GTZAN Dataset - Music Genre Classification

Proyecto Final

Tania Michelle Rubí Rojas

Facultad de Ciencias, UNAM

5 de febrero de 2021

Contenido

Objetivo

Dada una canción, queremos saber a qué genero musical pertenece.
Para atacar este problema, utilizaremos dos *modelos* de clasificación diferentes:

1. SVC (Support-Vector Clustering)
   Se explicará el preprocesamiento de los datos antes de usar SVC, y su precisión al momento de clasificar canciones.
2. PCA/SVC (Principal Component Analysis/ Singular Vector Clustering)
   Se explicará el preprocesamiento necesario para poder aplicar PCA y la precisión de este nuevo modelo al momento de clasificar canciones.

Conjunto de Datos

El conjunto de canciones que utlizaremos para entrenar nuestros modelos se obtuvo del sitio web *Kaggle*

`https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification`

Conjunto de Datos

el cual contiene 2 archivos y 2 carpetas (pero sólo nos enfocaremos en el primer archivo):

1. *features_30_sec.csv*
   Contiene un conjunto de datos con 60 atributos (características basadas en el timbre):
   - **filename**: el nombre del archivo.
   - **length**: el tamaño del archivo.
   - **spectral_bandwidth**: el ancho de banda espectral.
   - **spectral_centroid**: el centroide espectral.
   - **rms**: el nivel promedio de una onda (en el espectro).
   - **label**: el género musical al cual pertenece.

   con 1000 entradas (10 por cada género)

Conjunto de Datos

Tenemos 10 géneros musicales: blues, clásica, country, disco, hiphop, jazz, metal, pop, reggae, rock.

filename	length	chroma_stft_mean	chroma_stft_var	rms_mean	rms_var	spectral_centroid_mean	spectral_centroid_var	spectral_bandwidth_mean
blues.00000.wav	661794	0.350088	0.088757	0.130228	0.002827	1784.165850	1.297741e+05	2002.449060
blues.00001.wav	661794	0.340914	0.094980	0.095948	0.002373	1530.176679	3.758501e+05	2039.036516
blues.00002.wav	661794	0.363637	0.085275	0.175570	0.002746	1552.811865	1.564676e+05	1747.702312
blues.00003.wav	661794	0.404785	0.093999	0.141093	0.006346	1070.106615	1.843559e+05	1596.412872
blues.00004.wav	661794	0.308526	0.087841	0.091529	0.002303	1835.004266	3.433999e+05	1748.172116
blues.00005.wav	661794	0.302456	0.087532	0.103494	0.003981	1831.993940	1.030482e+06	1729.653287
blues.00006.wav	661794	0.291328	0.093981	0.141874	0.008803	1459.366472	4.378594e+05	1389.009131
blues.00007.wav	661794	0.307955	0.092903	0.131822	0.005531	1451.667066	4.495682e+05	1577.270941
blues.00008.wav	661794	0.408879	0.086512	0.142416	0.001507	1719.368948	1.632828e+05	2031.740381
blues.00009.wav	661794	0.273950	0.092316	0.081314	0.004347	1817.150863	2.982361e+05	1973.773306

Preprocesamiento de datos

Dividimos nuestro conjunto en dos:

- X para los atributos característica.
- y para las etiquetas (géneros).

```
1    # Seleccionamos las primeras 59 columnas.
2    X = df.drop(['filename', 'label'], axis=1)
3    y = df['label'] # Seleccionamos la columna 60.
```

Preprocesamiento de Datos

Normalizamos nuestro conjunto X usando `StandardScaler`. Ésta clase estándariza los datos eliminando la media y escalando los datos de forma que su varianza sea igual a 1.

length	chroma_stft_mean	chroma_stft_var	rms_mean	rms_var	spectral_centroid_mean	spectral_centroid_var	spectral_bandwidth_mean
-0.132822	-0.350137	0.312587	-0.010690	-0.061856	-0.583585	-0.848311	-0.456402
-0.132822	-0.462482	1.117572	-0.532852	-0.186821	-0.938516	-0.234194	-0.386852
-0.132822	-0.184225	-0.137701	0.679978	-0.084093	-0.906885	-0.781694	-0.940663
-0.132822	0.319639	0.990659	0.154810	0.907029	-1.581429	-0.712095	-1.228256
-0.132822	-0.859077	0.194163	-0.600165	-0.205909	-0.512542	-0.315178	-0.939770

Preprocesamiento de Datos

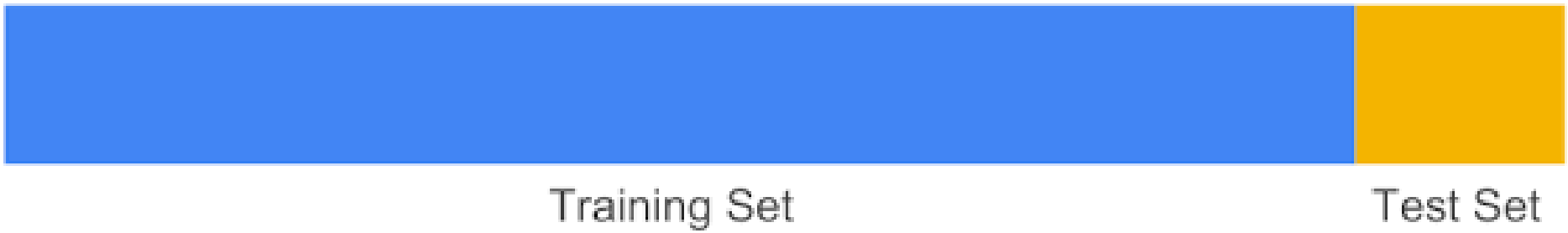
La variable de salida es un valor de string. Debemos convertirlos en valores enteros entre 0 y 9. Esto lo podemos lograr usando la clase `LabelEncoder`, pues ésta modelará la codificación requerida y creará una nueva variable de salida.

Así, obtenemos que:

- $0 \rightarrow$ blues $\quad 1 \rightarrow$ classical
- $2 \rightarrow$ country $\quad 3 \rightarrow$ disco
- $4 \rightarrow$ hiphop $\quad 5 \rightarrow$ jazz
- $6 \rightarrow$ metal $\quad 7 \rightarrow$ pop
- $8 \rightarrow$ reggae $\quad 9 \rightarrow$ rock

Preprocesamiento de Datos

Dividimos nuestros nuevos conjuntos de datos con un split del $80 - 20$ para entrenamiento y prueba, respectivamente.



SVC

Creamos nuestro modelo SVC (usando un kernel `lineal`) y lo entrenamos. Luego, ya podemos comenzar a realizar nuestras predicciones.

```
1     svclassifier = SVC(kernel='linear')
2     svclassifier.fit(X_train, y_train)
3     y_pred = svclassifier.predict(X_test)
```

SVC

En el conjunto de entrenamiento obtenemos un 98 % de precisión, mientras que en el conjunto de prueba obtenemos un 76 % de precisión.

PCA

Usaremos PCA para obtener la lista de atributos que tienen mayor varianza (mayor poder explicativo). Éstos serán las componentes principales.

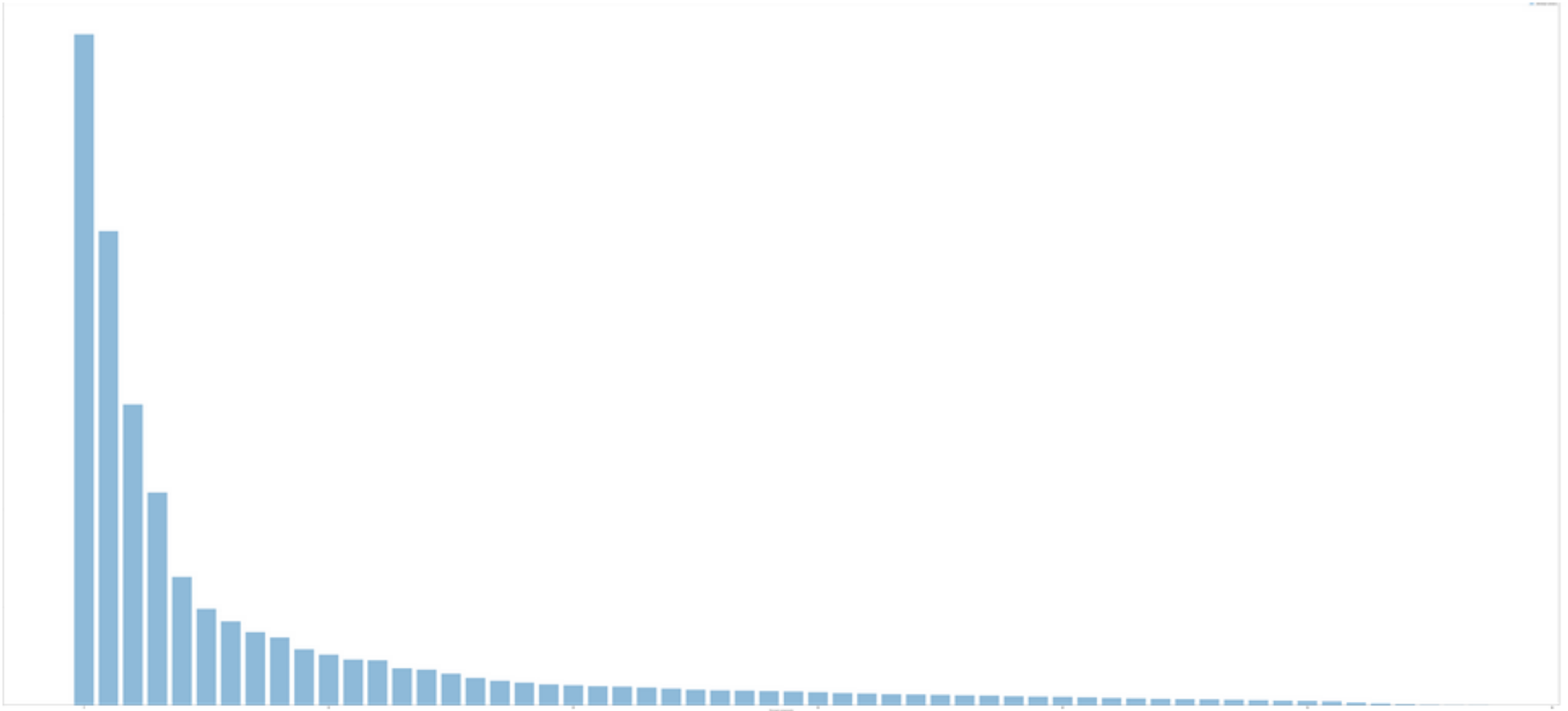


Figura: Variance ratio vs Principal components

PCA

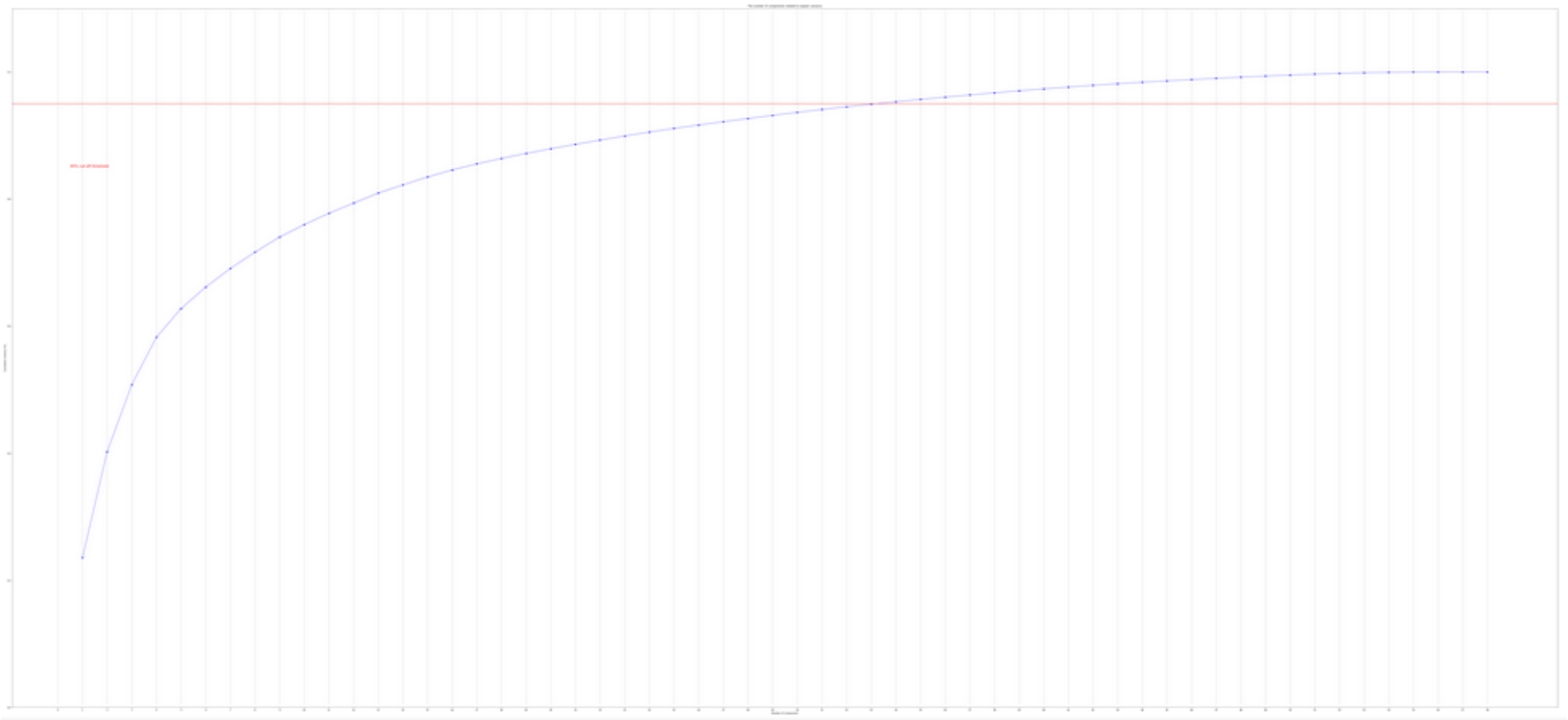


Figura: Cumulative variance vs The number of components needed to explain variance

PCA

Obtenemos que el número de componentes para obtener una varianza explicada del 98 % es de 43.

PCA

Instanciamos un objeto de PCA y lo aplicamos.

```
1     svclassifier = SVC(kernel='linear')
2     svclassifier.fit(X_train, y_train)
3     y_pred = svclassifier.predict(X_test)
```

Finalmente, usamos SVC con este nuevo conjunto reducido.

PCA

En el conjunto de entrenamiento obtenemos un 96 % de precisión, mientras que en el conjunto de prueba obtenemos un 74 % de precisión.

Resultados

Obtenemos que gracias a PCA logramos una precisión muy buena con 43 componentes principales.

- SVC

  98 % `train` 76 % `test`

- PCA/SVC

  96 % `train` 74 % `test`

Resultados

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

(a) SVC

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

(b) PCA

Figura: Predicciones

Resultados

```
Mean Absolute Error: 1.00
Mean squared error: 5.24
Root Mean Squared Error: 2.29
Variance score: 0.35
```

(a) SVC

```
Mean Absolute Error: 0.98
Mean squared error: 4.93
Root Mean Squared Error: 2.22
Variance score: 0.42
```

(b) PCA

Figura: Precisión

Conclusiones