Clasificación de géneros musicales

Integrantes
Alberto Olvera Trejo
Ricardo Bernabé Nicolás
Miguel Ángel Romero González

CONTENIDO

Motivación

A lo largo de este curso aprendimos a usar varias herramientas para clasificar datos y hacer predicciones.

Nosotros decidimos aplicar estos conocimientos para ver si se puede lograr clasificar géneros musicales

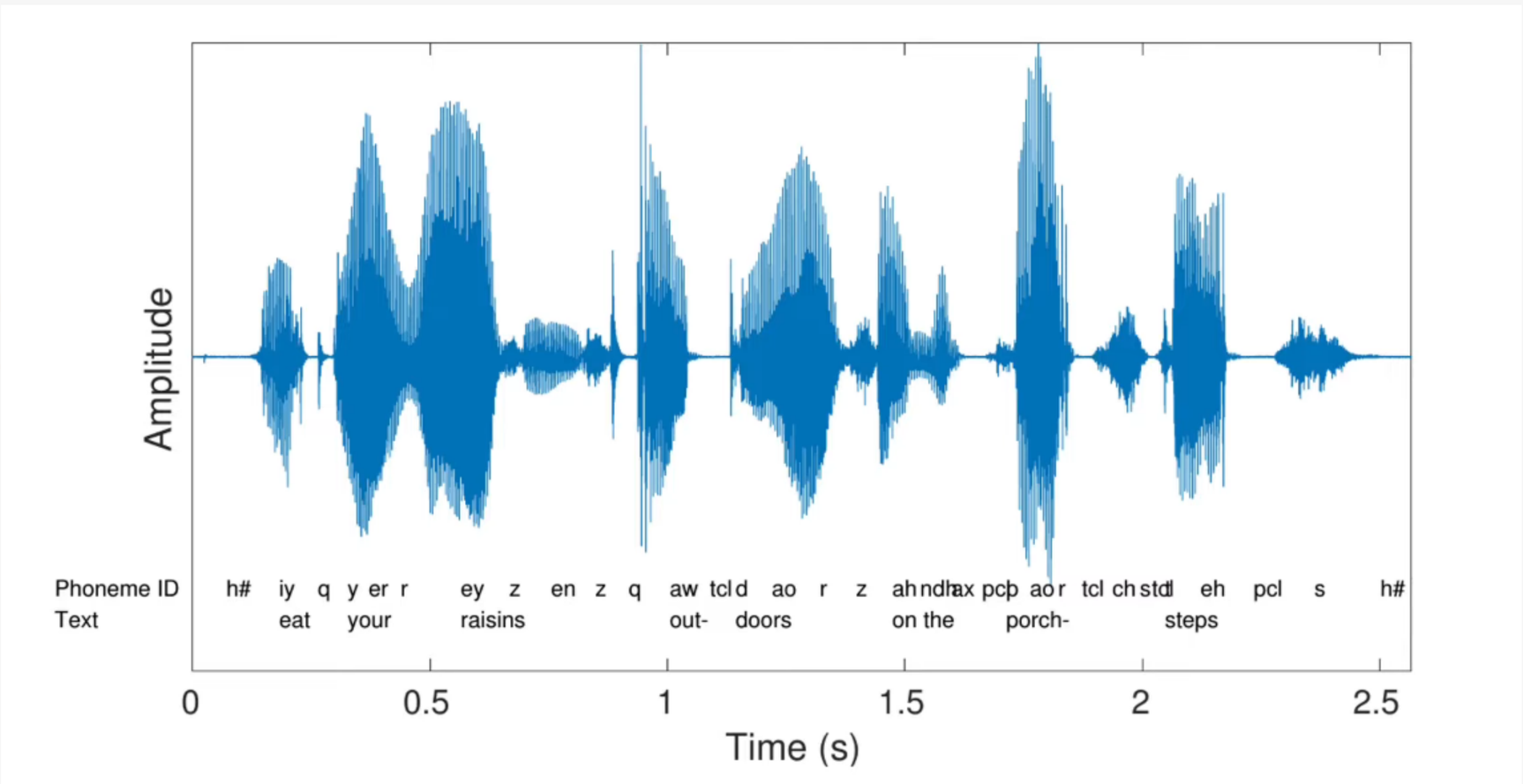
Introducción

Si nosotros oímos una pieza musical, dependiendo de nuestra experiencia y que tan abiertos nuestros horizontes musicales, quizá logremos identificar a qué género pertenece.

Para que una computadora pueda analizarla, debemos de analizarla de manera especial.

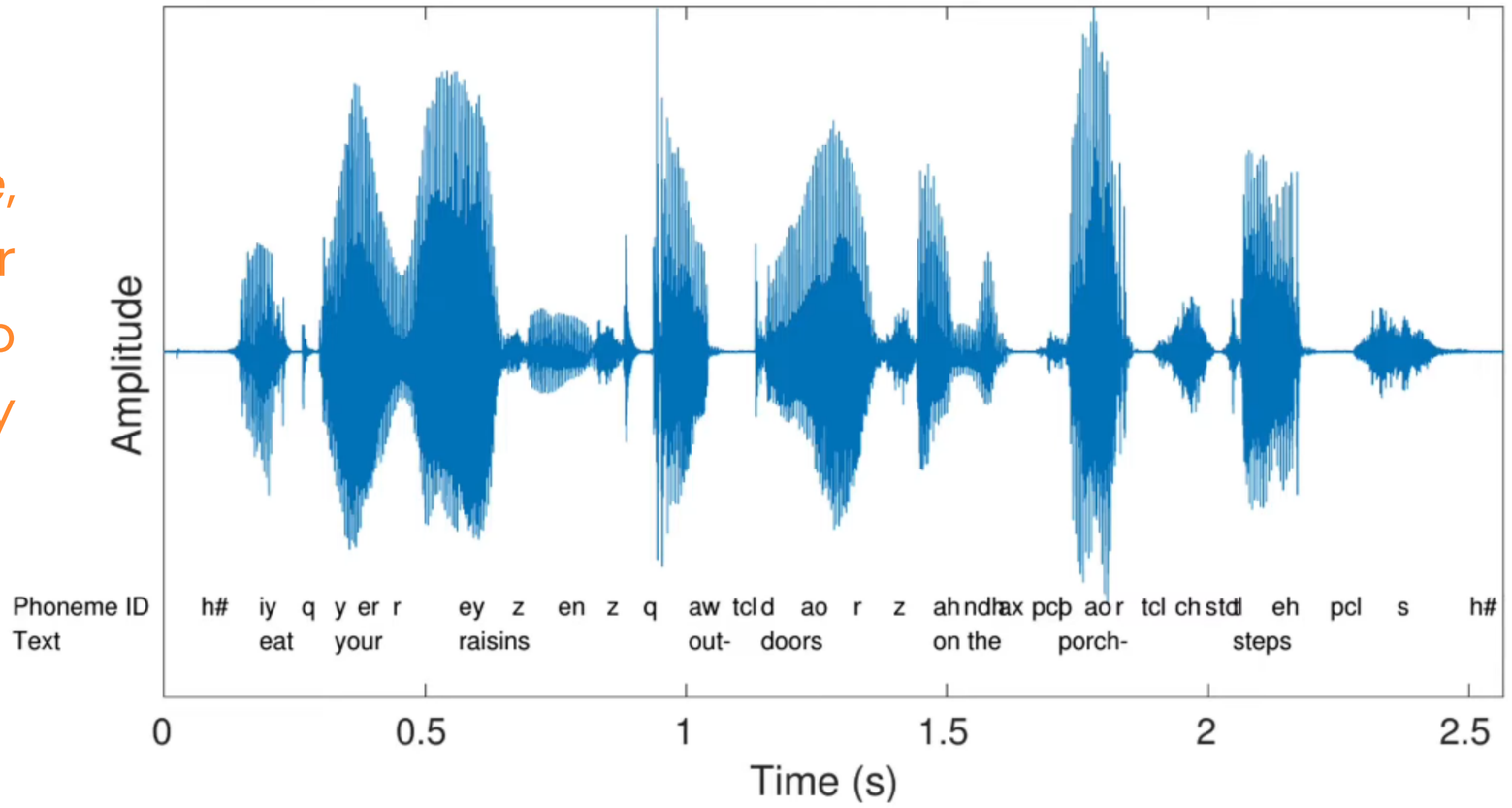
Audio Sample

Para procesar una canción debemos de analizar la muestra de audio que genera



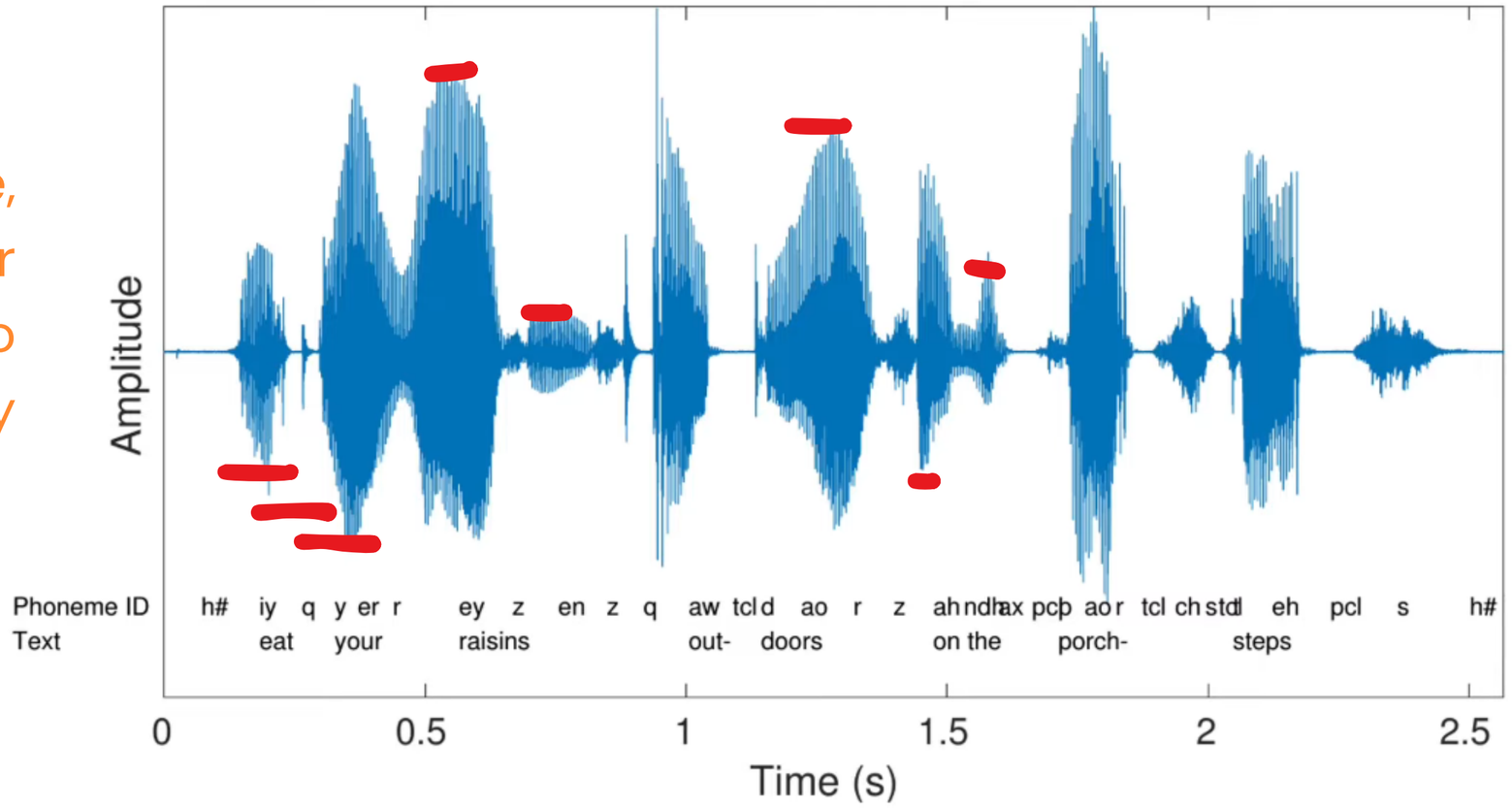
Windowing

La muestra es muy grande, por lo que debemos de tomar partes pequeñas (por lo general de 25ms) y procesarlas



Windowing

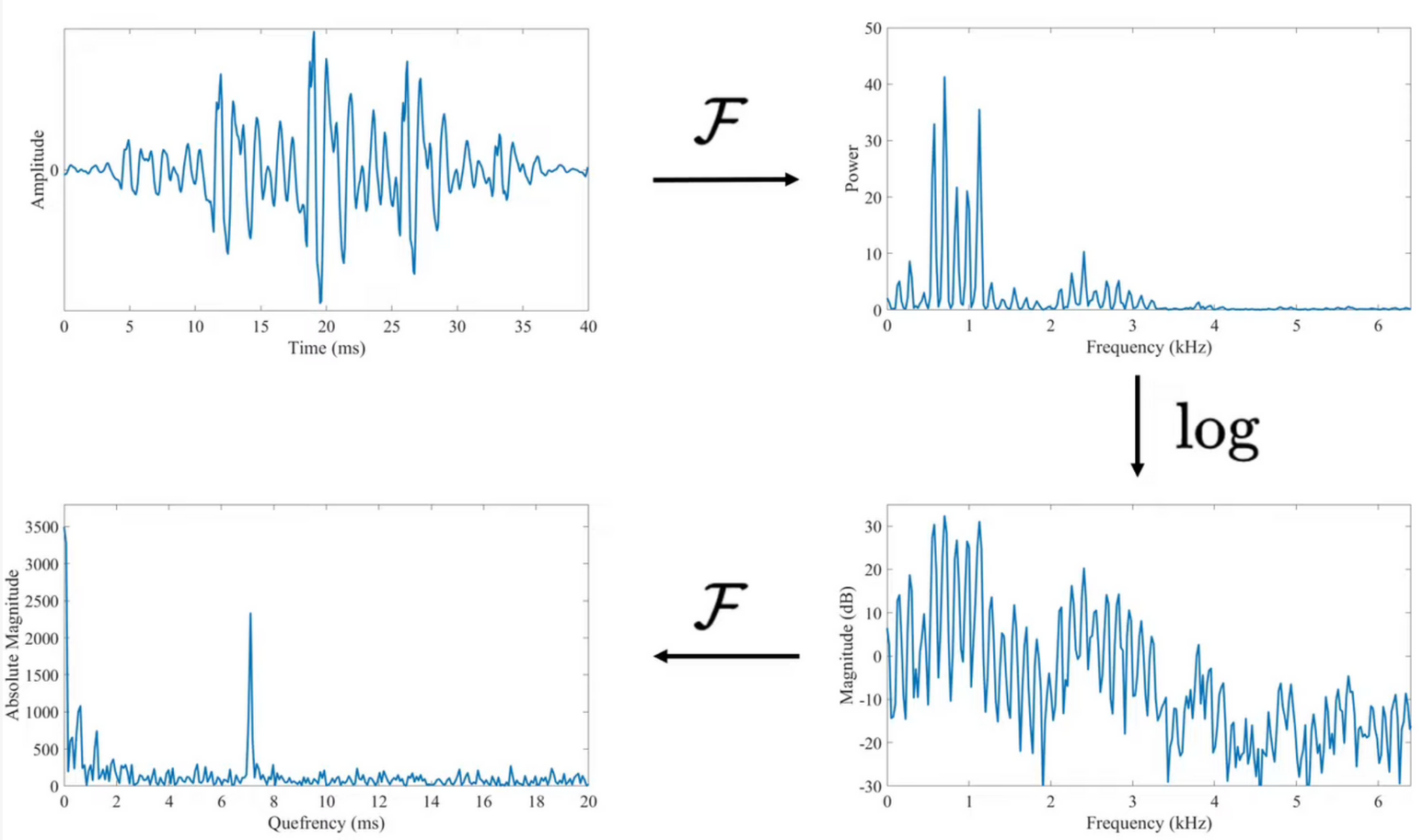
La muestra es muy grande, por lo que debemos de tomar partes pequeñas (por lo general de 25ms) y procesarlas



Matematicas complicadas

Cepstrum

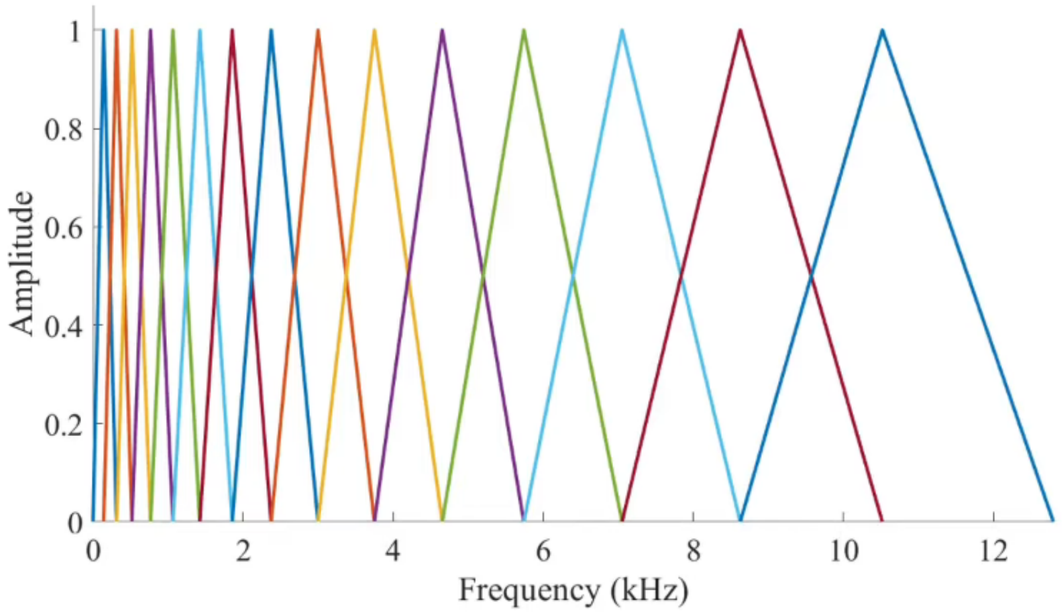
$$C_p = \left| \mathcal{F}\left\{ \log\left(\left| \mathcal{F}\{f(t)\} \right|^2 \right) \right\} \right|^2$$



Triangular Filterbank

Un paso extra para poder normalizar mejor nuestras ventadas y hacer un espectrograma más simple

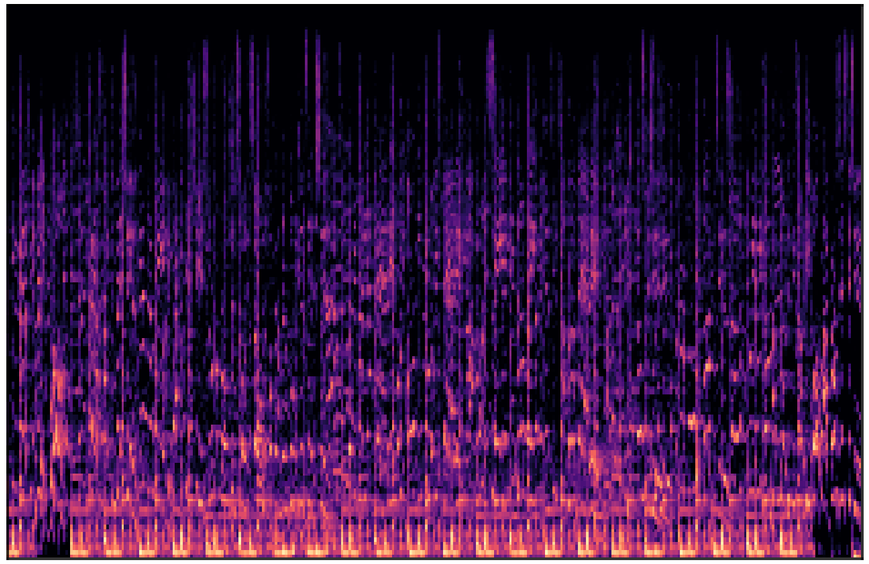
Triangular Filterbank



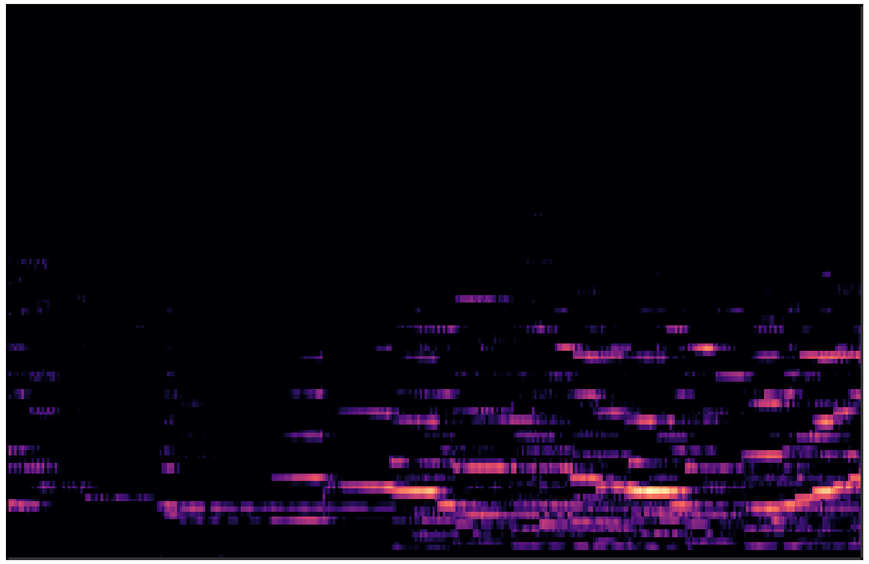
$$u_k = \sum_{h=f_{k-1}+1}^{f_{k+1}-1} w_{k,h} |x_h|^2$$

Espectograma

La unión de cada una de nuestras ventanas da como resultado un espectograma



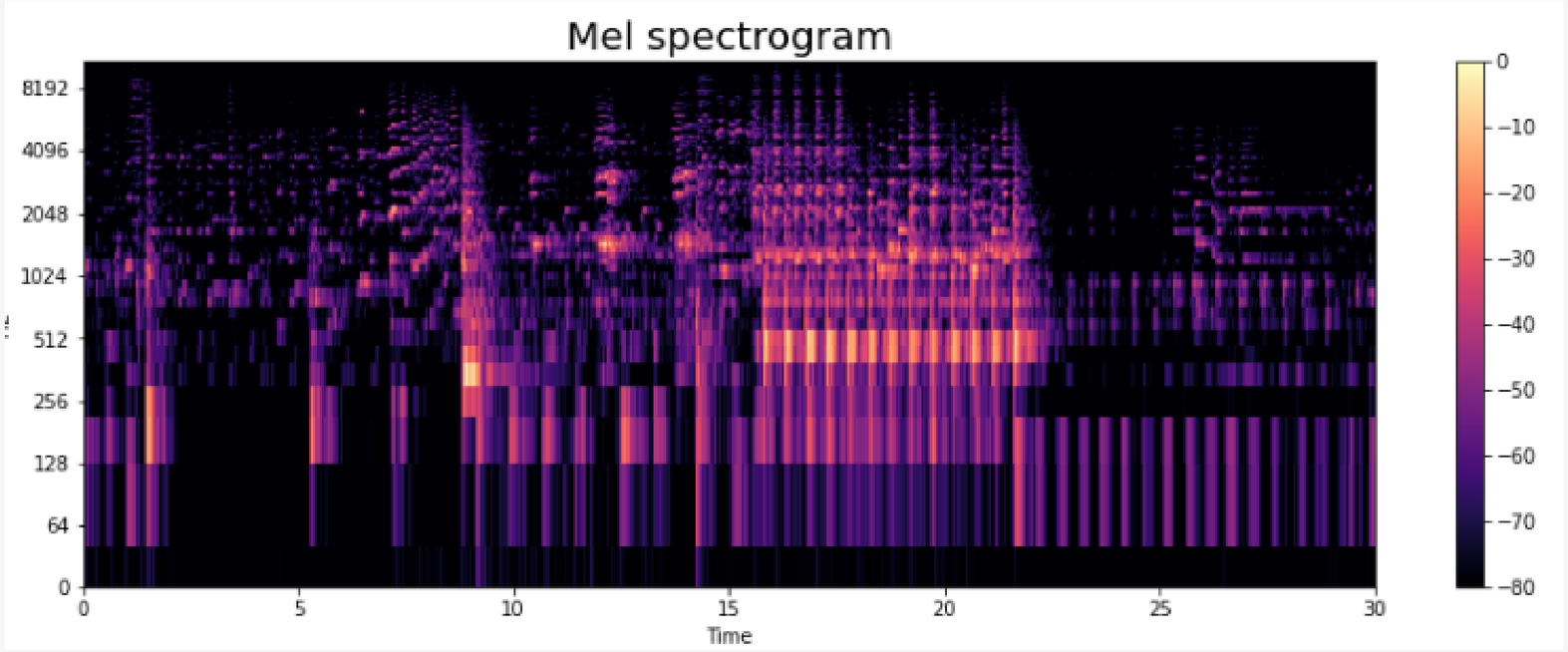
Espectrogama de una canción de hiphop



Espectrogama de una canción M clásica

Espectograma

Aplicando el triangular bank en cada una de nuestras ventana, conseguimos un espectrograma más simple



Dataset

Usamos GTZAN Dataset - Music Genre Classification que contiene 10 géneros distintos y 100 canciones por cada uno, y además contiene el desglose de cada una de las ventanas llamadas mfcc

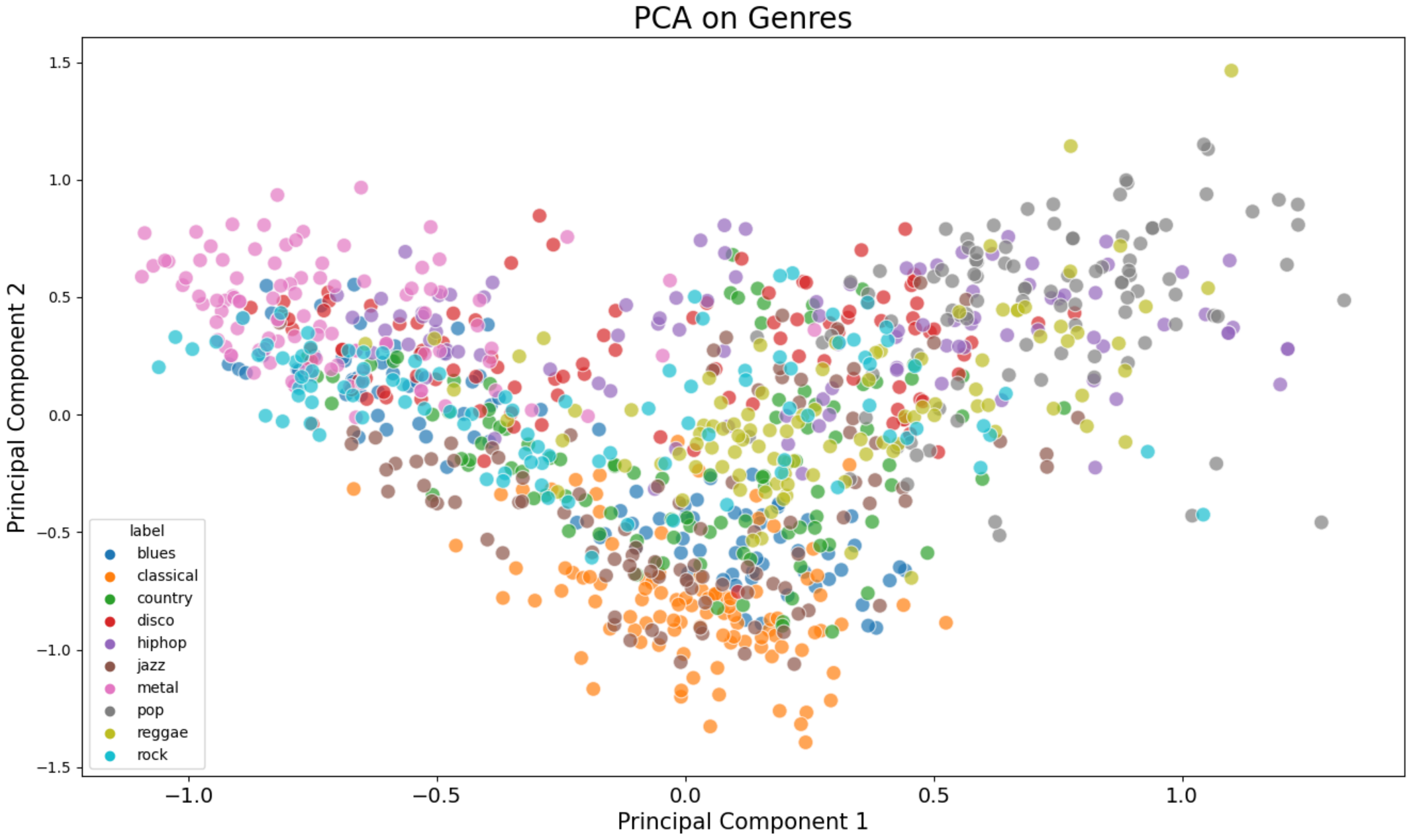
	filename	length	chroma_stft_mean	chroma_stft_var	rms_mean	rms_var	spectral_centroid_mean	spectral_centroid_var	spectral_bandwidth_mean	spectral_bandwidth_var	...	mfcc16_var
0	blues.00000.wav	661794	0.350088	0.088757	0.130228	0.002827	1784.165850	129774.064525	2002.449060	85882.761315	...	52.420910
1	blues.00001.wav	661794	0.340914	0.094980	0.095948	0.002373	1530.176679	375850.073649	2039.036516	213843.755497	...	55.356403
2	blues.00002.wav	661794	0.363637	0.085275	0.175570	0.002746	1552.811865	156467.643368	1747.702312	76254.192257	...	40.598766
3	blues.00003.wav	661794	0.404785	0.093999	0.141093	0.006346	1070.106615	184355.942417	1596.412872	166441.494769	...	44.427753
4	blues.00004.wav	661794	0.308526	0.087841	0.091529	0.002303	1835.004266	343399.939274	1748.172116	88445.209036	...	86.099236

5 rows × 60 columns

mfcc16_var	mfcc17_mean	mfcc17_var	mfcc18_mean	mfcc18_var	mfcc19_mean	mfcc19_var	mfcc20_mean	mfcc20_var	label
52.420910	-1.690215	36.524071	-0.408979	41.597103	-2.303523	55.062923	1.221291	46.936035	blues
55.356403	-0.731125	60.314529	0.295073	48.120598	-0.283518	51.106190	0.531217	45.786282	blues
40.598766	-7.729093	47.639427	-1.816407	52.382141	-3.439720	46.639660	-2.231258	30.573025	blues
44.427753	-3.319597	50.206673	0.636965	37.319130	-0.619121	37.259739	-3.407448	31.949339	blues
86.099236	-5.454034	75.269707	-0.916874	53.613918	-4.404827	62.910812	-11.703234	55.195160	blues

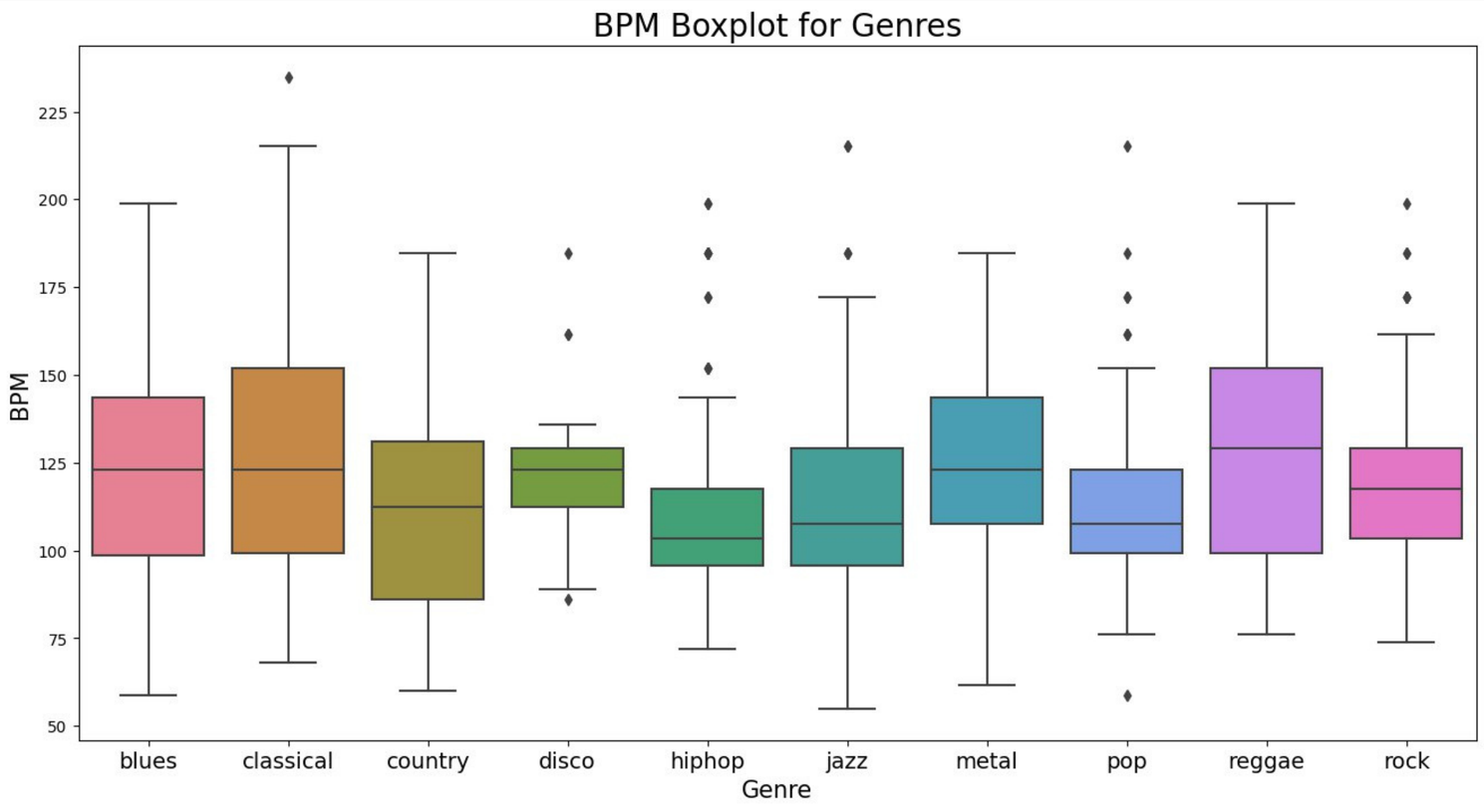
Visualización

Se hizo un PCA para obtener dos componentes principales y poder graficar los diferentes géneros


PCA on Genres
Principal Component 2
Principal Component 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
label
blues
classical
country
disco
hiphop
jazz
metal
pop
reggae
rock

Visualización

También se hizo un boxplot de los beats por minuto para cada género



Uso de algoritmos para clasificar

Se utilizo Grid Search, el cual es un método para poder encontrar lo mejores hiperparámetros, como por ejemplo para poder encontrar cual es la mejor profundidad en un random forest, o la k óptima en K-Neighbors, etc. Durante el uso de Grid Search se empleo cross validation para mejorar los resultados

Descision Tree Classifier

```
              precision    recall  f1-score   support

       blues       0.58      0.64      0.61       183
   classical       0.91      0.83      0.87       190
     country       0.53      0.52      0.53       208
       disco       0.62      0.56      0.59       215
      hiphop       0.66      0.66      0.66       200
        jazz       0.66      0.76      0.71       199
       metal       0.81      0.75      0.78       228
         pop       0.68      0.72      0.70       184
      reggae       0.67      0.66      0.66       195
        rock       0.58      0.59      0.59       196

    accuracy                           0.67      1998
   macro avg       0.67      0.67      0.67      1998
weighted avg       0.67      0.67      0.67      1998

------------------------------
Accuracy of Decision Tree Classifier : 0.6681681681681682
OrderedDict([('criterion', 'entropy'), ('max_depth', 100), ('min_impurity_decrease', 0.0), ('min_samples_leaf', 3), ('min_samples_split', 5), ('splitter', 'best')])
```

Logistic Regression

	precision	recall	f1-score	support
blues	0.65	0.68	0.66	183
classical	0.93	0.94	0.93	190
country	0.68	0.64	0.66	208
disco	0.69	0.66	0.68	215
hiphop	0.68	0.62	0.65	200
jazz	0.81	0.81	0.81	199
metal	0.85	0.82	0.84	228
pop	0.72	0.76	0.74	184
reggae	0.62	0.64	0.63	195
rock	0.51	0.56	0.54	196
accuracy			0.71	1998
macro avg	0.71	0.71	0.71	1998
weighted avg	0.71	0.71	0.71	1998

```
------------------------------
Accuracy of Logistic Regression : 0.7127127127127127
OrderedDict([('C', 100.0), ('penalty', 'l2'), ('solver', 'newton-cg')])
--------------------------------------------------------------------------------
```

K-Neighbors

	precision	recall	f1-score	support
blues	0.88	0.91	0.89	183
classical	0.90	0.98	0.94	190
country	0.81	0.88	0.84	208
disco	0.89	0.94	0.91	215
hiphop	0.93	0.90	0.91	200
jazz	0.91	0.85	0.88	199
metal	0.98	0.93	0.95	228
pop	0.96	0.89	0.92	184
reggae	0.87	0.90	0.89	195
rock	0.88	0.81	0.84	196
accuracy			0.90	1998
macro avg	0.90	0.90	0.90	1998
weighted avg	0.90	0.90	0.90	1998

```
------------------------------
Accuracy of K-Neighbors Classifier : 0.8978978978978979
OrderedDict([('algorithm', 'auto'), ('n_jobs', -1), ('n_neighbors', 5)])
------------------------------------------------------------------------------
```

Random Tree Forest

	precision	recall	f1-score	support
blues	0.75	0.85	0.80	183
classical	0.89	0.97	0.93	190
country	0.72	0.68	0.70	208
disco	0.73	0.67	0.69	215
hiphop	0.80	0.69	0.74	200
jazz	0.79	0.88	0.84	199
metal	0.84	0.90	0.87	228
pop	0.77	0.80	0.78	184
reggae	0.68	0.81	0.74	195
rock	0.72	0.47	0.57	196
accuracy			0.77	1998
macro avg	0.77	0.77	0.77	1998
weighted avg	0.77	0.77	0.77	1998

```
------------------------------
Accuracy of Random Forest Classifier : 0.7712712712712713
OrderedDict([('criterion', 'entropy'), ('max_depth', 8), ('max_features', 'sqrt'), ('n_estimators', 462)])
--------------------------------------------------------------------------------
```

Gradient Boosting

```
GBC  Classification
/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_gb.py:280: FutureWarning: The loss parameter name 'deviance' was deprecated i
  warnings.warn(
              precision    recall  f1-score   support

       blues       0.48      0.54      0.51       183
   classical       0.70      0.86      0.77       190
     country       0.55      0.45      0.50       208
       disco       0.59      0.47      0.52       215
      hiphop       0.58      0.45      0.51       200
        jazz       0.54      0.56      0.55       199
       metal       0.69      0.79      0.74       228
         pop       0.50      0.71      0.59       184
      reggae       0.49      0.57      0.53       195
        rock       0.33      0.17      0.23       196

    accuracy                           0.56      1998
   macro avg       0.54      0.56      0.54      1998
weighted avg       0.55      0.56      0.55      1998

------------------------------
Accuracy of GBC Classifier : 0.5585585585585585
OrderedDict([('criterion', 'squared_error'), ('learning_rate', 0.2), ('loss', 'deviance'), ('max_depth', 5), ('max_features', 'sqrt'),

, ('min_samples_leaf', 0.13636363636363638), ('min_samples_split', 0.13636363636363638), ('n_estimators', 10), ('subsample', 0.95)])
```

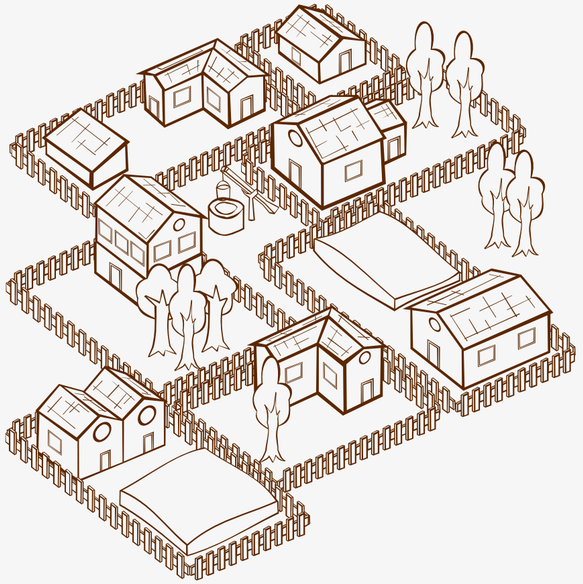
XGB

```
XGB  Classification
/usr/local/lib/python3.10/dist-packages/skopt/optimizer/optimizer.py:449: UserWarning: The objective has been evaluated at this point before.
  warnings.warn("The objective has been evaluated "
/usr/local/lib/python3.10/dist-packages/skopt/optimizer/optimizer.py:449: UserWarning: The objective has been evaluated at this point before.
  warnings.warn("The objective has been evaluated "
/usr/local/lib/python3.10/dist-packages/skopt/optimizer/optimizer.py:449: UserWarning: The objective has been evaluated at this point before.
  warnings.warn("The objective has been evaluated "
[23:26:54] WARNING: ../src/learner.cc:767:
Parameters: { "n_stimators" } are not used.

              precision    recall  f1-score   support

           0       0.87      0.89      0.88       183
           1       0.94      0.98      0.96       190
           2       0.78      0.91      0.84       208
           3       0.85      0.84      0.84       215
           4       0.90      0.88      0.89       200
           5       0.90      0.89      0.90       199
           6       0.95      0.91      0.93       228
           7       0.91      0.83      0.87       184
           8       0.87      0.84      0.85       195
           9       0.76      0.75      0.75       196

    accuracy                           0.87      1998
   macro avg       0.87      0.87      0.87      1998
weighted avg       0.87      0.87      0.87      1998

--------------------------------
Accuracy of xgb Classifier : 0.8713713713713713
OrderedDict([('learning_rate', 0.09996710392811119), ('n_stimators', 191)])
--------------------------------------------------------------------------------
```

¿Ganador?

¡K-neighbors!

Modelo	Accuracy
Decision Tree	0.67
Regresión Logística	0.71
K-Neighbors	0.90
Random Forest	0.77
GradientBoost	0.56
XGB	0.87
Red Multicapa	0.9

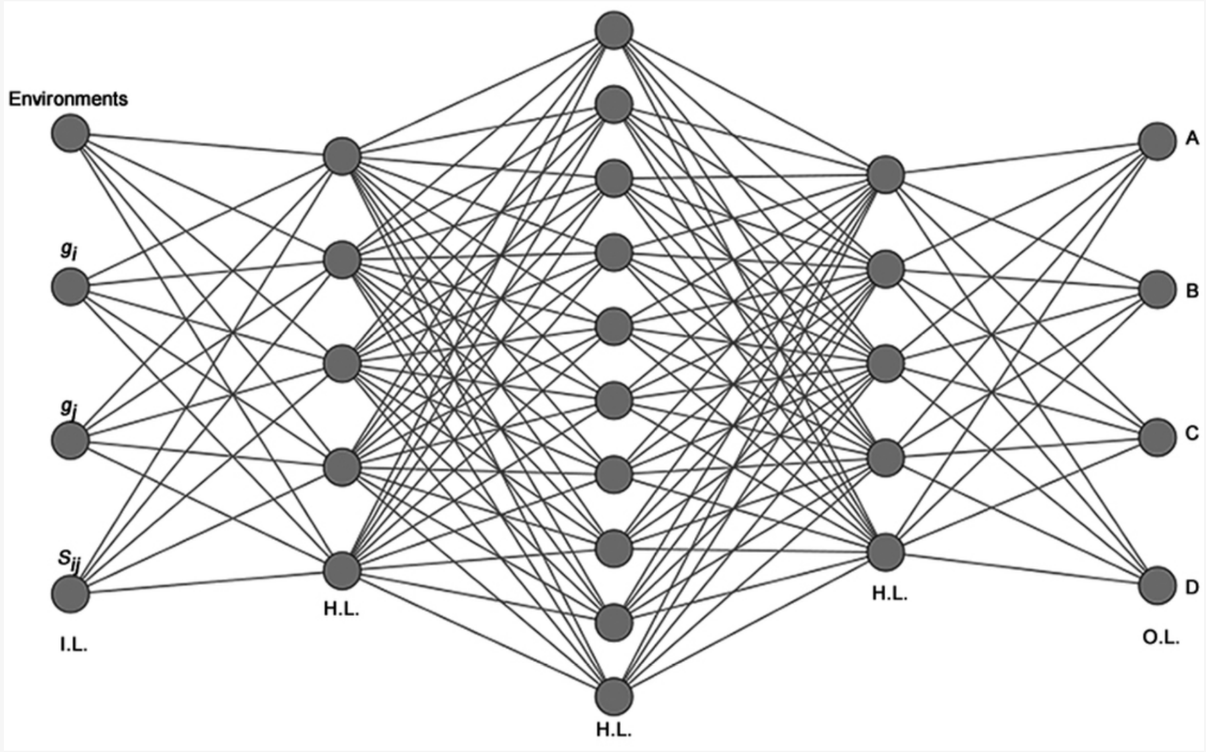


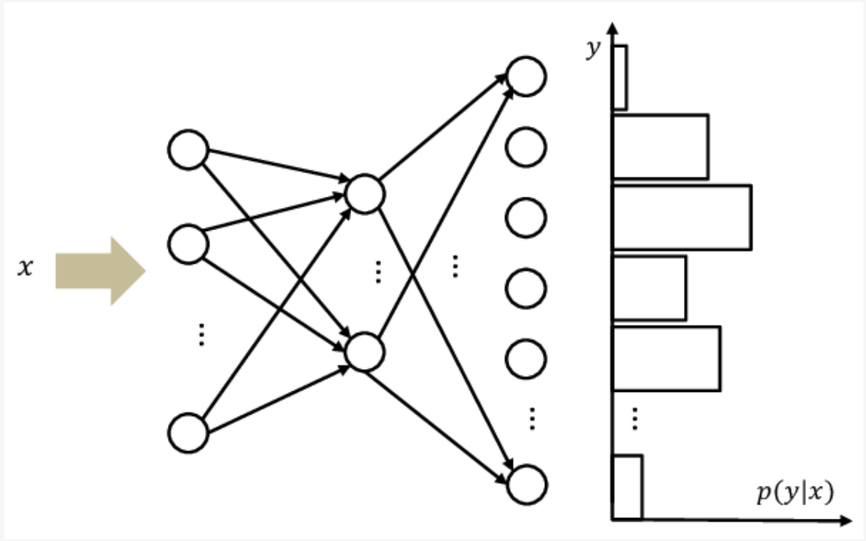
Red neuronal

La arquitectura usada fue la siguiente:

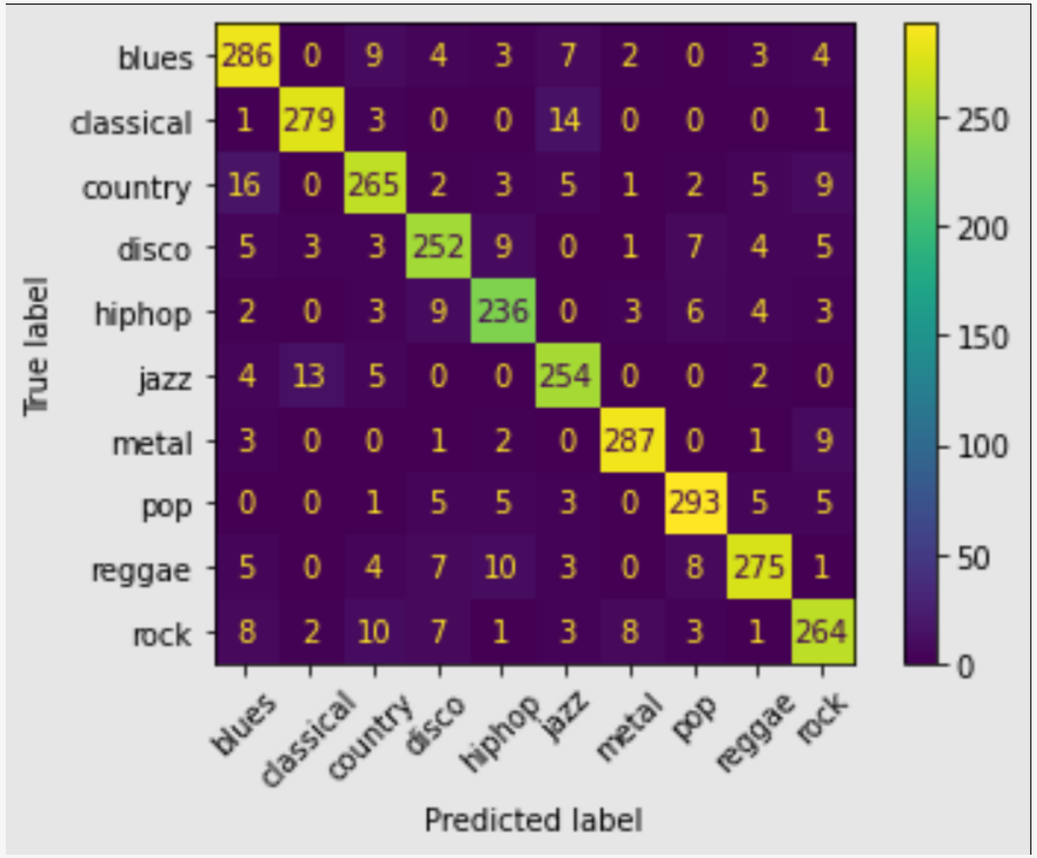
- Capa de 1024 neuronas
- Capa de 512 neuronas
- Capa de 512 neuronas
- Capa de 256 neuronas
- Capa de 128 neuronas
- Capa de 64 neuronas
- Capa softmax para clasificar

Entre cada capa se usó un dropout





Red neuronal



	precision	recall	f1-score	support
0	0.90	0.92	0.91	318
1	0.93	0.92	0.92	298
2	0.89	0.83	0.86	308
3	0.85	0.89	0.87	289
4	0.87	0.89	0.88	266
5	0.85	0.91	0.88	278
6	0.96	0.94	0.95	303
7	0.94	0.92	0.93	317
8	0.92	0.88	0.90	313
9	0.86	0.87	0.87	307
accuracy			0.90	2997
macro avg	0.90	0.90	0.90	2997
weighted avg	0.90	0.90	0.90	2997

Conclusiones

Para algunos géneros, se pueden observar diferencias claras como el metal y el pop, pero para otros como el blues y el Jazz, debido a origenes históricos no tienen diferencias tan claras.

Podemos detectar cómo hay géneros que influencian a otros géneros debido a sus similitudes.

La red neuronal logró clasificar los 10 géneros de manera exitosa