

# Reconocimiento sonoro de instrumentos musicales



**Universidad**  
Internacional  
de Valencia

**Por: William Steve Rodriguez Villamizar**

# Objetivos

- Seleccionar un dataset adecuado para el problema.
- Entrenar un modelo de IA para clasificación de instrumentos en un audio.
- Aplicar una serie de transformaciones sobre los audios para adecuarlos para el modelo de IA.

# Estado del arte

- Aurora Salgado, Reconocimiento automático de instrumentos mediante aprendizaje máquina, 2019
- Towardsdatascience, Extracción de características
- Silvia Jiménez Gómez, Generación y evaluación de secuencias melódicas mediante inteligencia artificial, 2018
- Manuel Enrique Aldana Sánchez y Juan Sebastian Perdomo Méndez, Reconocimiento de partituras musicales por medio de visión artificial, 2013

# Dataset

- John Thickstun, Zaid Harchaoui y Sham Kakade tienen su dataset de musinet en kaggle



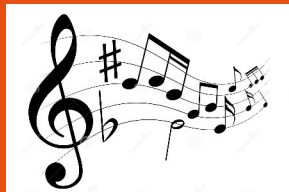
330 Archivos



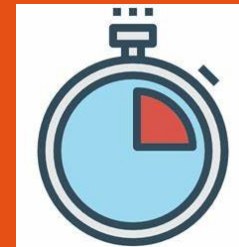
+1M etiquetas  
Instrumentos



33,5GB

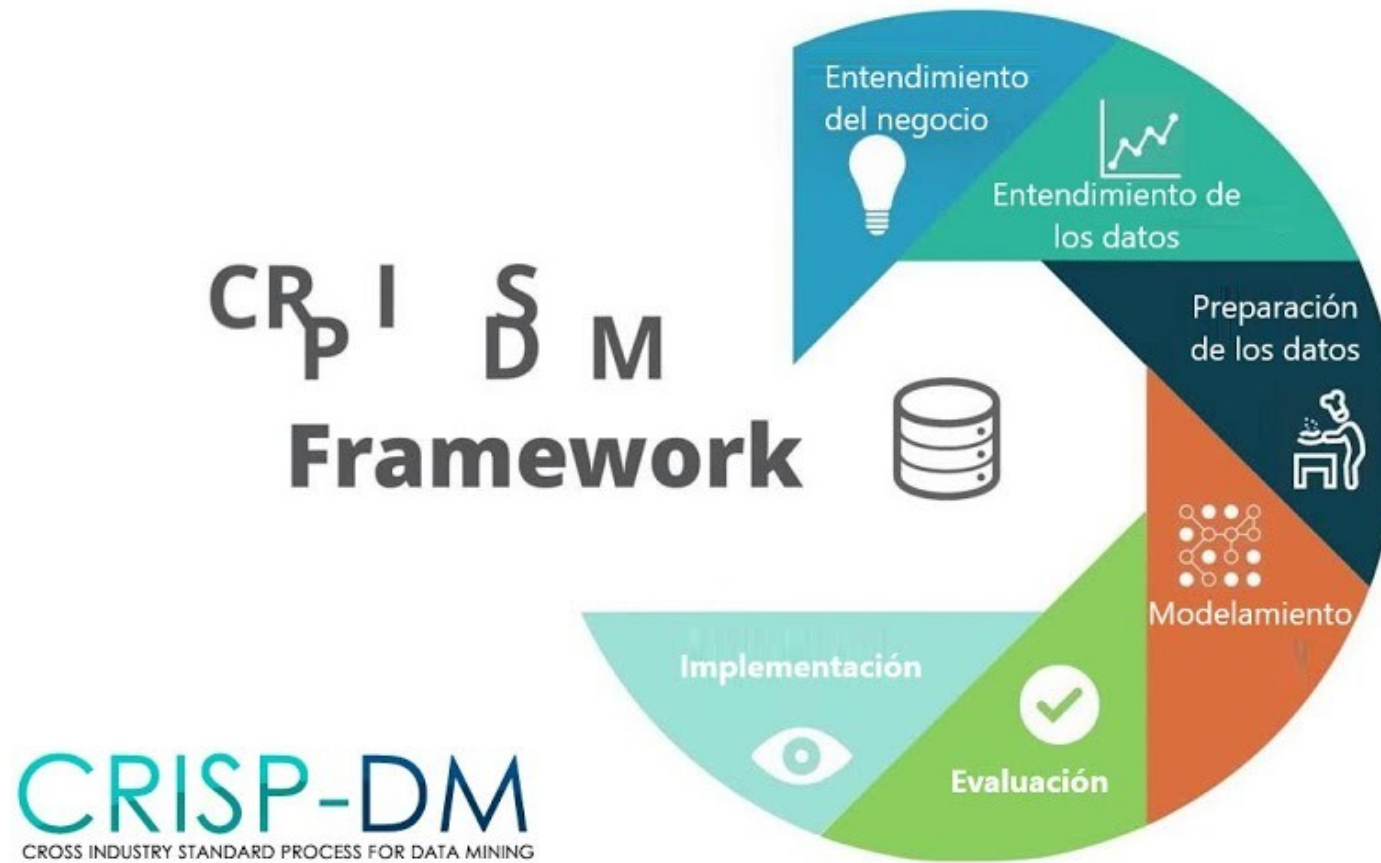


10 compositores



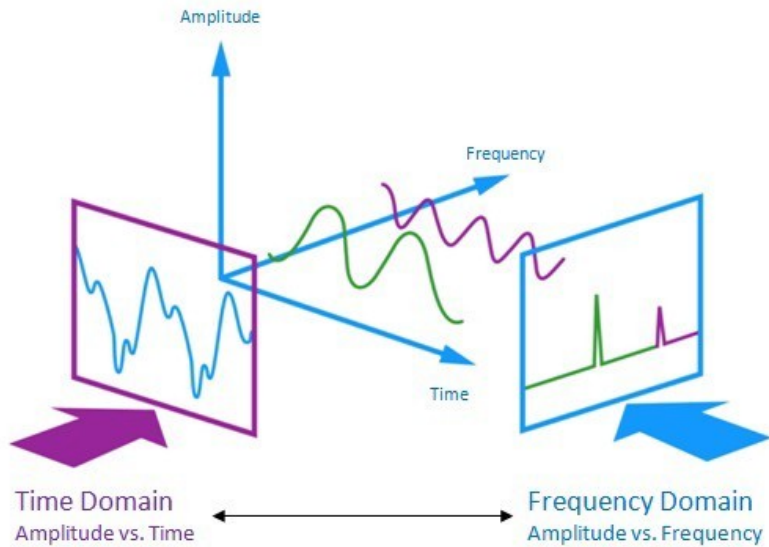
122.709  
segundos

# Metodología a aplicar

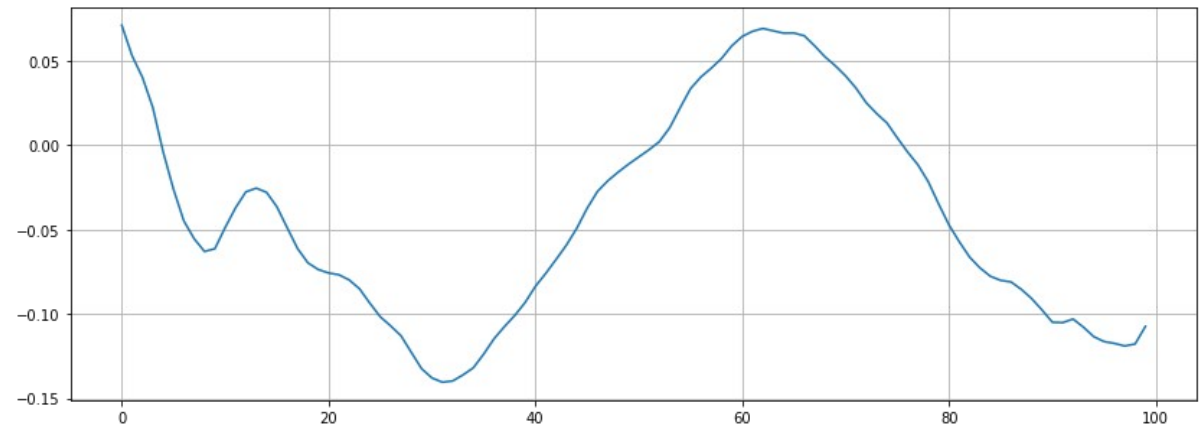


# Preparación de los datos

- Extracción de características



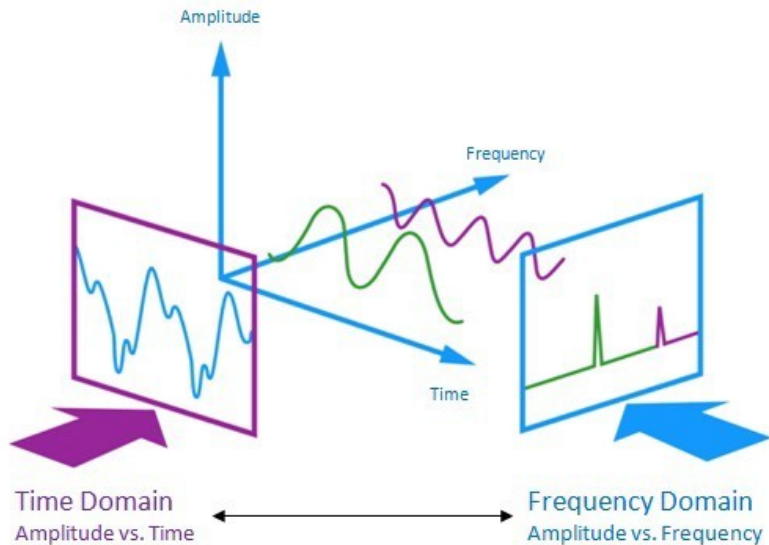
- Cruce por cero



# Preparación de los datos

- Extracción de características

- Cruce por cero
- Centroide espectral



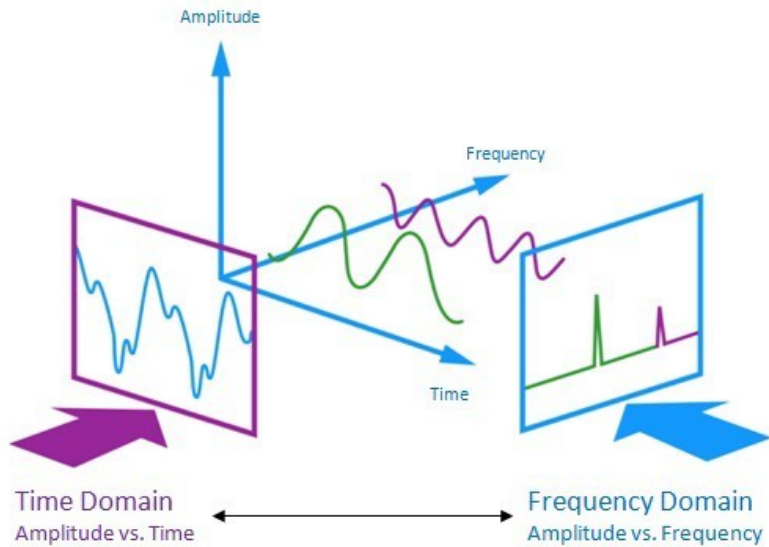
$$Media(X) = \frac{\sum_j X_j}{N}$$

Frecuencias

# Preparación de los datos

- Extracción de características

- Cruce por cero
- Centroide espectral
- Reducción espectral

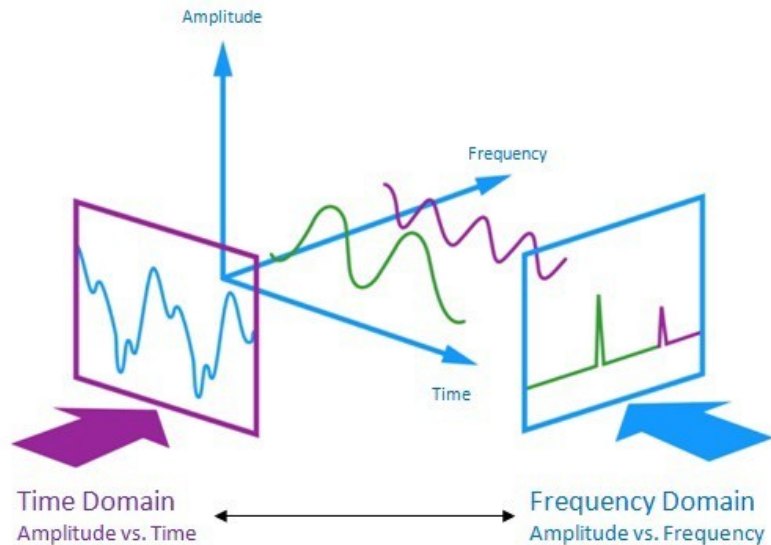


Amplitud en primera frecuencia encontrada  
del centroide espectral



# Preparación de los datos

- Extracción de características



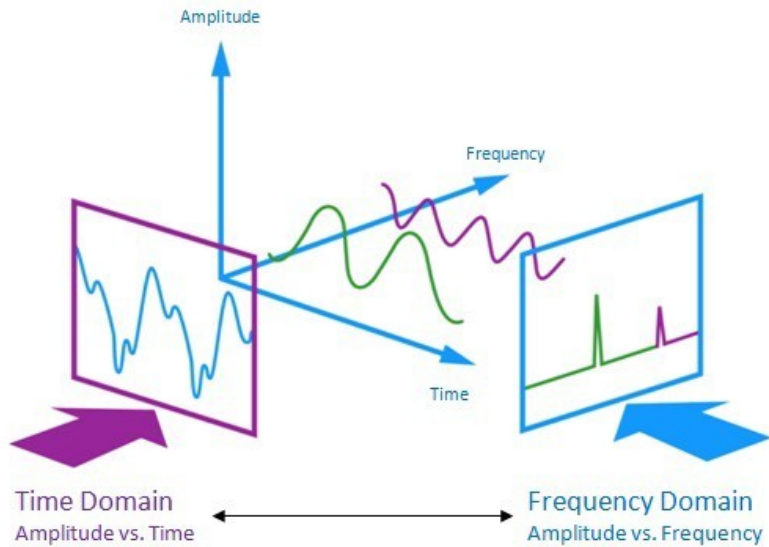
- Cruce por cero
- Centroide espectral
- Reducción espectral
- RMS

$$\sqrt{\text{Media}(X) = \frac{\sum_j X_j}{N}}$$

Amplitud

# Preparación de los datos

- Extracción de características

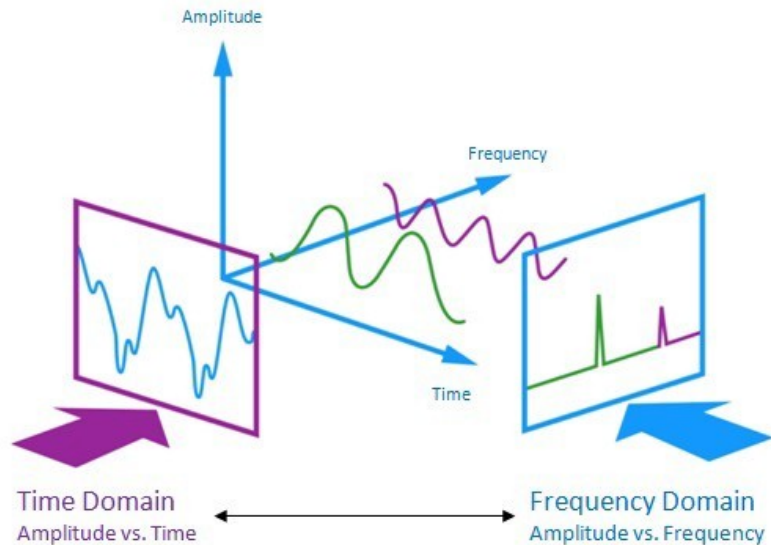


- Cruce por cero
- Centroide espectral
- Reducción espectral
- RMS
- Croma

Promedios de los cromagramas

# Preparación de los datos

- Extracción de características

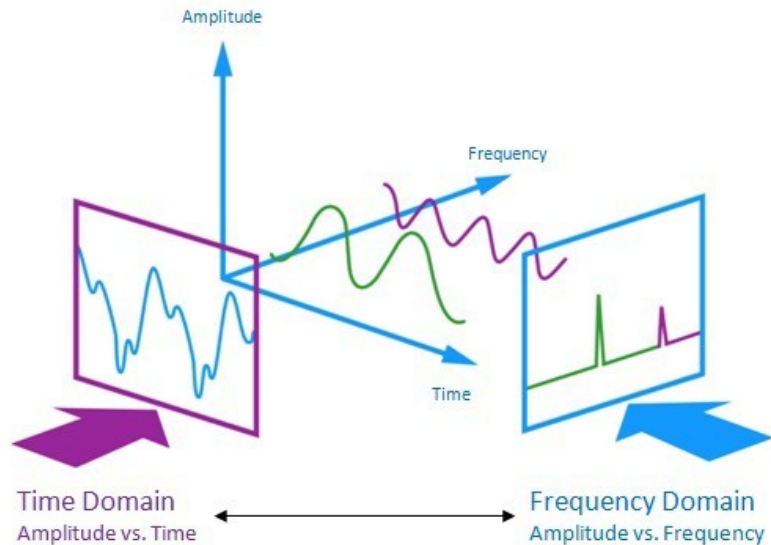


- Cruce por cero
- Centroide espectral
- Reducción espectral
- RMS
- Croma
- Ancho de banda espectral

Promedios de la longitud de frecuencias  
donde esta la mayor potencia de la señal

# Preparación de los datos

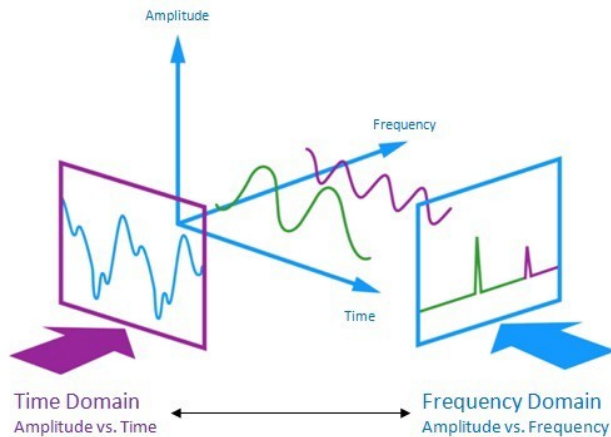
- Extracción de características



- Cruce por cero
- Centroide espectral
- Reducción espectral
- RMS
- Croma
- Ancho de banda espectral
- MFCC — Coeficientes centrales de frecuencia Mel

# Preparación de los datos

- Extracción de características

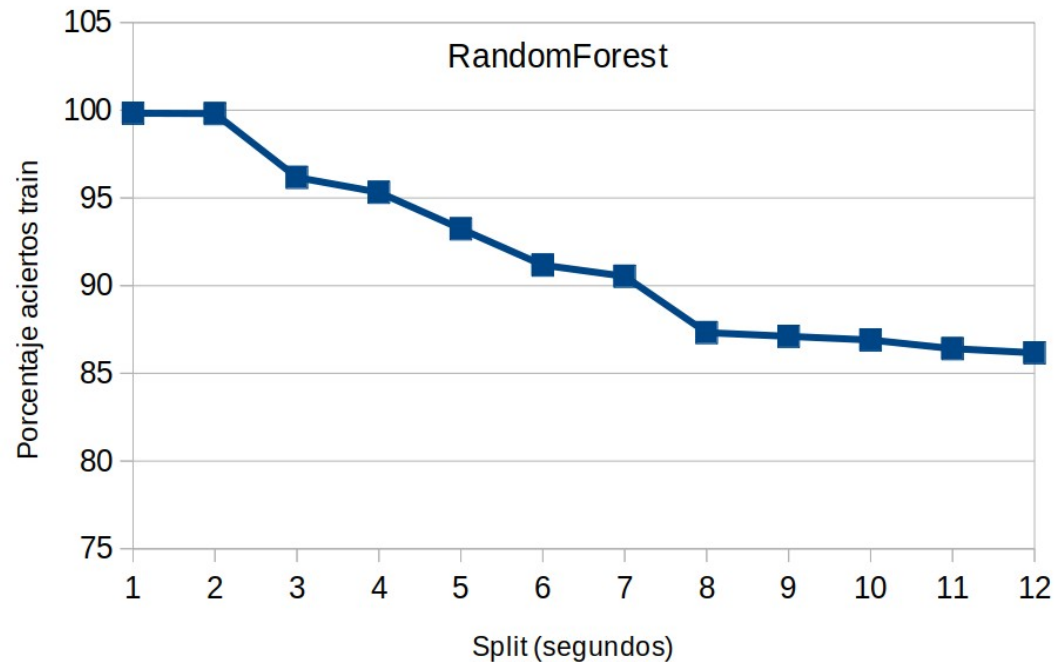


En resumen

croma	rms	centroide espectral	ancho banda	reducción espectral	cruces por cero	MFCC
1 valor	1 valor	1 valor	1 valor	1 valor	1 valor	20 valores

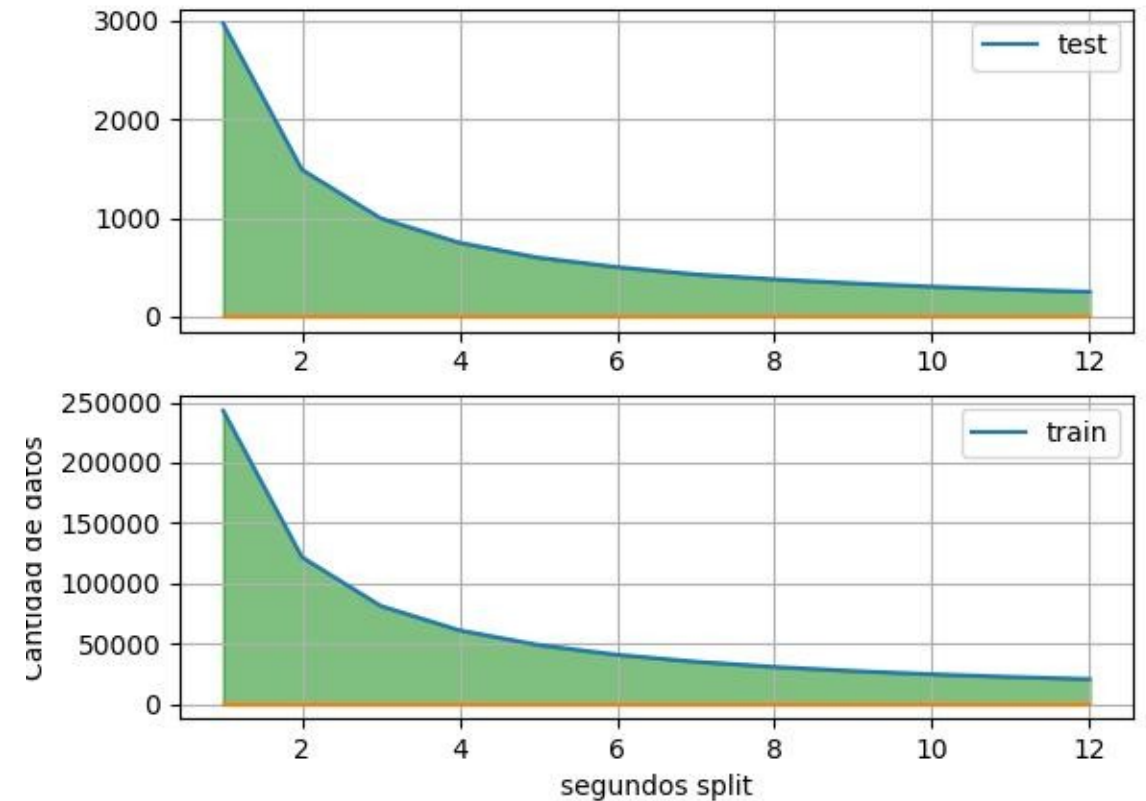
# Preparación de los datos

- Split a 2 segundos



Con split de 2 segundos, el tamaño de cada dato es un vector de 88.200 posiciones.

Cantidad de datos segun el split elegido



Antes: train: 330, test 10  
Después (split 2 segundos): train:121.700, test: 1.488

# Preparación de los datos

Datos luego del split

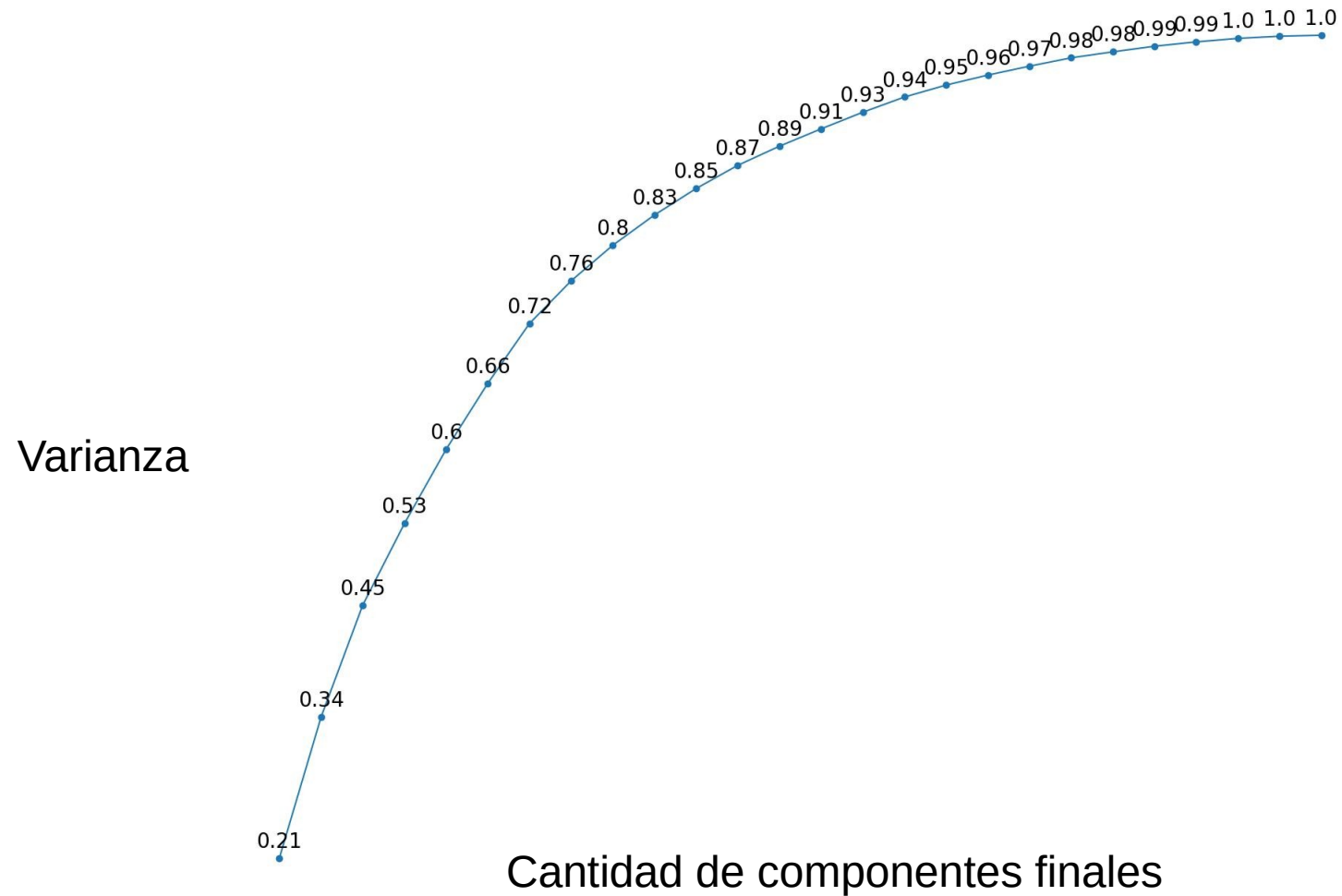
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	.	.	.	88196	88197	88198	88199	88200
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	-------	-------	-------	-------	-------

Datos luego de la extracción de características

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

# Metodología a aplicar

- PCA





# Preparación de los datos

Datos luego del split

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	.	.	.	88196	88197	88198	88199	88200
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	-------	-------	-------	-------	-------

Datos luego de la extracción de características

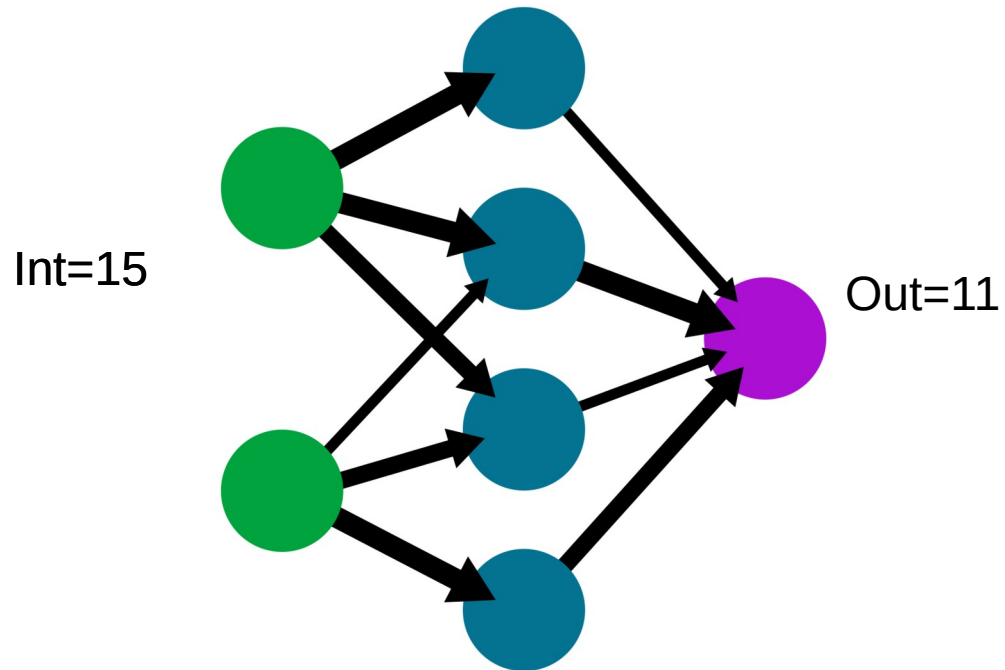
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Datos luego del PCA (93%)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----

# Metodología a aplicar

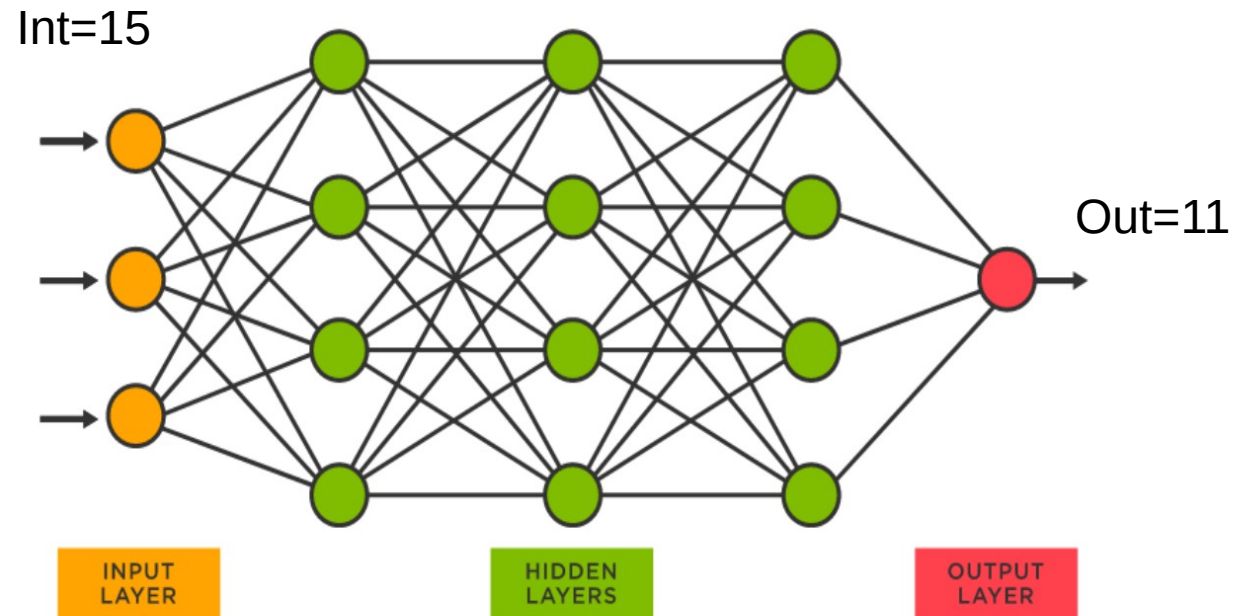
- Modelado RNA simple



train	valid	test	tiempo respuesta (seg)
90,36%	90,27%	90,09%	0,06

# Metodología a aplicar

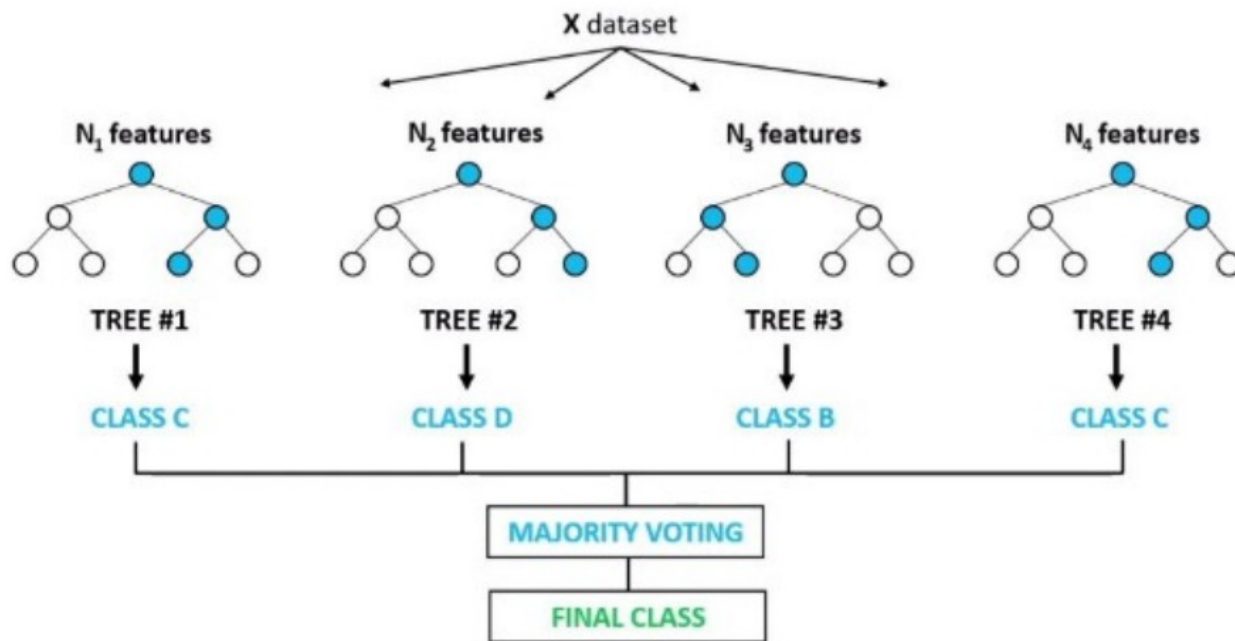
- Modelado RNA Experimental



train	valid	test	tiempo respuesta (seg)
78,63%	78,72%	80,09%	0,32

# Metodología a aplicar

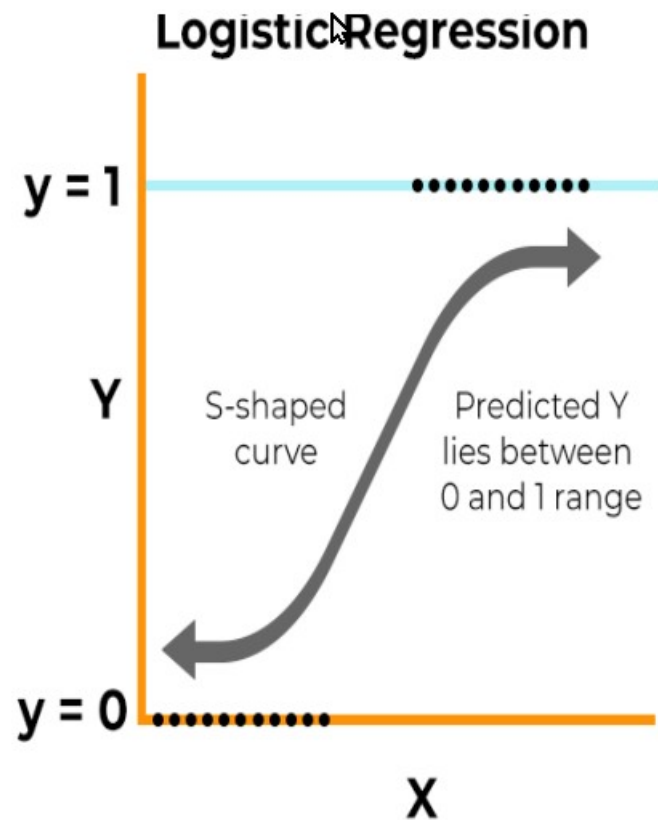
- Modelado RandomForestClassifier



train	valid	test	tiempo respuesta (seg)
99,81%	95,72%	94,90%	0,79

# Metodología a aplicar

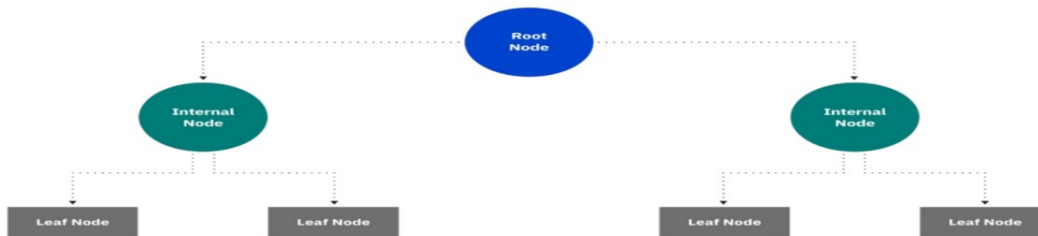
- Modelado LogisticRegression



train	valid	test	tiempo respuesta (seg)
92,18%	92,36%	90,72%	1,86

# Metodología a aplicar

- Modelado DecisionTreeClassifier



train	valid	test	tiempo respuesta (seg)
100,00%	92,54%	91,45%	1,15

# Metodología a aplicar

- Modelado DummyClassifier

		Predicted Class	
		Negative	Positive
Actual Class	Negative	True Negative (TN) 912	False Positive (FP) 0
	Positive	False Negative (FN) 102	True Positive (TP) 0

Predict negative class for every instance.

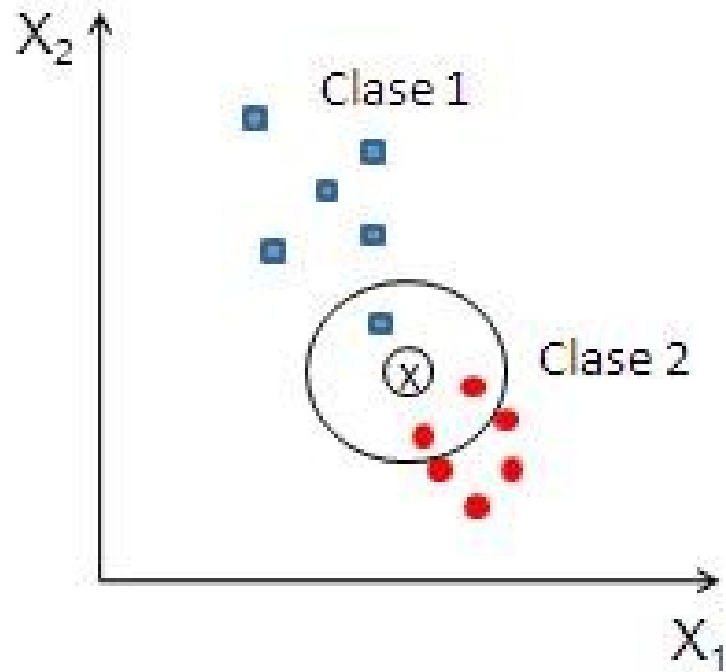
Calculate accuracy.

train	valid	test	tiempo respuesta (seg)
85,54%	85,63%	81,54%	3,1

$$\text{Accuracy} = \frac{0 + 912}{0 + 102 + 0 + 912} = 89.94\%$$

# Metodología a aplicar

- Modelado KNeighborsClassifier



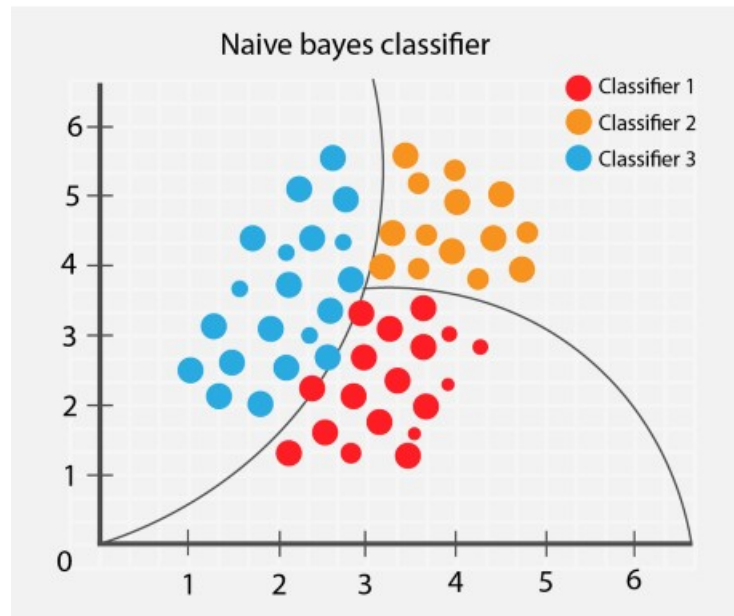
train	valid	test	tiempo respuesta (seg)
96,63%	96,72%	96,00%	2,3



# Metodología a aplicar

- Modelado GaussianNB

Aplicando el teorema de Bayes (probabilística) con una fuerte (Naive) independencia de las características



train	valid	test	tiempo respuesta (seg)
90,63%	90,81%	89,00%	1,82

# Metodología a aplicar

- Resumen modelado

Modelos		Algoritmo	train	valid	test	tiempo all test (seg)
Machine learning	Clásicos	RandomForestClassifier	99,81%	95,72%	94,90%	0,79
		LogisticRegression	92,18%	92,36%	90,72%	1,86
		DecisionTreeClassifier	100,00%	92,54%	91,45%	1,15
		DummyClassifier	85,54%	85,63%	81,54%	3,1
		KNeighborsClassifier	96,63%	96,72%	96,00%	2,3
		GaussianNB	90,63%	90,81%	89,00%	1,82
	Deep Learning	Basico 122.000 parametros	90,36%	90,27%	90,09%	0,06
		Experimental	78,63%	78,72%	80,09%	0,32

# Metodología a aplicar

- Elección modelo usar

**se elije el randomForest, principalmente por sus porcentajes superiores al 95%** en validación y casi el 95% en el testeo, si bien el KNeighborsClassifier tiene mejores métricas, su tiempo de predicción es muy alto lo que lo hace ineficiente para un entorno de producción. No se elije la red neuronal básica, pues si bien otorga buenas métricas y gran velocidad de respuesta, no son tan altas las métricas como lo es con RandomForest, pero si el tema tiempo de respuesta es un factor clave se podría optar por usar la red neuronal básica).

# Conclusiones

- Las redes neuronales no siempre son la mejor opción para todos los problemas.
- Limpiar y preparar los datos es sumamente importante
- El PCA ayuda considerablemente a mejorar el aprendizaje del modelo
- Los algoritmos clásicos dieron mejor resultado de lo esperado.

Gracias