



# Spike Challenge

Diseño e implementación de un clasificador  
de tipo de música

Sebastián Camberos  
Febrero 2018



# Descripción Spike Challenge



- Objetivo: clasificar 2 tipos de música a partir de descriptores varios
- Libre elección de algoritmo de clasificación
- Origen datos: API de Spotify
- Muestra (sin procesar)
  - Entrenamiento, música tipo reggaeton, 70 registros x 15 campos
  - Entrenamiento, otros tipos de música, 2.230 registros x 16 campos
  - Validación, 50 registros x 16 campos



# Pregunta 1 Análisis Datasets

Variable	Composición	Estilo	Estructura	Estacional	Redundancia
acousticness	X				
danceability		X			
duration			X		
energy		X			
id_new				X	
instrumentalness	X				
key	X				
liveness			X		
loudness					X
mode					X
popularity				X	
speechiness		X			
tempo	X				
valence		X			

- Criterios selección variables
  - Descripción variable Spotify vs elementos de composición y estilo musical (1)
  - Corroboración diferencias estilo musical con distribuciones (promedio, desviación estándar y kurtosis)
  - Eliminación descriptores que no alteran estructura musical
  - Eliminación variables estacionales
  - Eliminación redundancias según correlación o composición musical

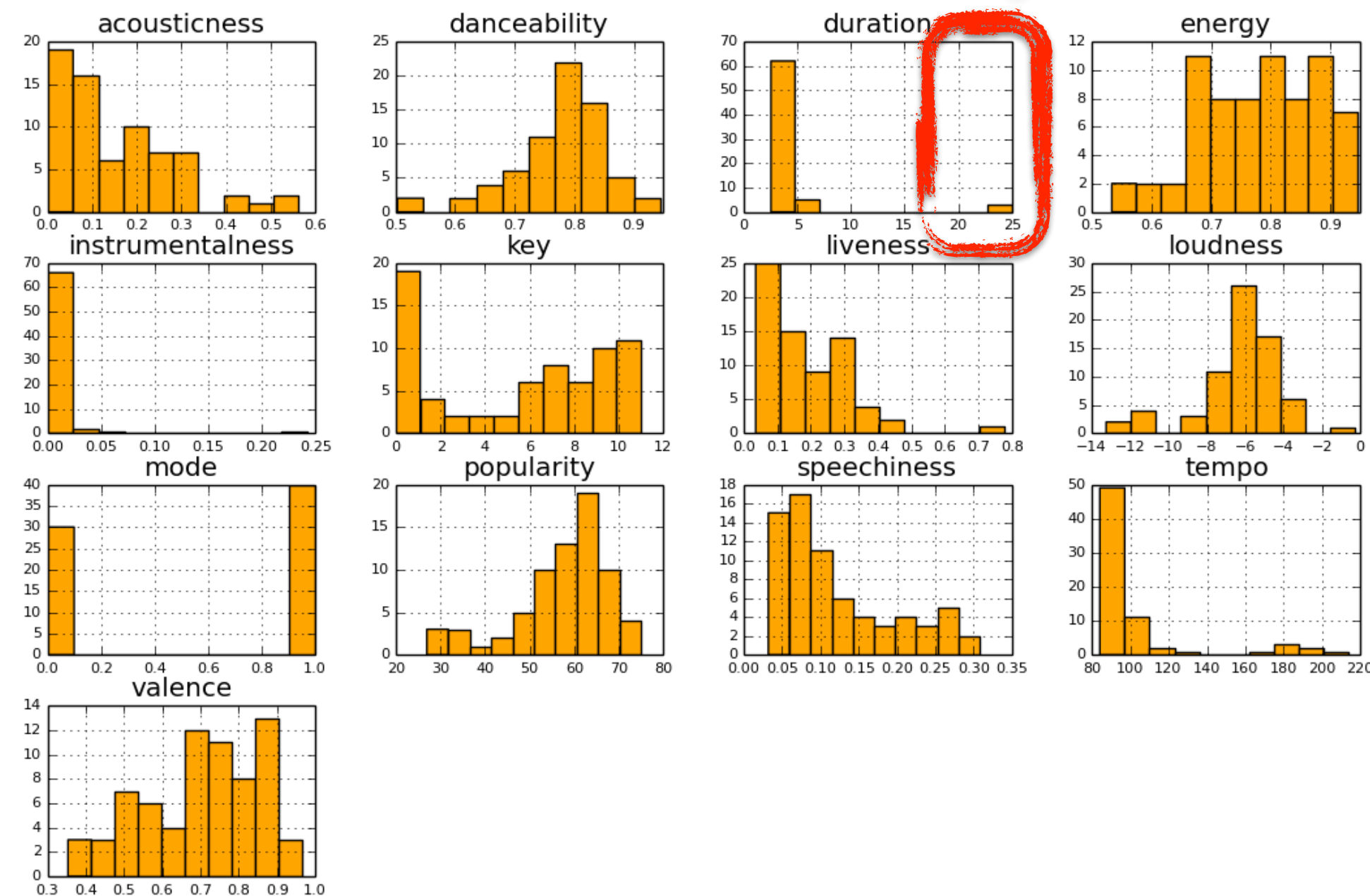
(1) Fuente descripción: <https://beta.developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/>



# Pregunta 1 Histogramas

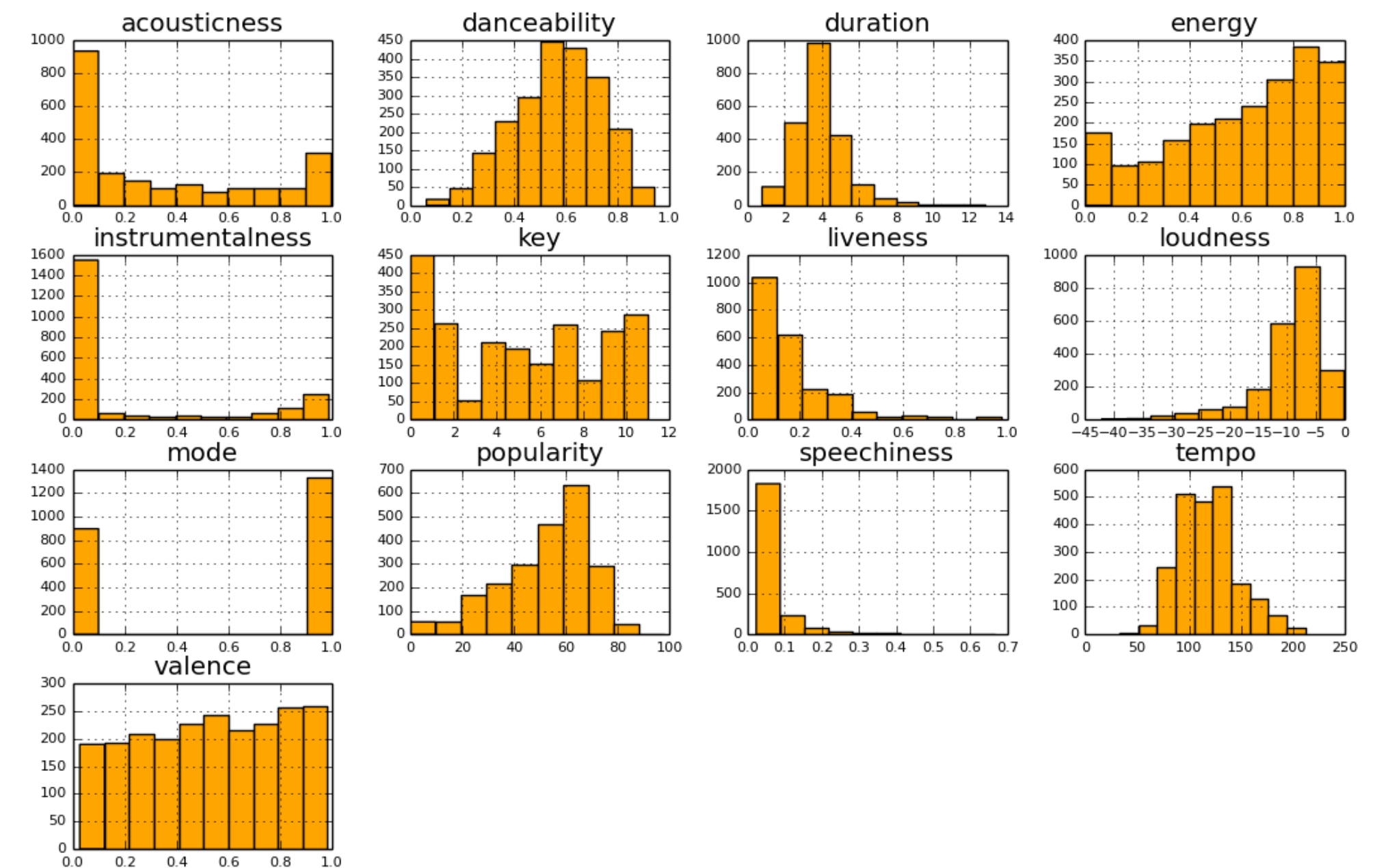
## Música tipo reggaeton

Posibles outliers



Muestra: data\_reggaeton.csv.

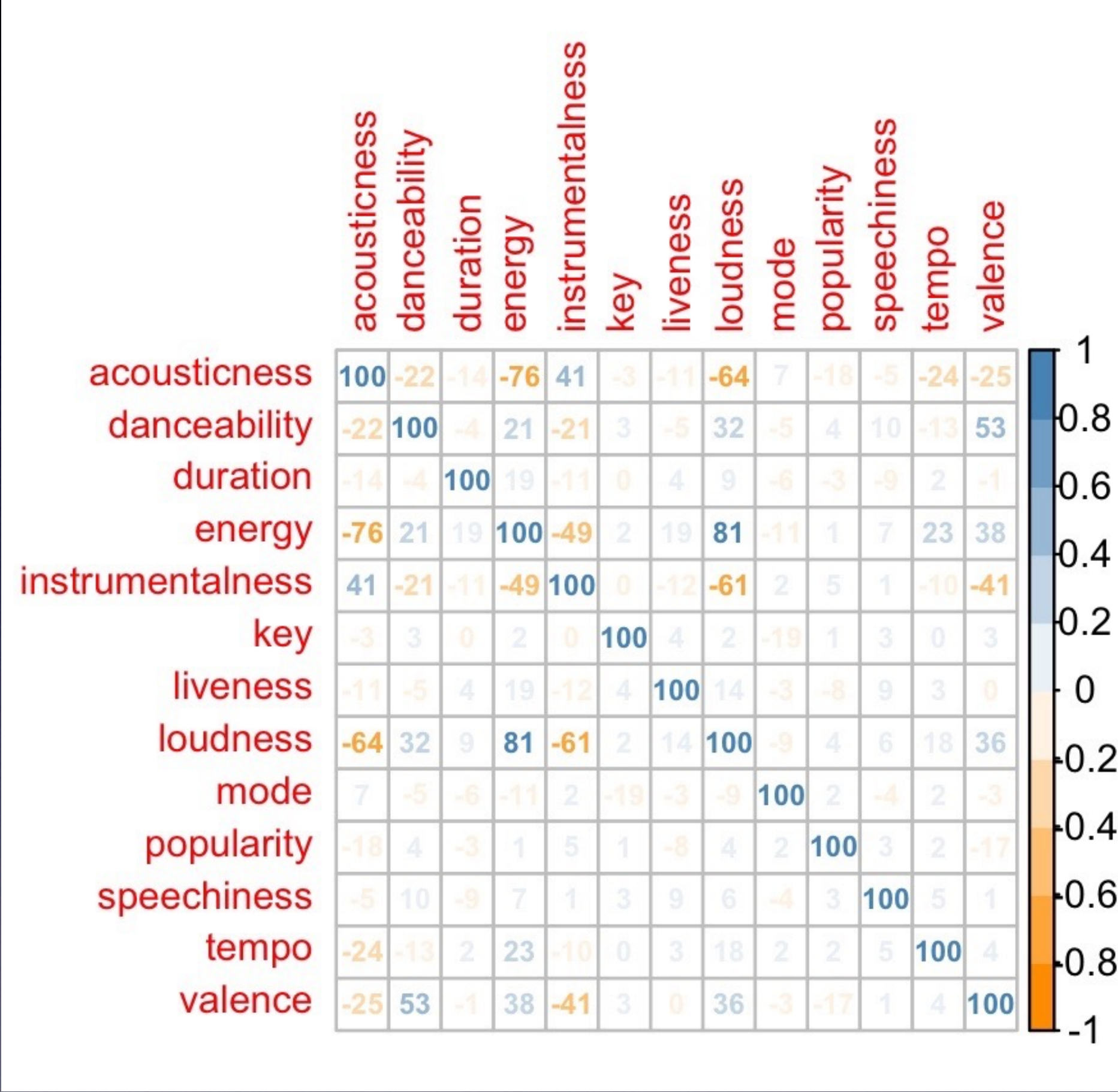
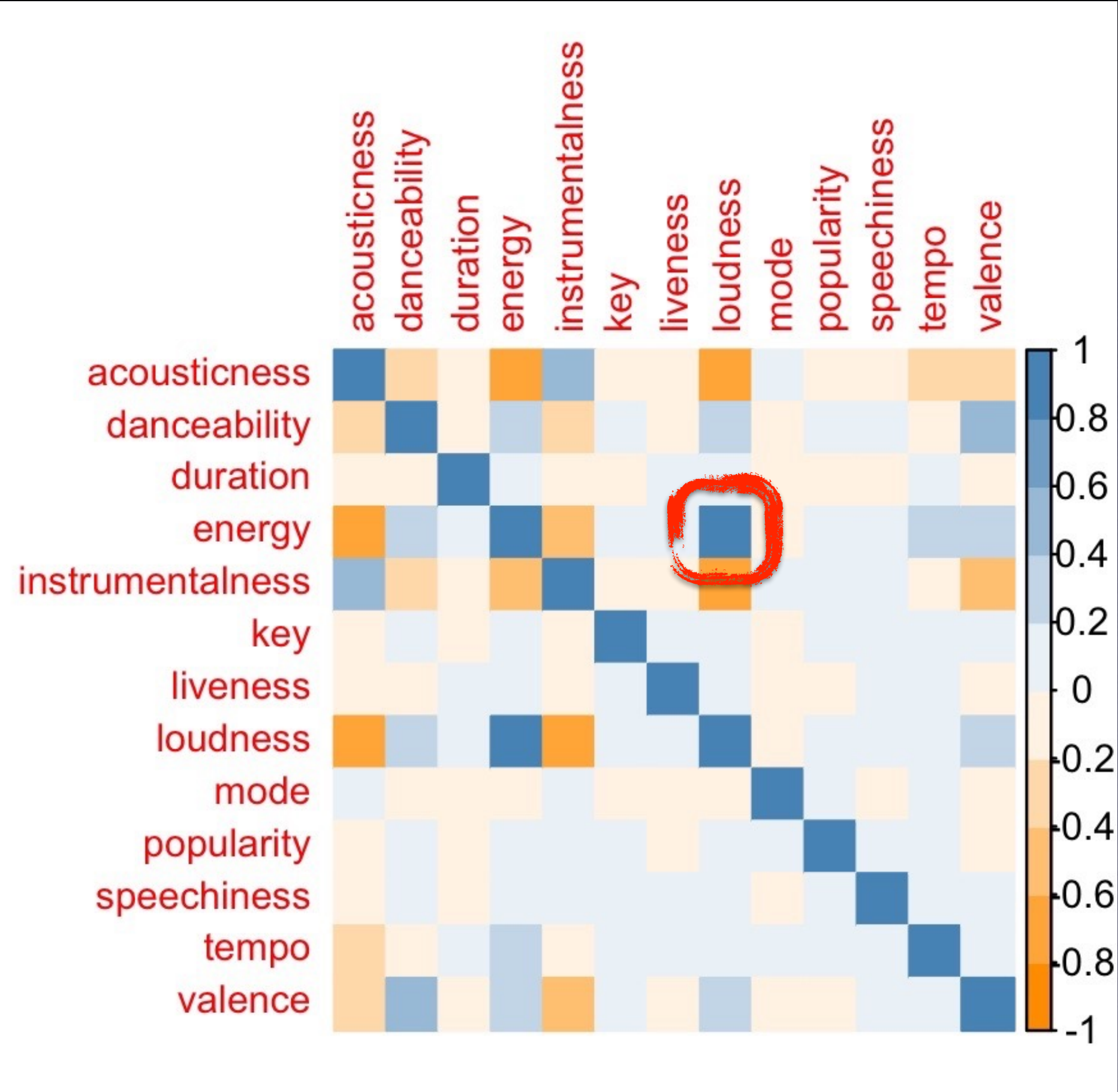
## Otros tipos de música



Muestra: data\_todotipo.csv, excluye registros vacíos.



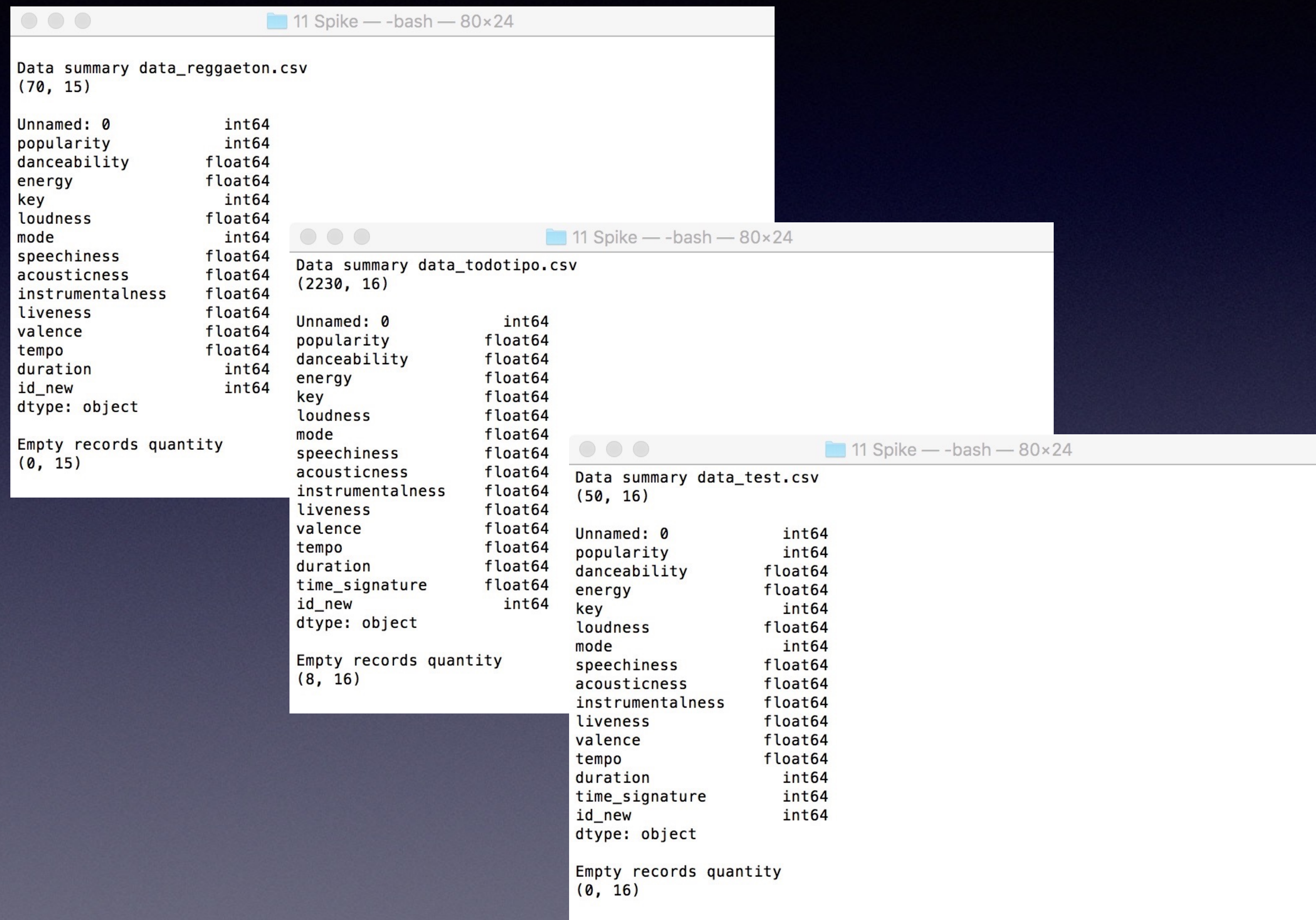
# Pregunta 1 Correlaciones



Muestra: data\_train.csv, consolidado de reggaeton y todotipo, excluye registros vacíos.



# Pregunta 2 Consolidación Datasets



```
11 Spike — -bash — 80x24
Data summary data_reggaeton.csv
(70, 15)

Unnamed: 0      int64
popularity      int64
danceability    float64
energy          float64
key            int64
loudness        float64
mode            int64
speechiness     float64
acousticness    float64
instrumentalness float64
liveness        float64
valence         float64
tempo           float64
duration        int64
id_new          int64
dtype: object

Empty records quantity
(0, 15)

11 Spike — -bash — 80x24
Data summary data_todotipo.csv
(2230, 16)

Unnamed: 0      int64
popularity      float64
danceability     float64
energy          float64
key             float64
loudness         float64
mode            float64
speechiness      float64
acousticness     float64
instrumentalness float64
liveness         float64
valence          float64
tempo           float64
duration         float64
time_signature   float64
id_new          int64
dtype: object

Empty records quantity
(8, 16)

11 Spike — -bash — 80x24
Data summary data_test.csv
(50, 16)

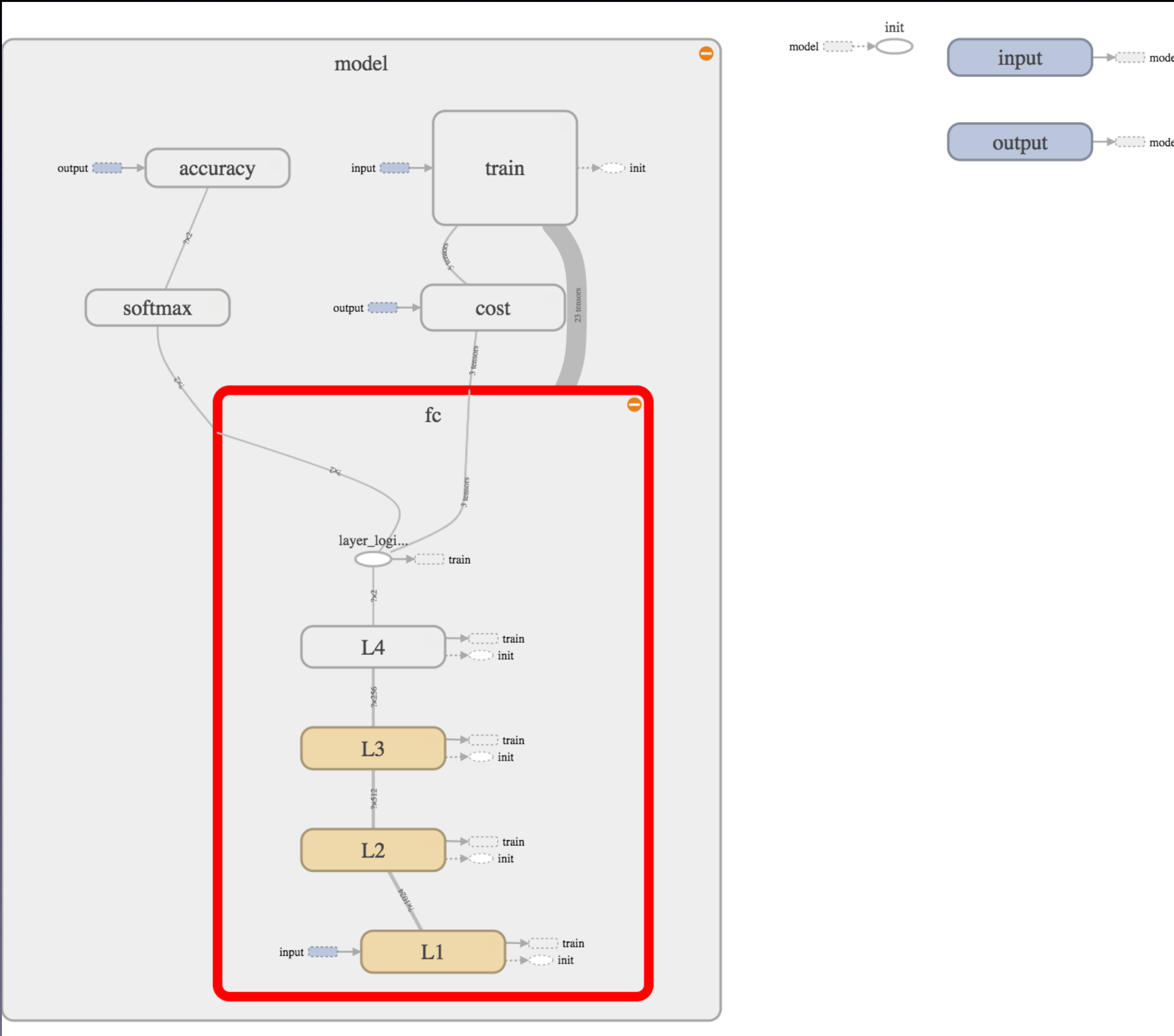
Unnamed: 0      int64
popularity      int64
danceability     float64
energy          float64
key            int64
loudness        float64
mode            int64
speechiness      float64
acousticness     float64
instrumentalness float64
liveness         float64
valence          float64
tempo           float64
duration        int64
time_signature   int64
id_new          int64
dtype: object

Empty records quantity
(0, 16)
```

- Preprocesamiento datos
  - Filtrado de registros vacíos, 8 en total
  - Normalización cantidad de campos
  - Marcas identificación tipo de música: “reggaeton” y “otros”
  - Transformación int a float
  - Consolidación muestra reggaeton y otros tipos de música, 2.292 registros x 15 campos
- Preentrenamiento modelo
  - Extracción media y normalización
  - Hot-encode: “reggaeton”, 01; “otros”, 10



# Pregunta 3 Modelo



- Descripción modelo
  - Basado en clasificador de dígitos (LeCun, 1998) y autoencoder (Hinton, 2006)
  - Red neuronal fully connected
  - 4 hidden layers con pesos 1024 - 512 - 256 - 2
  - Función activación ReLU
  - Salida softmax interpretada como probabilidad
  - Código en Python y TensorFlow
- Calibración hiper parámetros
  - Pesos iniciales aleatorios
  - Optimizador Adam
  - Entrenamiento por batches



# Pregunta 4 Performance Modelo

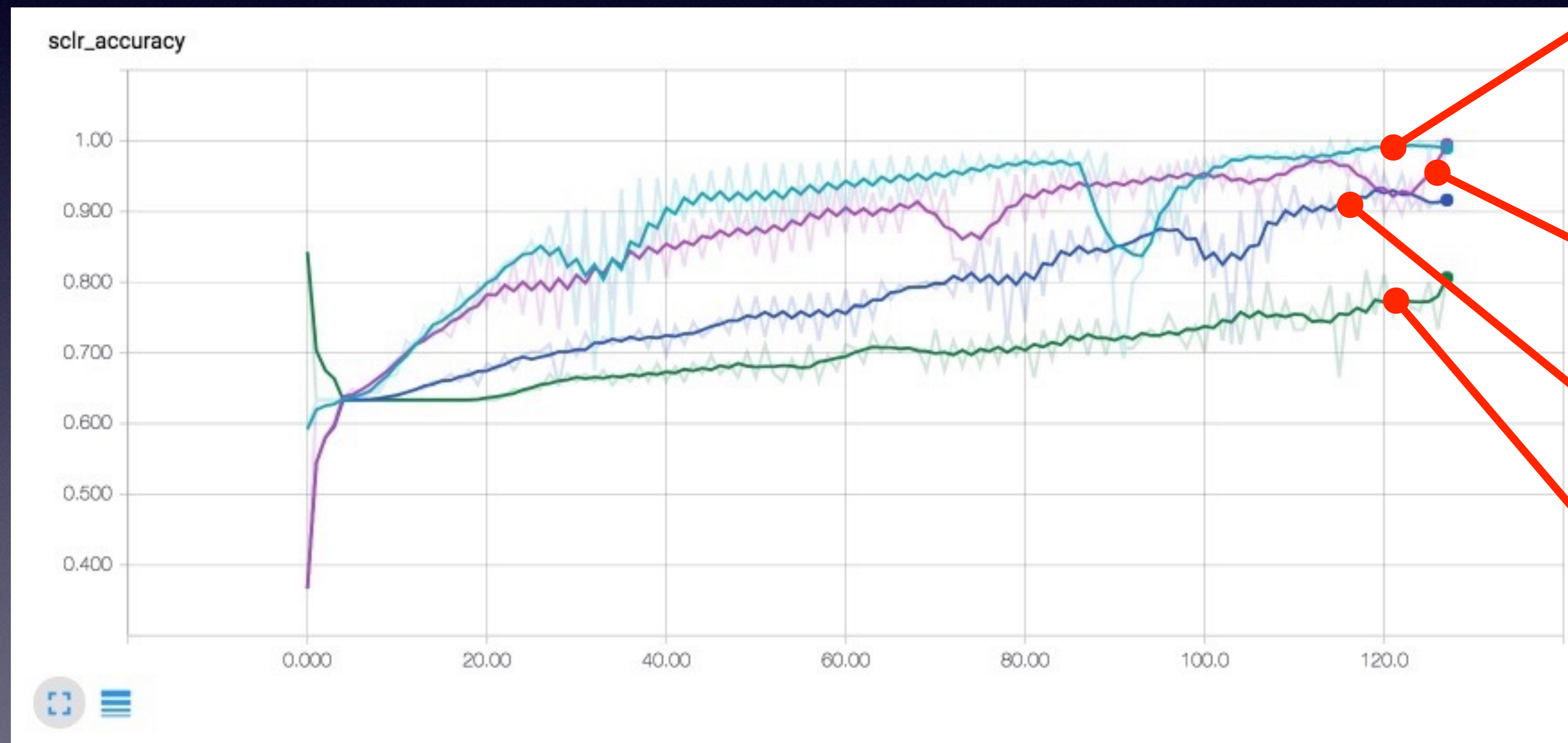
		Modelo 1	Modelo 2	Modelo 3	Modelo 4
Variable	acousticness	X			
	danceability	X	X	X	X
duration					
energy		X	X	X	X
id_new					
instrumentalness		X	X	X	X
key		X	X	X	X
liveness					
loudness					
mode					
popularity					
speechiness		X	X	X	
tempo		X	X		
valence		X	X	X	X
Predictores		8	7	6	5
Accuracy	epochs	128	128	128	128
	data_train	0.9978	0.9948	0.9930	0.9747
	data_reggaeton	0.9286	0.8429	0.7857	0.1714

- Observar durante entrenamiento
  - Función de pérdida (cross entropy)
  - Performance (accuracy)
  - Inspección separación tipos de música
- Performance
  - Pruebas iniciales con 4 modelos (búsqueda parsimonia)
  - Inspección visual da cuenta de separación de tipos de música en logits y softmax



# Pregunta 4 Performance modelo

Accuracy de experimento inicial  
@128 epochs



Modelo 1

Modelo 2

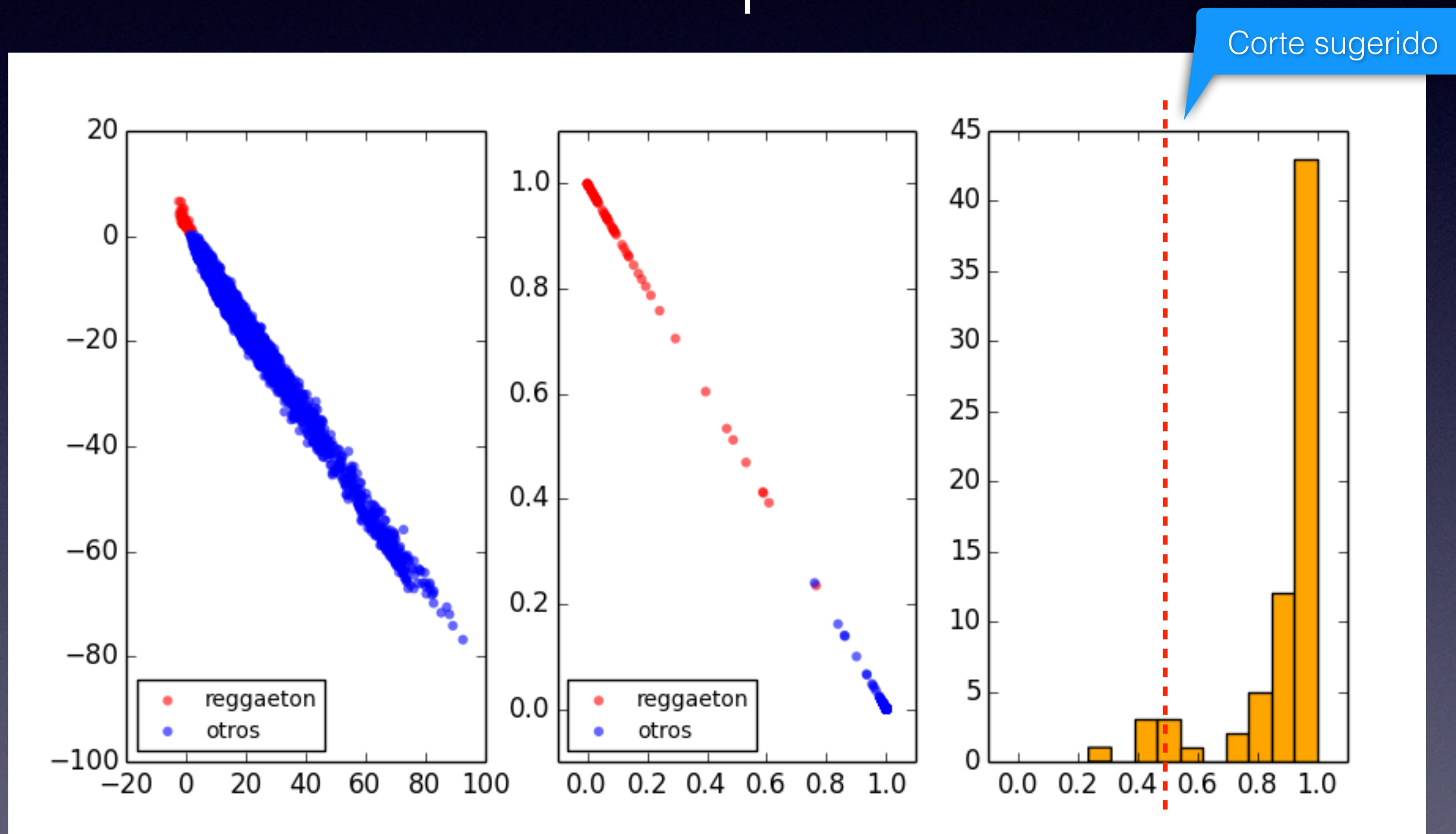
Modelo 3

Modelo 4



# Pregunta 4 Performance modelo

## Separación de datos Modelo 1 @128 epochs

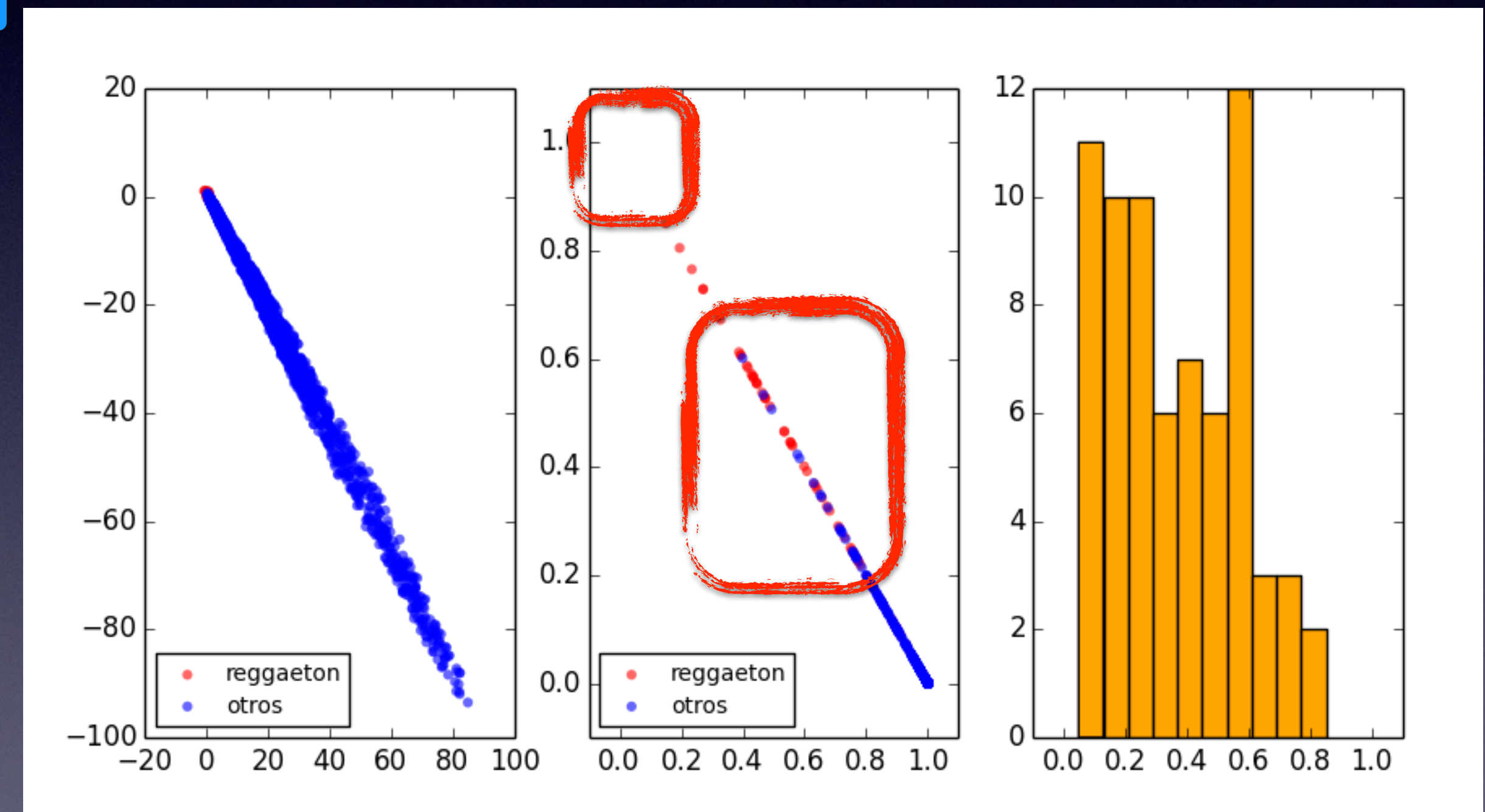


Salida red neuronal  
Layer 4 (logits)

Salida red neuronal  
Softmax (clasificador)

Histograma clasificador  
(solo reggaeton)

## Separación de datos Modelo 4 @128 epochs



Salida red neuronal  
Layer 4 (logits)

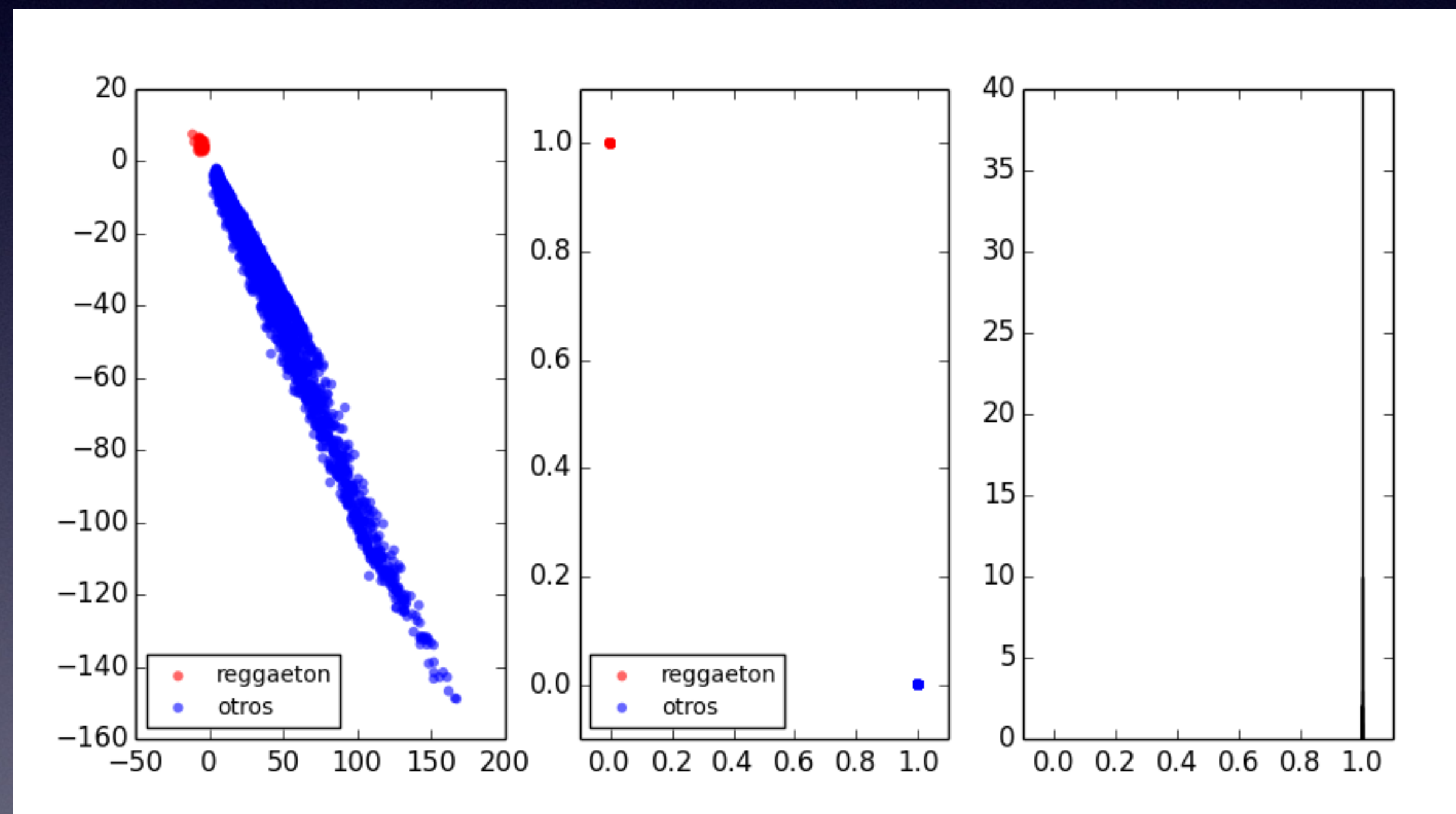
Salida red neuronal  
Softmax (clasificador)

Histograma clasificador  
(solo reggaeton)



# Pregunta 4 Performance modelo

## Separación de datos Modelo 1 @384 epochs



Salida red neuronal  
Layer 4 (logits)

Salida red neuronal  
Softmax (clasificador)

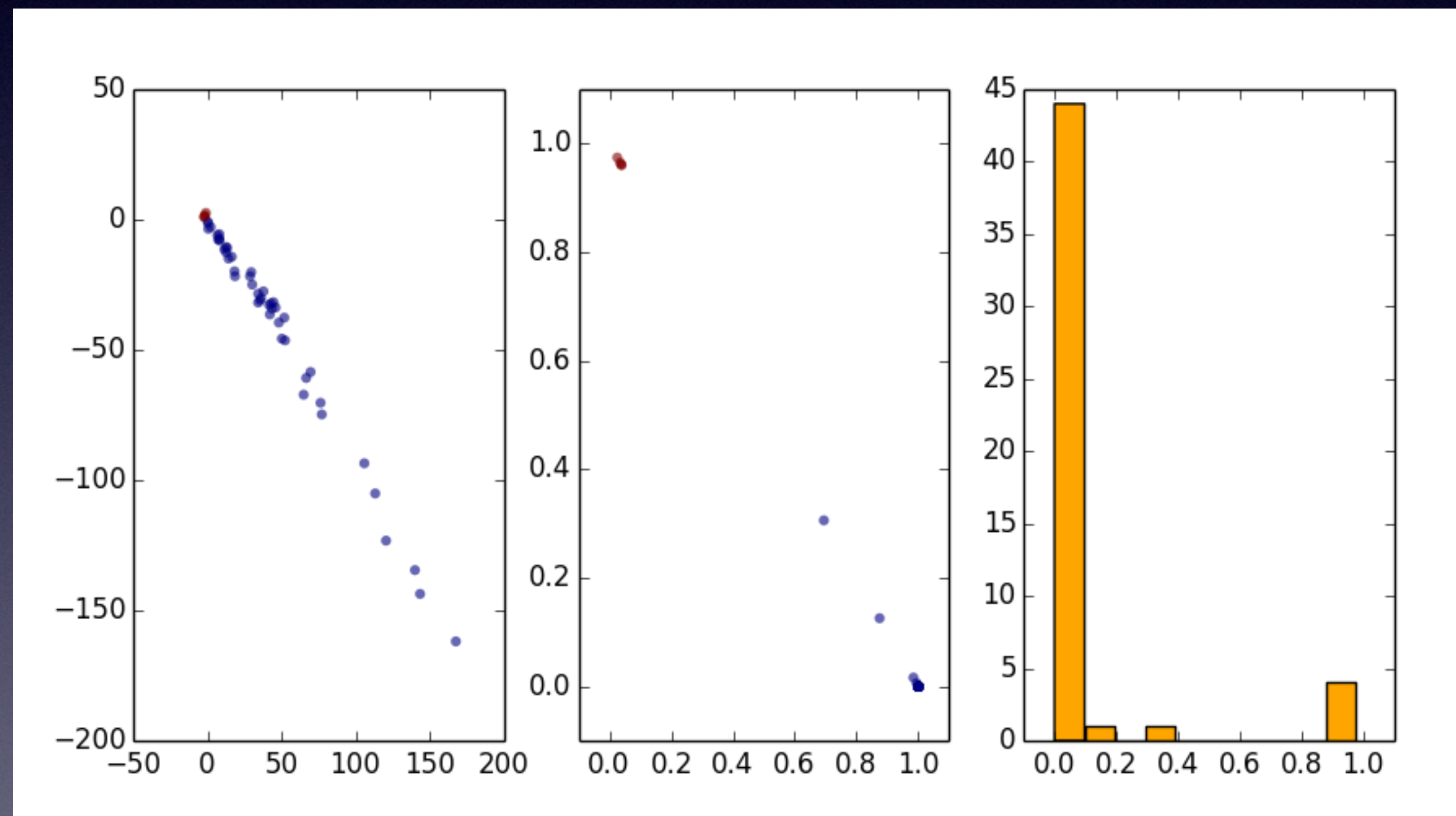
Histograma clasificador  
(solo reggaeton)

- Existen mejoras de optimización y de reducción de consumo de energía
- Incrementar performance
  - Uso de factores y variables puras
  - Aumentar epochs evitando overfitting
- Reducir tiempo de entrenamiento
  - Explorar curriculum learning (Bengio, 2009)
- Reducir tiempo de procesamiento en producción
  - Implementar pruning
  - Reducir descriptores
  - Reducir pesos por unidad neuronal
  - Reducir layers



# Pregunta 5 Validación Modelo

Inferencia Modelo 1  
@384 epochs, 80% nivel de corte



Salida red neuronal  
Layer 4 (logits)

Salida red neuronal  
Softmax (clasificador)

Histograma clasificador  
(solo reggaeton)

- Performance validación
  - Ocorre separación de datos
  - Probabilidad de tipo de música señalada por magnitudes de softmax
  - 4 canciones clasificadas como reggaeton
- Elementos para correcta identificación de tipo de música
  - Arquitectura modelo para descubrir estructura subyacente y lograr amplia separación de datos
  - Predictores elegidos
  - Nivel de corte obtenido durante entrenamiento