Análisis de Datos - Especialización en Inteligencia Artificial

Trabajo práctico integrador

Grupo 3:

- Avalos Ribas Gonzalo
- Rassi Francisco
- Silva Victor

1 - Introducción y motivación

De una lista de posibles datasets se eligió “Datos de distintas canciones en Spotify“. Con este se desea poder estimar si un tema nuevo será del gusto de la persona que tiene esta playlist activa.

2. Análisis exploratorio inicial

Se visualizan las primeras filas para del dataset:

	acousticness	danceability	duration	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	label
0	0.713	0.514	100125	0.521	0.816000	8	0.1120	-14.835	0	0.0444	119.879	4	0.143	1
1	0.192	0.714	207019	0.614	0.000000	4	0.2630	-6.935	1	0.0319	123.969	4	0.582	1
2	0.333	0.630	216200	0.455	0.000004	5	0.1270	-9.290	1	0.0292	139.931	4	0.199	1
3	0.601	0.810	136413	0.221	0.210000	5	0.1840	-11.005	1	0.0429	109.960	4	0.798	1
4	0.883	0.465	181440	0.459	0.000173	6	0.0692	-8.137	0	0.0351	90.807	4	0.288	1

Figura 1 - Primeras filas del dataset usando el método .head()

Se realiza un resumen de 5 números usando el método .describe(), donde se puede ver datos estadísticos de las features del dataset.

	acousticness	danceability	duration	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	label
count	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000	750.000000
mean	0.357394	0.596439	220112.733333	0.594188	0.100245	4.829333	0.203376	-8.509339	0.741333	0.098966	120.405761	3.902667	0.497321	0.602667
std	0.338405	0.172036	65587.690483	0.253301	0.259921	3.636001	0.177609	5.039488	0.438194	0.104715	28.378116	0.400091	0.239615	0.489673
min	0.000001	0.107000	33840.000000	0.009250	0.000000	0.000000	0.024000	-29.601000	0.000000	0.023400	55.747000	1.000000	0.033200	0.000000
25%	0.037150	0.480000	185490.250000	0.423250	0.000000	1.000000	0.094550	-10.173500	0.000000	0.035900	98.998000	4.000000	0.297000	0.000000
50%	0.244500	0.606000	215108.500000	0.631500	0.000010	5.000000	0.129000	-7.270000	1.000000	0.048750	120.104500	4.000000	0.483000	1.000000
75%	0.678500	0.715750	244236.750000	0.804750	0.002245	8.000000	0.264750	-5.097750	1.000000	0.113000	138.074750	4.000000	0.684500	1.000000
max	0.994000	0.986000	675360.000000	0.995000	0.967000	11.000000	0.979000	-0.533000	1.000000	0.721000	204.162000	5.000000	0.975000	1.000000

Figura 2 - Resumen de 5 números usando el método .describe()

Los tipos de datos del dataset se pueden clasificar como cualitativos o cuantitativos, en el siguiente cuadro se muestra esta clasificación así como también la escala de medida, si es informativa y un comentario de lo que representa:

Variable	Tipo	Escalas de medida	Informativa	Comentario
Acousticness	Cuantitativa (continua)	De razón	Sí	Esta variable mide el nivel de sonido acústico de una canción
Danceability	Cuantitativa (continua)	De razón	Sí	Bailabilidad de una canción
Duration	Cuantitativa (discreta)	De razón	Sí	Duración de la canción en ms
Energy	Cuantitativa (continua)	De razón	Sí	Medida perceptiva de la intensidad
instrumentalness	Cuantitativa (continua)	De razón	Sí	Indica la Instrumentalidad de la canción
Key	Cualitativa	Nominal	No	Indica el tono en que fue escrita la canción,
Liveness	Cuantitativa (continua)	De razón	Sí	Indica qué tan probable es que la canción haya sido grabada en vivo
Loudness	Cuantitativa (continua)	De razón	Sí	Esta variable nos indica el volumen general de una canción en dB
Mode	Cualitativa	Nominal	Sí	Indica si la canción se encuentra en modo menor (valor de 0) o mayor (valor de 1)
Speechiness	Cuantitativa (continua)	De razón	Sí	Información sobre la presencia de partes habladas en una canción
Tempo	Cuantitativa (continua)	De razón	Sí	Indica los beats por minuto que tiene la canción
Time_signature	Cualitativa	Ordinal	Sí	Indica en que compás está la canción
Valence	Cuantitativa (continua)	De razón	Sí	Indica la felicidad de la canción
Label	Cualitativa	Nominal	Sí	Indica si recomienda la canción o no

Figura 3 - Clasificación de variables

Las variables de entrada son todas las detalladas en el cuadro anterior, a excepción de “Label” que es la variable de salida.

Se crean histogramas para cada una de las variables y así observar su distribución. En este caso se utilizan 30 bins para cada histograma:

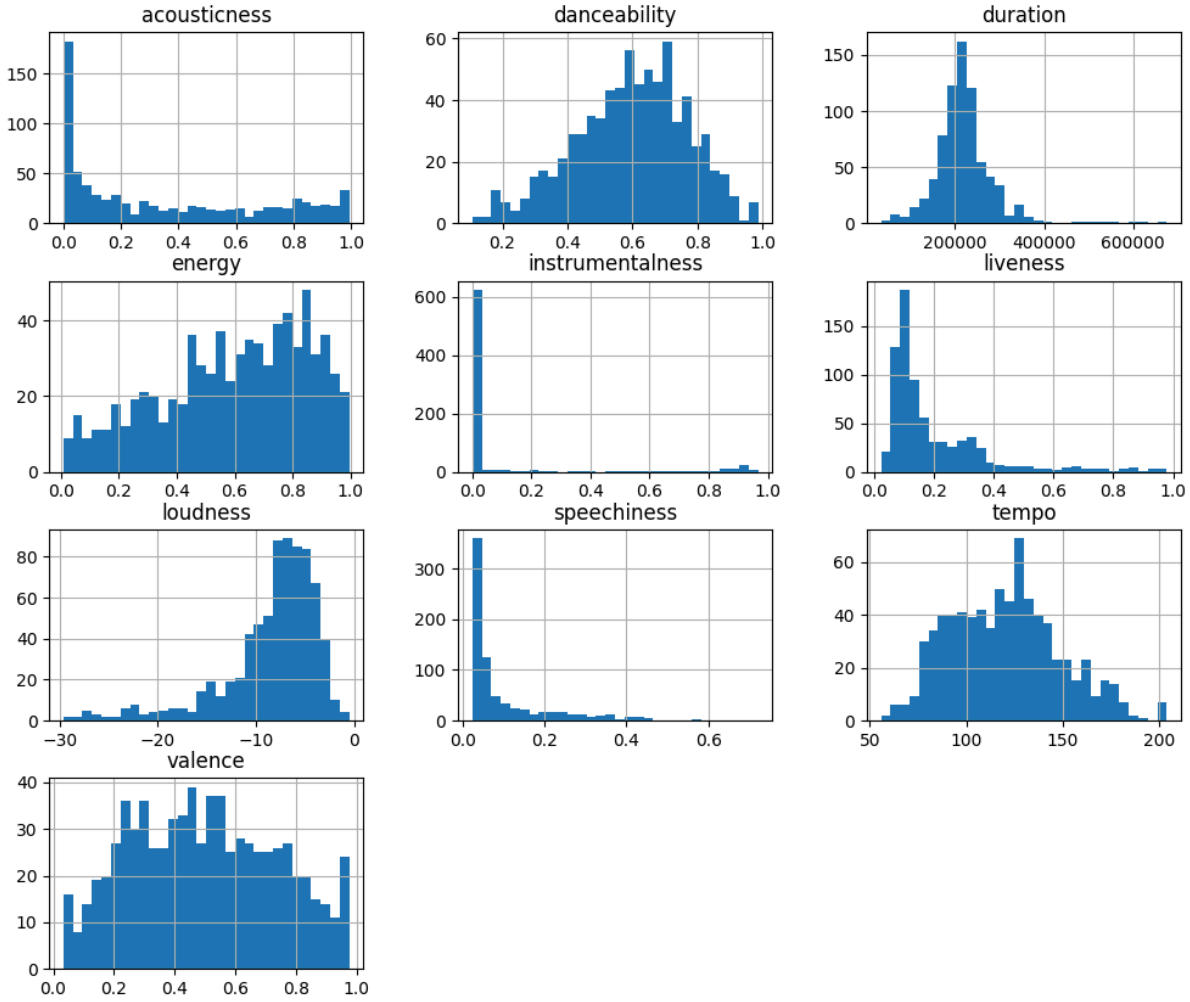

Figura 4 - Histogramas de las features

A continuación, se analiza la oblicuidad de las variables:

Variable	Oblicuidad
acousticness	0.534804
danceability	-0.311981
duration	1.576231
energy	-0.458765
instrumentalness	2.488166
liveness	2.156240
loudness	-1.693115

speechiness	2.040370
tempo	0.334763
valence	0.104812

Tabla 1 - Oblicuidad de las variables

Se observa que aquellas variables que tienen Oblicuidad mayor que cero tienden a tener una tendencia hacia la derecha, mientras las que son menor que cero, hacia la izquierda.

Se continúa analizando la curtosis:

Variable	**Curtosis**
acousticness	-1.210296
danceability	-0.296706
duration	8.321863
energy	-0.758962
instrumentalness	4.518012
liveness	4.842701
loudness	3.232917
speechiness	4.149336
tempo	-0.246166
valence	-0.911007

Tabla 2 - Curtosis de las variables

La curtosis permite caracterizar las colas de la distribución. Los valores cercanos a cero, son más parecidos a una normal. Ej: Bailabilidad. Los valores más grandes, muestran distribuciones más apuntadas que la normal, y los menores cero, más aplanadas.

Generamos ahora el QQ Plot de las variables:

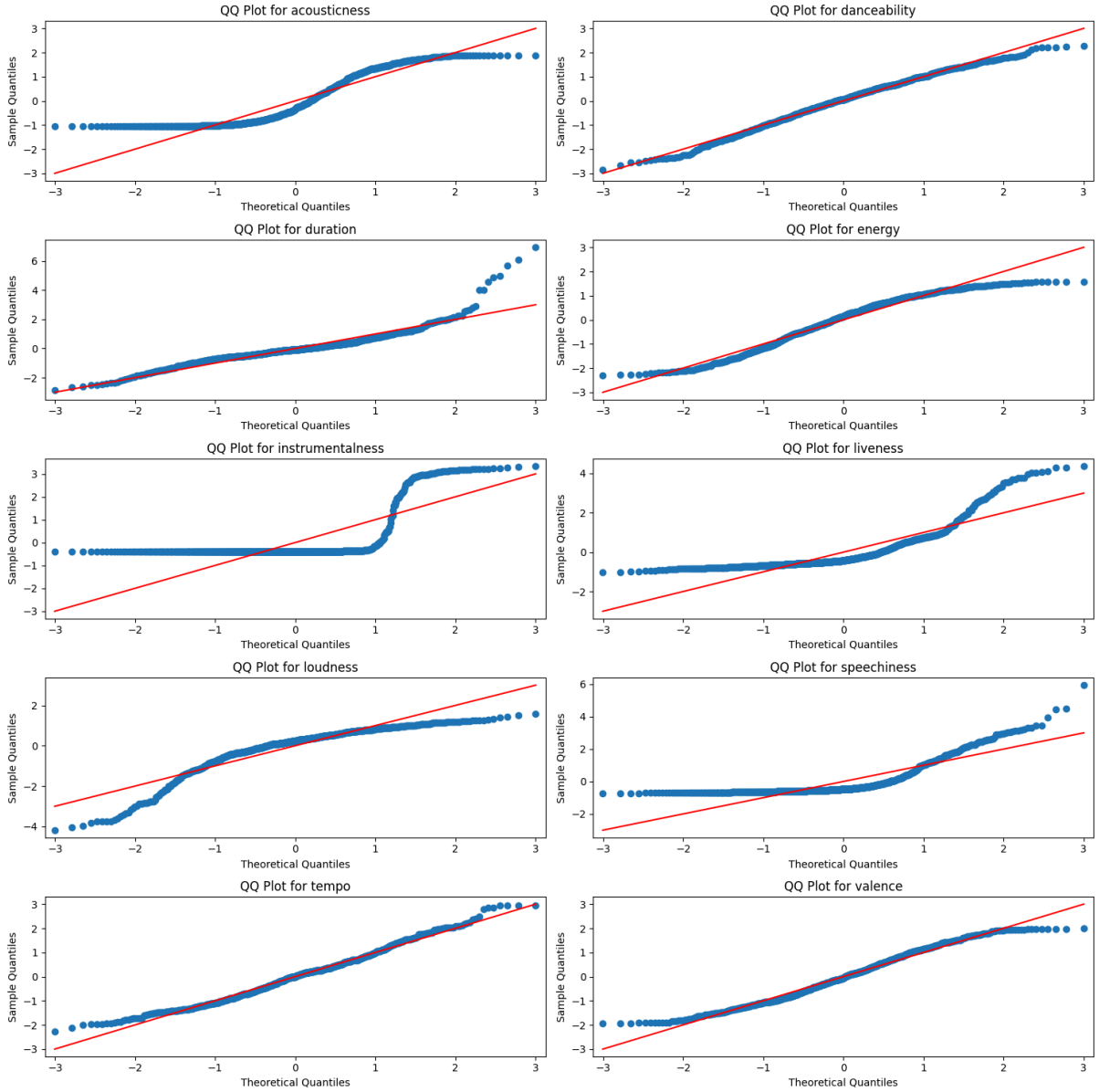

Figura 5 - QQ plot de las variables

Los gráficos de QQ plot llevan a las mismas conclusiones que lo obtenido anteriormente a partir del análisis de oblicuidad y curtosis. Se seleccionan algunas variables para ver en particular:

- Bailabilidad (danceability): Tanto en curtosis (por exceso) como en oblicuidad tiene valores similares a cero, lo que le da una tendencia bastante normal, por eso se mantiene cerca de la diagonal.

- Instrumentalidad (Instrumentalness): Al observar el QQ plot parece que no hay cuantiles de muestra hasta que se en un pequeño tramo la curva azul sube rápidamente hasta incorporar todos los percentiles. Esto ocurre por la forma del histograma.

A continuación, se genera el pairplot de las variables que permite visualizar relaciones entre estas de todo tipo (más allá de las lineales):

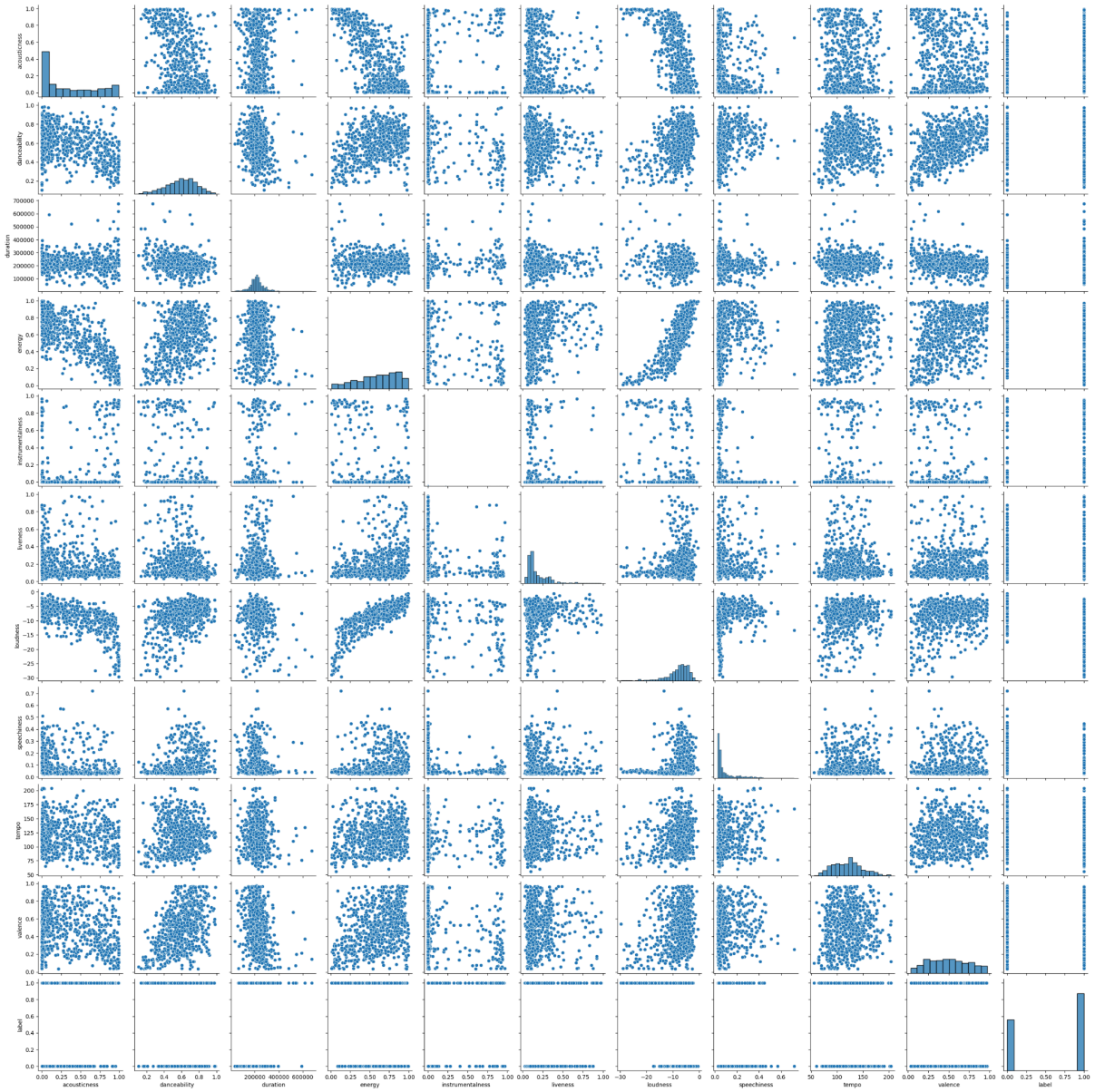

Figura 6 - Pairplot de las variables

Para las variables categóricas, se calcula su cardinalidad:

key:	12
mode:	2
time_signature:	4

Para la variable de salida "label", se muestra la cantidad de salidas por clase:

1:	452
0:	298

Se puede apreciar que el valor 0 para la variable label se repite en 298 muestras, mientras que 452 muestras toman el valor 1.

Las clases no se encuentran balanceadas, pero este desbalance es aceptable (se tiene una relación 60-40 entre clases).

3. Limpieza y preparación de datos / ingeniería de features

A continuación, se analizará la relación entre las variables de entrada. Para eso, se comienza calculando la matriz de correlación:

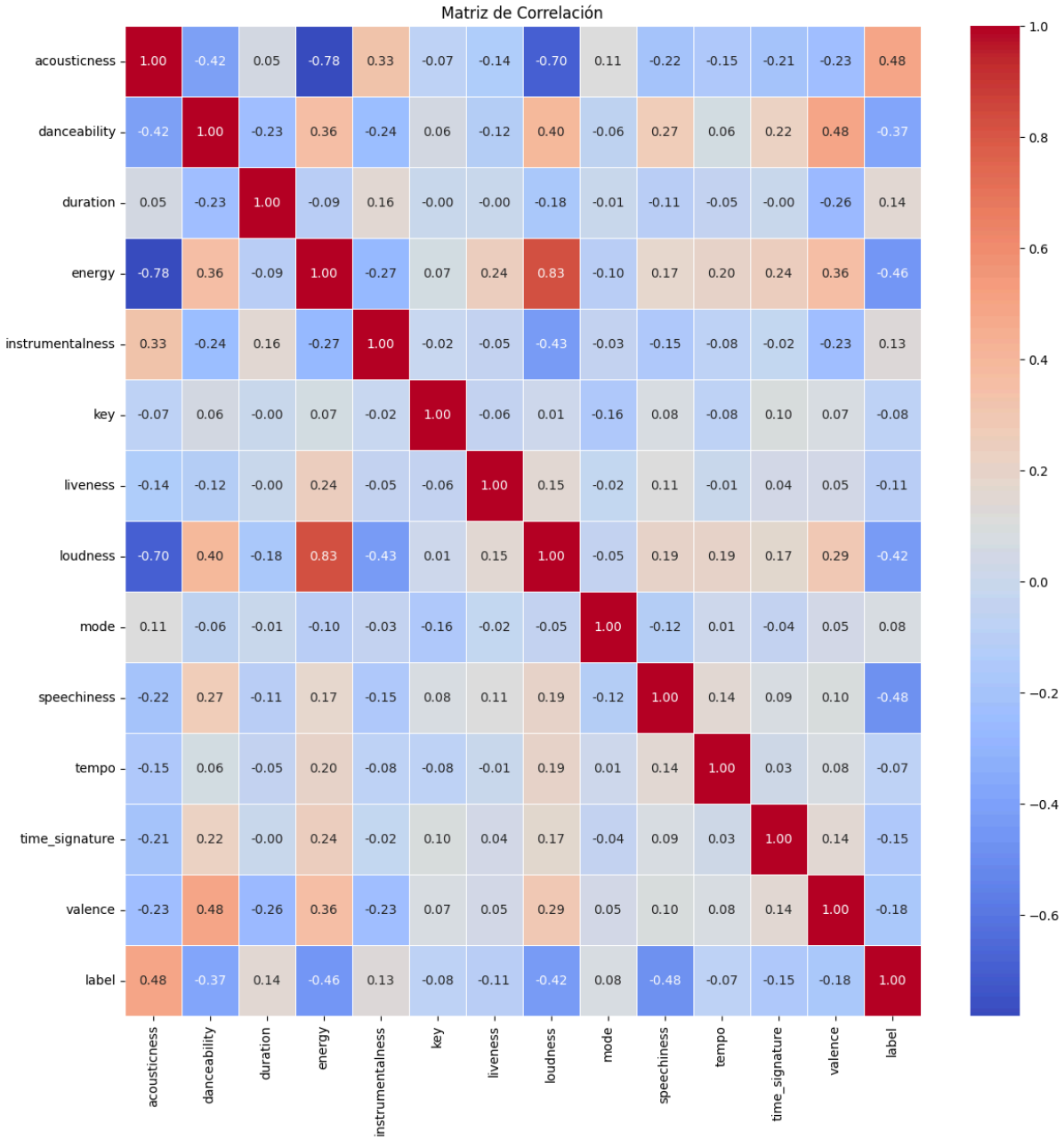

Figura 7 - Matriz de correlación

Analizando la matriz, se observa una fuerte correlación (de a pares) entre las variables "energy", "loudness" y "accousticness". A la hora de pensar en generar un modelo, podría ser conveniente prescindir de dos de estas variables, quedándose solo con "energy". Esto es debido a que se observa que a medida que "energy" aumenta, también lo hace "loudness", y por el contrario, decrece "accousticness". Es decir, la información representada por esas 3 variables puede ser aportada

prácticamente en su totalidad solo por "energy". Esta conclusión se ratifica al observar el pairplot, y analizar las relaciones entre las variables mencionadas.

Antes de entrenar un modelo de aprendizaje automático, se deberían descartar aquellas variables que no parecen tener relación con la salida. En general, un método para observar eso es la matriz de correlación. Esta muestra la relación lineal que existe entre dos variables. En particular, utilizando el coeficiente de Pearson. Las variables con mayor correlación a la salida son:

- Accousticness
- Danceability
- Energy
- Loudness
- Speachiness

Sin embargo, por lo explicado anteriormente, se observa una fuerte correlación entre Energy, Loudness y Accousticness. Pudiendo entonces utilizar por ejemplo:

- Danceability
- Energy
- Speachiness

Imputación de datos faltantes
Usando el método .info() se puede conocer más información de las columnas, donde se muestra que no hay datos faltantes.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 750 entries, 0 to 749
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   acousticness      750 non-null    float64
 1   danceability      750 non-null    float64
 2   duration          750 non-null    int64  
 3   energy            750 non-null    float64
 4   instrumentalness  750 non-null    float64
 5   key               750 non-null    int64  
 6   liveness          750 non-null    float64
 7   loudness          750 non-null    float64
 8   mode              750 non-null    int64  
 9   speechiness       750 non-null    float64
 10  tempo             750 non-null    float64
 11  time_signature    750 non-null    int64  
 12  valence           750 non-null    float64
 13  label             750 non-null    int64  
dtypes: float64(9), int64(5)
memory usage: 82.2 KB
```

Figura 8 - Uso del método .info()

Usando el método .isna() seguido de .sum() se puede obtener la cantidad de datos faltante para cada una de las features que se observa que es cero.

```
acousticness        0
danceability        0
duration            0
energy              0
instrumentalness    0
key                 0
liveness            0
loudness            0
mode                0
speechiness         0
tempo               0
time_signature      0
valence             0
label               0
dtype: int64
```

Figura 9 - Uso del método .isna().sum()

Como no tiene datos faltantes, no es necesario aplicar técnicas de imputación. Pero con el objetivo de poner en práctica los conocimientos adquiridos en la asignatura, se va a simular la generación de datos faltantes y aplicar técnicas de imputación para posteriormente realizar un análisis comparativo de sus resultados.

Se simulan datos faltantes usando las variables de mayor importancia del dataset (danceability, energy, peachiness y label). Para ello se va a suponer que los datos faltantes son del tipo Missing Completely At Random (MCAR), comenzando con un missing ratio de 0.8. Se van a usar métodos del tipo univariado como imputación por media y del tipo multivariado como KNN y MICE. En la siguiente tabla se observan los resultados obtenidos:

Imputer		Missing ratio	MSE	Score
Mean		0.8	0.028	0.631
KNN	K = 1	0.8	0.037	0.626
	K = 3		0.030	0.630
	K = 5		0.028	0.629
	K = 7		0.028	0.631
	K = 9		0.028	0.634
	K = 11		0.028	0.631
MICE	None	0.8	0.031	0.631
	2		0.031	0.631
	1		0.030	0.624

Tabla 3 - Comparación de técnicas de imputación con missing ratio igual a 0.8

Además de la comparación tabular se pueden usar diagramas de caja y bigotes para comparar entre la misma técnica de imputación.

Se observa para mean el diagrama de caja obtenido.

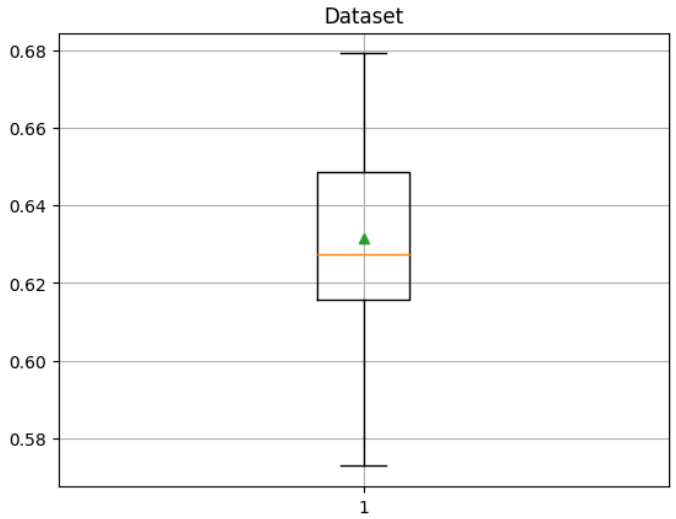

Figura 10 - Box and whisker plot para el imputador mean con missing ratio de 0.8

Para KNN se prueban diferentes valores del parámetro K, siendo estos 1, 3, 5, 7, 9 y 11 respectivamente al diagrama. Donde se observa que el mejor resultado obtenido fue el de K = 9 en el formato tabular pero este tiene mucho outliers, por lo que se podría elegir otro con un buen desempeño como K = 7 o K = 11, lo cual se observa gracias al diagrama de caja y bigotes. Si tuviéramos que decidir entre estos últimos dos convendría K = 7 ya que al tener menor valor de K es menos costoso que K = 11.

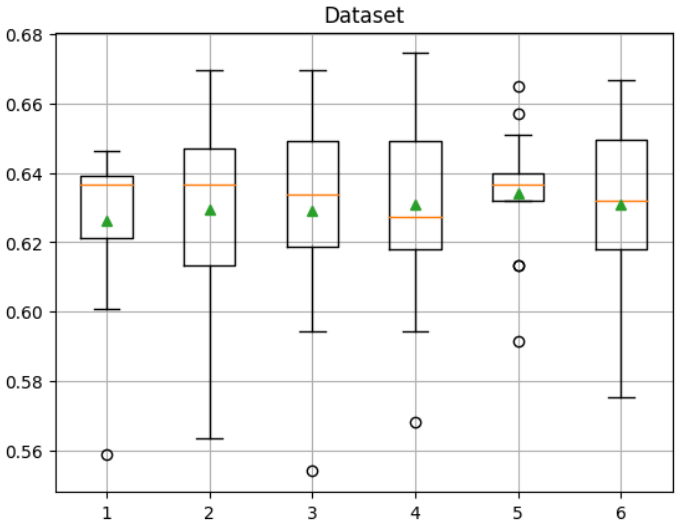

Figura 11 - Box and whisker plot para el imputador KNN con missing ratio de 0.8

Para MICE se observa que se obtuvieron resultados similares cambiando el parámetro n-nearest-features, el cuál determina el número de otras features usadas para estimar los datos faltantes de cada columna, algunos con mayor MSE y menor score y viceversa.

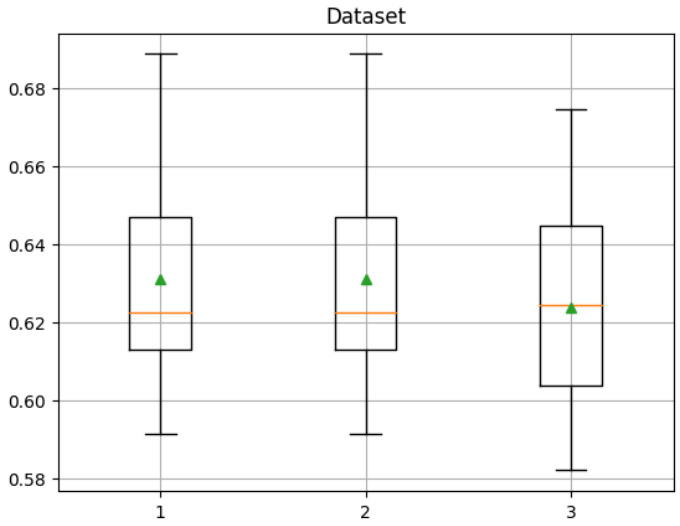

Figura 12 - Box and whisker plot para el imputador MICE con missing ratio de 0.8

Si bien todos dieron un valor bajo de score, hay un empate entre KNN y Mean, pero debido al costo computacional convendría usar Mean.

Ahora se usará un missing ratio menor para ver cómo baja el MSE y aumenta el score para cada uno de estos imputadores:

Imputer		Missing ratio	MSE	Score
Mean		0.1	0.004	0.732
KNN	K = 1	0.1	0.007	0.741
	K = 3		0.006	0.722
	K = 5		0.005	0.728
	K = 7		0.004	0.738
	K = 9		0.004	0.710
	K = 11		0.004	0.724
MICE	None	0.1	0.004	0.734
	2		0.004	0.734
	1		0.004	0.741

Tabla 4 - Comparación de técnicas de imputación con missing ratio igual a 0.1

Para el caso del imputador mean se observa el diagrama obtenido. Tiene un mejor MSE y score que el caso anterior porque tenemos menores datos faltantes.

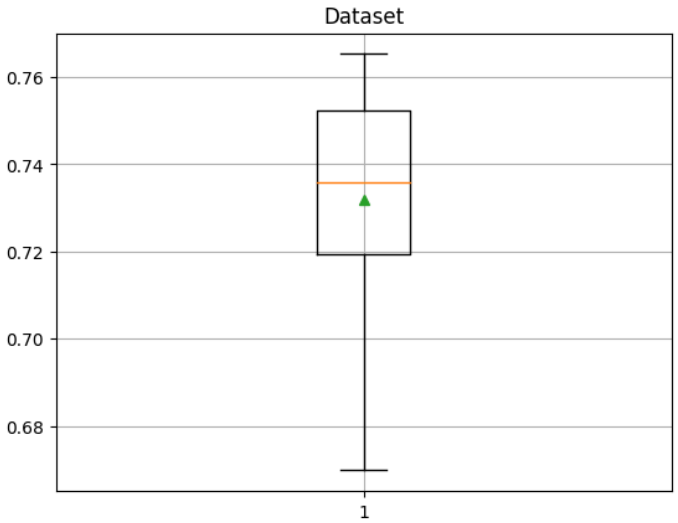

Figura 13 - Box and whisker plot para el imputador mean con missing ratio de 0.1

Imputando con KNN se observa que el que mejor resultado obtuvo fue el de K=7.

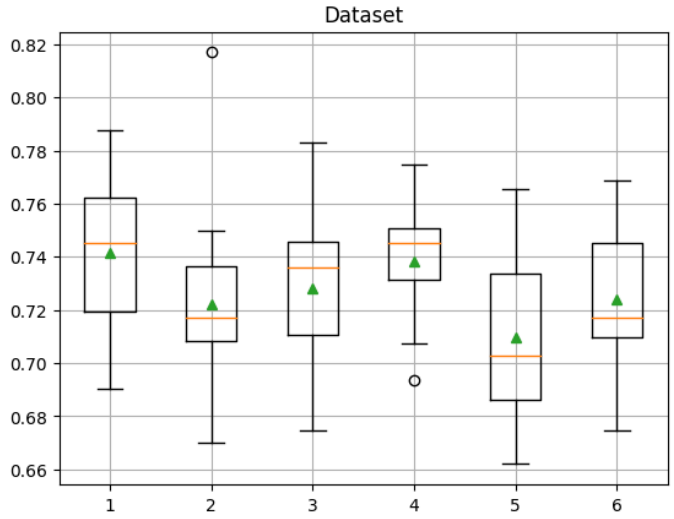

Figura 14 - Box and whisker plot para el imputador KNN con missing ratio de 0.1

En cuanto a MICE se observan que los 3 casos fueron muy similares en cuanto a MSE y scores, siendo ligeramente mejor el tercer caso (n-nearest-features = 1).

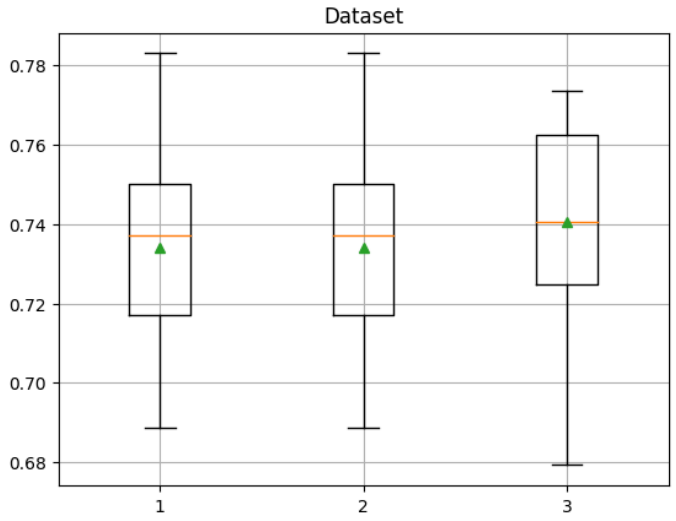

Figura 15 - Box and whisker plot para el imputador MICE con missing ratio de 0.1

Comparando los 3 métodos el que obtuvo un mejor rendimiento en este caso fue KNN.

Transformación

Para la transformación de los datos se utilizó Yeo-Johnson, dado que permite normalizar las variables incluso con presencia de valores negativos.

Los datos luego de aplicar la transformación a las variables numéricas:

	acousticness	danceability	duration	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	label
count	7.500000e+02	7.500000e+02	7.500000e+02	7.500000e+02	7.500000e+02	750.000000	7.500000e+02	7.500000e+02	750.000000	7.500000e+02	7.500000e+02	750.000000	7.500000e+02	750.000000
mean	1.326346e-16	4.014566e-16	2.439530e-16	1.989520e-16	2.842171e-17	4.829333	-4.547474e-16	-8.029133e-16	0.741333	2.297422e-16	-1.707671e-15	3.902667	-3.327708e-16	0.602667
std	1.000667e+00	1.000667e+00	1.000667e+00	1.000667e+00	1.000667e+00	3.636001	1.000667e+00	1.000667e+00	0.438194	1.000667e+00	1.000667e+00	0.400091	1.000667e+00	0.489673
min	-1.248611e+00	-2.481447e+00	-4.169952e+00	-1.960501e+00	-4.726283e-01	0.000000	-2.209135e+00	-2.506314e+00	0.000000	-1.285083e+00	-2.756005e+00	1.000000	-2.068207e+00	0.000000
25%	-1.035304e+00	-7.187568e-01	-4.869987e-01	-7.593564e-01	-4.726283e-01	1.000000	-7.613741e-01	-6.142714e-01	0.000000	-8.148002e-01	-7.242840e-01	4.000000	-8.236695e-01	0.000000
50%	-1.022526e-01	-1.195300e-03	-1.836481e-03	4.994525e-02	-4.722616e-01	5.000000	-2.666817e-01	1.181732e-02	1.000000	-4.084284e-01	6.366320e-02	4.000000	-2.012908e-02	1.000000
75%	1.039521e+00	6.744885e-01	4.422603e-01	8.334856e-01	-3.931029e-01	8.000000	9.132423e-01	6.769223e-01	1.000000	8.421253e-01	6.683866e-01	4.000000	7.960884e-01	1.000000
max	1.528997e+00	2.540577e+00	5.160180e+00	1.813059e+00	2.427739e+00	11.000000	1.995320e+00	4.126633e+00	1.000000	1.887941e+00	2.541564e+00	5.000000	1.892270e+00	1.000000

Figura 16 - Resumen tras aplicar transformación de Yeo-Johnson

Donde se puede ver que las mismas pasan a tener media 0 y varianza unitaria.

Además, si se realizan pair plots entre las variables de alta correlación, se observa cómo las mismas cambian su distribución.

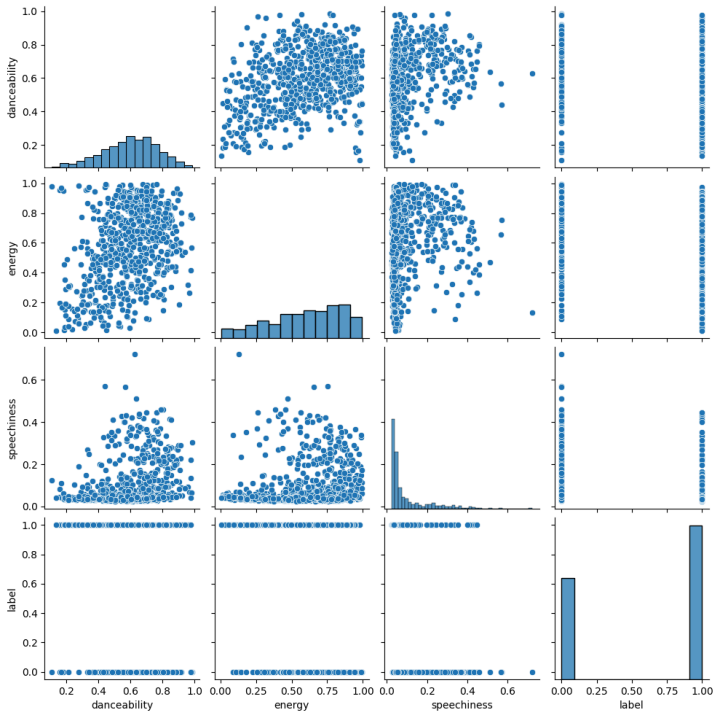

Figura 17 - Pair plot previo a la transformación

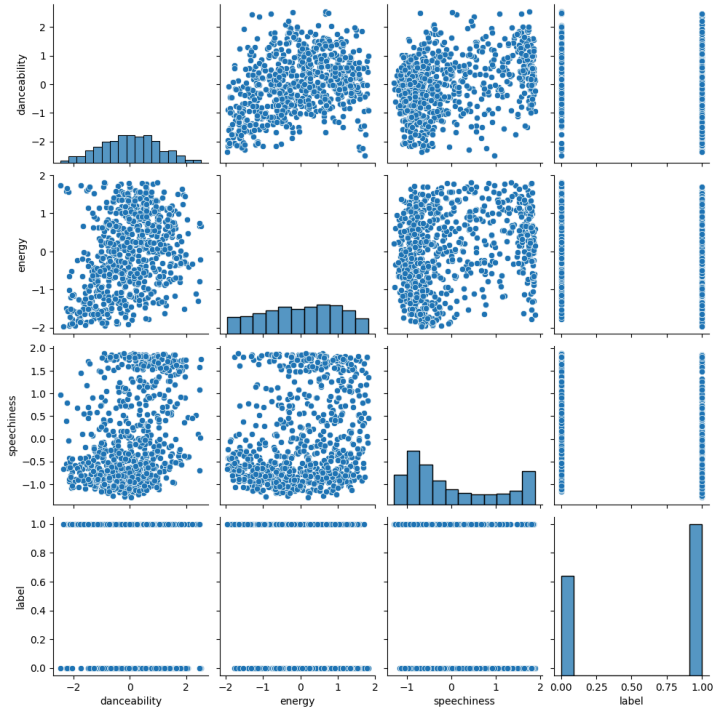

Figura 18 - Pair plot luego de la transformación

4. Entrenamiento de modelos (opcional)

Se propone el uso de los siguientes modelos para este problema de clasificación:
- Regresión Logística
- K-means
- Bosques aleatorios

Para realizar estos modelos es fundamental una buena preparación de los datos mediante la limpieza de los mismos, la codificación de variables categóricas y el escalamiento de características numéricas, puede mejorar la capacidad del modelo.
En el caso de K-means, la elección de la métrica de distancia y la escala de las características pueden afectar significativamente el rendimiento del modelo.
En bosques aleatorios, la selección de características y el manejo de la información faltante pueden afectar la construcción del árbol y la interpretación de las reglas de decisión.

4.1 Separación de datos

Los datos se separaron utilizando una semilla aleatoria. Un 75% de los datos se tomaron como datos de entrenamiento y un 25% datos de test.

4.2 Evaluación de resultados

Dado que se trata de un caso de clasificación, se determina la calidad del modelo utilizando el índice f1.
Este índice se puede obtener realizando:

$$F1 = \frac{2TP}{2TP + FP + FN}$$

Donde TP representa los correctamente anotados, FP los falsos positivos, y FN los falsos negativos.
Dependiendo del modelo entrenado, se decidieron ciertos hiperparametros:

- K-means: Para todos los casos K=2 dado que sabemos que la salida es un label binario. Inicialización ‘k-means++’ y se prueban un total de 100 inicializaciones diferentes.
- Bosques Aleatorios: Para evitar el overfitting, se utiliza una penalización para árboles con mucha complejidad a partir del parámetro ‘cpp_alpha’ que es determinado en 0.1.

Método	F1-score (train)	F1-score (test)
Regresión logística	0.716	0.715
Regresión logística con selección de variables	0.827	0.834
Regresión logística con transformación	0.869	0.816
Regresión logística con selección y transformación de variables	0.831	0.831
K-means	0.391	0.348
K-means con selección de variables	0.412	0.486

K-means con transformación	0.581	0.527
K-means con selección y transformación de variables	0.327	0.374
Bosques aleatorios	0.839	0.838
Bosques aleatorios con selección de variables	0.824	0.827
Bosques aleatorios con transformación	0.828	0.835
Bosques aleatorios con selección y transformación de variables	0.824	0.827

Tabla 5 - Comparación del F1-score para los distintos métodos de clasificación

En todos los casos la transformación fue realizada utilizando Yeo-Johnson.
En todos los casos la selección de variables fue solo utilizando las columnas:

- Danceability
- Energy
- Speachiness

Por las razones explicadas en pasos anteriores.

A partir de los datos, se puede determinar para cada algoritmo cuál resultó el mejor modelo y por qué. Los marcados en verde son aquellos que presentan mejor F1-score en el conjunto de test.

- Regresión logística: La selección de variables la ayuda mucho. Para ambos casos donde hubo selección de variables, el modelo tuvo su mejor performance.
- K-means: Se sabe que es un algoritmo que utiliza conceptos de distancia. Estos se tienden a distorsionar mucho si las variables no están correctamente normalizadas. Por esto es que el mejor modelo es aquel que incorpora las transformaciones que normalizan el dataset.
- Bosques Aleatorios: Resulta el más robusto. En nuestros casos, independientemente de la acción que tomamos, se comportó muy bien.

Conclusión

Se pudo aplicar múltiples métodos de análisis de datos. Esto tiene mucha importancia para poder describir los mismos, pero además como técnicas de preparación para entrenar modelos de machine learning. Esto último es transversal a toda la especialización.
También se pudo simular la falta de datos y aplicar técnicas de imputación de datos.
Se verificó, que se mejora la precisión de los algoritmos en la etapa cuatro de nuestro proyecto, donde K-means y la regresión logística lograron una mejora considerable a partir de estos instrumentos.