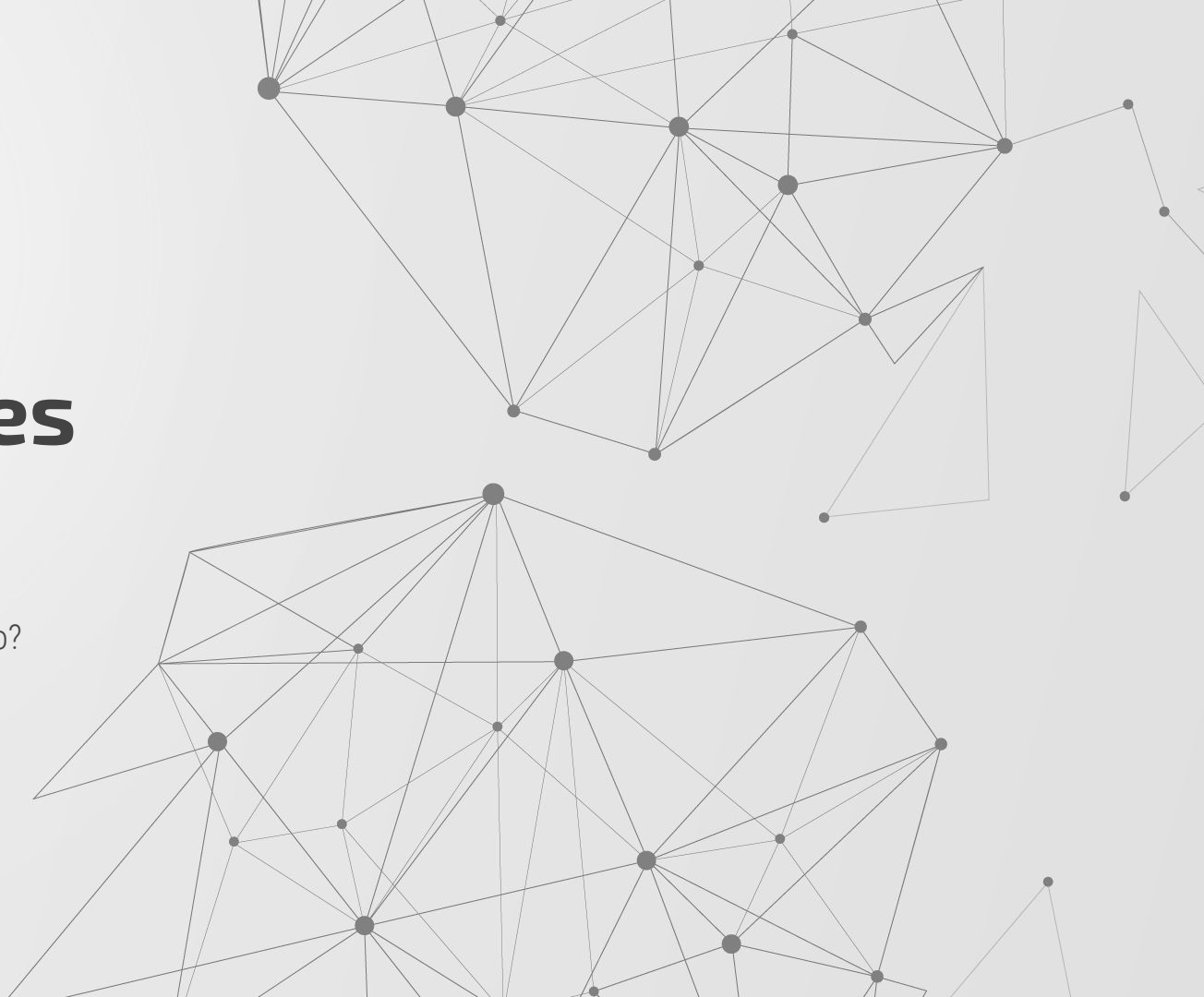
Con un control más exhaustivo y específico de los umbrales para cada clase, podemos obtener precisiones cercanas al ~92%

# 06

## Conclusiones

---

¿Qué resultados finales hemos obtenido?



## Resultados y observaciones

- Usar un modelo para cada paciente mejora bastante los resultados respecto a lo obtenido con un modelo más general.
- Además, los tiempos de entrenamiento son cortos por lo que es un planteamiento asequible.
- Usando un 20%-30% de los datos etiquetados por el médico, nos bastaría para terminar de predecir el resto de fases.
- No es una solución perfecta al problema, pero nuestra propuesta ayudaría a los médicos a que no tuvieran que hacer a mano el etiquetado entero, ahorrándoles hasta un 70% del tiempo de etiquetado.



# Gracias

¿Preguntas?

## Recursos y bibliografía usada:

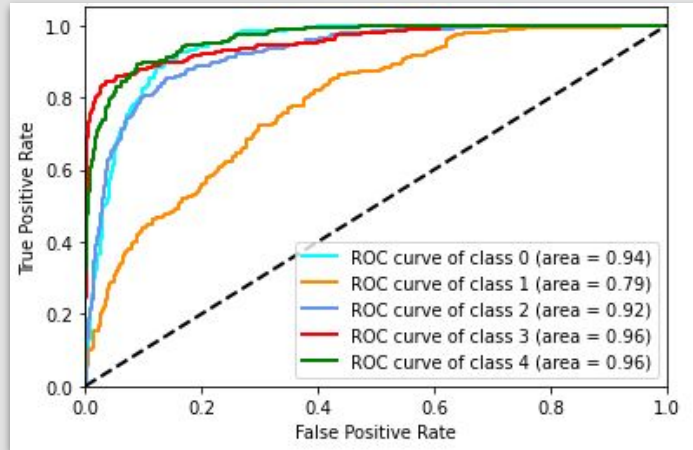
- Sirvan Khalighi , Teresa Sousa, Gabriel Pires, Urbano Nunes. *Automatic sleep staging: A computer assisted approach for optimal combination of features and polysomnographic channels.*
- Sirvan Khalighi, Teresa Sousa, José Moutinho Santos, Urbano Nunes. *ISRUC-Sleep: A comprehensive public dataset for sleep researchers.*

*Presentación hecha como parte de la evaluación  
del Máster en Ciencia de Datos (UV)*

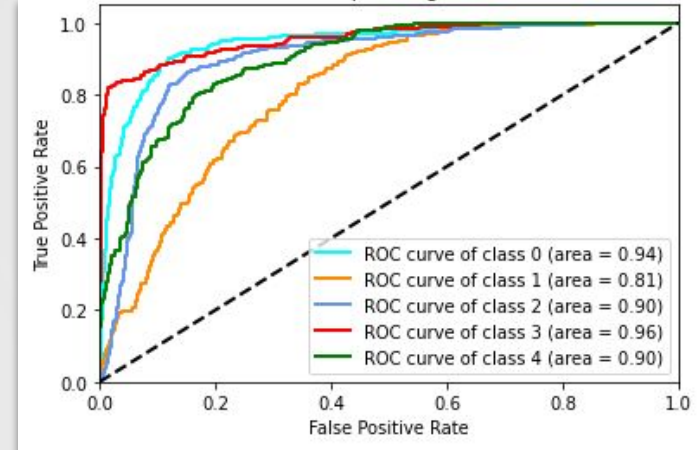


# Resultados en test

## Modelo con todas las características



## Modelo con todas las características reducidas



- Sorprende que los resultados en test mejoran.
- Seguramente porque éstos sean más claros para el modelo en este caso.

	Kappa_kohen_train	AUC_train	Kappa_kohen_test	AUC_test
Modelo				
RF	0.387963	0.767444	0.634378	0.914172
RF_carac_reducidas	0.355220	0.799261	0.558272	0.900522