Predicción de la Popularidad de Canciones

Group 03, ID Kaggle: Grupo 03

Garcia Garcia, Bernat – bernat.garcia.garcia@estudiantat.upc.edu (Doble grado Economía-Estadística, UB-UPC)

Maria Montes, Iker – ikermaria.estudios@gmail.com (Doble grado Economía-Estadística, UB-UPC)

Rota, Davide - davide.rota@estudiantat.upc.edu (Mathematical engineering at PoliMi, currently at MESIO, UPC)

Tobella Jacomet, Pol - ptobelja7@alumnes.ub.edu (Doble grado Economía-Estadística, UB-UPC)

GitHub link: https://github.com/benet1one/Mineria

Entendimiento Del Negocio

Problema

- Predecir la popularidad de canciones en Spotify usando atributos musicales y metadatos.
- Minimizar el error (MAPE) para obtener predicciones estables y útiles.

Objetivo Principal

- Desarrollar un modelo de regresión que prediga *song_popularity*.

Objetivos Secundarios

- Identificar qué variables musicales influyen más en la popularidad.
- Detectar problemas de calidad del dataset.
- Proponer mejoras futuras (nuevas variables, ajustes de metodología).

Fuentes Externas

- Kaggle competition: https://www.kaggle.com/competitions/prediccion-de-la-popularidad-de-canciones/overview
- Spotify Web API (atributos musicales): https://developer.spotify.com/documentation/web-api

Entendimiento de los Datos

Dataset

- 13.186 registros
- 15 variables (12 cuantitativas, 3 cualitativas)
- Fuente: *train.csv* (Kaggle)
- Alcance: información técnica y musical de tracks

Calidad de Datos

- Sin missing values relevantes.
- Distribuciones correctas para variables ratio (0–1).
- Posibles outliers en loudness, tempo y duration_ms.
- Variables categóricas con valores completos y consistentes.

Fuentes externas (datos)

- train.csv (Kaggle): https://www.kaggle.com/competitions/prediccion-de-la-popularidad-de-canciones/overview
- Documentación de atributos Spotify API: https://developer.spotify.com/documentation/web-api

Variable	Tipo	Descripción	Rango
song_popularity	Num	Popularidad objetivo	0–100
danceability	Num (0–1)	Ritmo y facilidad para bailar	0–1
energy	Num (0–1)	Intensidad y actividad	0–1
valence	Num (0–1)	Positividad emocional	0–1
liveness	Num (0–1)	Probabilidad de performance en vivo	0–1
acousticness	Num (0–1)	Probabilidad de ser acústica	0–1
instrumentalness	Num (0–1)	Probabilidad de no tener voz	0–1
speechiness	Num (0–1)	Cantidad de palabras habladas	0–1
loudness	Num	Nivel dB	-60–0
tempo	Num	BPM	variable
duration_ms	Num	Duración en ms	variable
mode	Cat	Modo mayor/minor	0–1
key	Cat	Tonalidad	-1–11
time_signature	Cat	Compás	3–7
id	Num	Identificador	1–13186

Framework Metodológico

1. Business Understanding

- Definición problemas, KPIs, alcance de predicción, alineación de objetivos de negocio con objetivos técnicos.
- Responsable: **Iker**

2. Data Understanding

- Exploración inicial, revisión metadatos, análisis de datos, identificación preliminar de anomalías y variables clave.
- Responsable: **Bernat, Iker, Davide, Pol**

3. Data Preparation

- Limpieza, transformación, tratamiento de missing/outliers, normalizaciones, encoding, feature engineering.
- Responsable: **Bernat, Iker**

4. Modeling

- Construcción de reglas tipo Market Basket, interpretación de patrones de co-ocurrencia.
- Selección de algoritmos, definición de particiones training/test, selección de predictores finales.
- Responsable: **Davide, Pol**

5. Evaluation

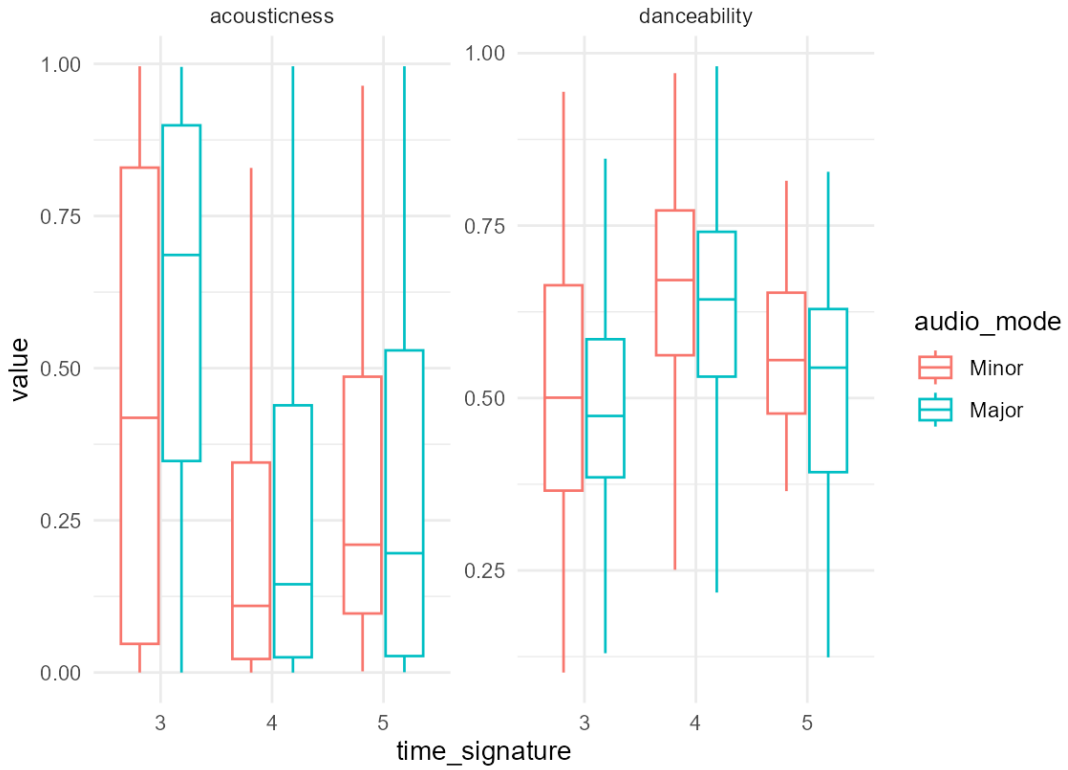
- Comparación entre modelos, selección del mejor candidato, interpretabilidad de variables.
- Responsables: **Bernat, Davide**

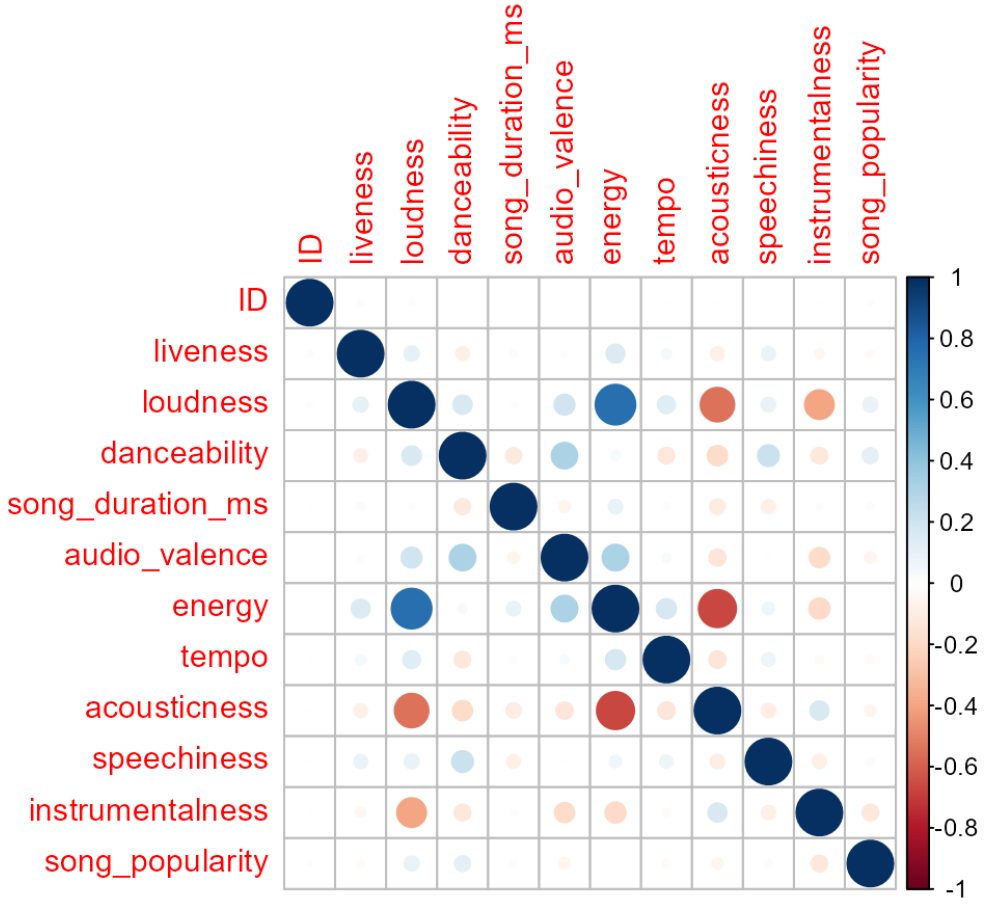
6. Deployment (No incluido en alcance actual)

- Se deja conceptual por completitud del framework.
- Responsable: N/A

Análisis Exploratorio

Existen muchas dependencias entre variables explicativas.
Estas són algunas de ellas.





Feature Engineering

Escala

Las variables de tonalidad *(key)* i modalidad *(audio_mode)* están muy relacionadas. La escala *(key_signature)* se puede obtener a partir de estas dos variables, calculando el Mayor Relativo:

- `Modalidad Mayor  =>  Escala = Tonalidad`
- `Modalidad Menor  =>  Escala = MayorRelativo(Tonalidad)`

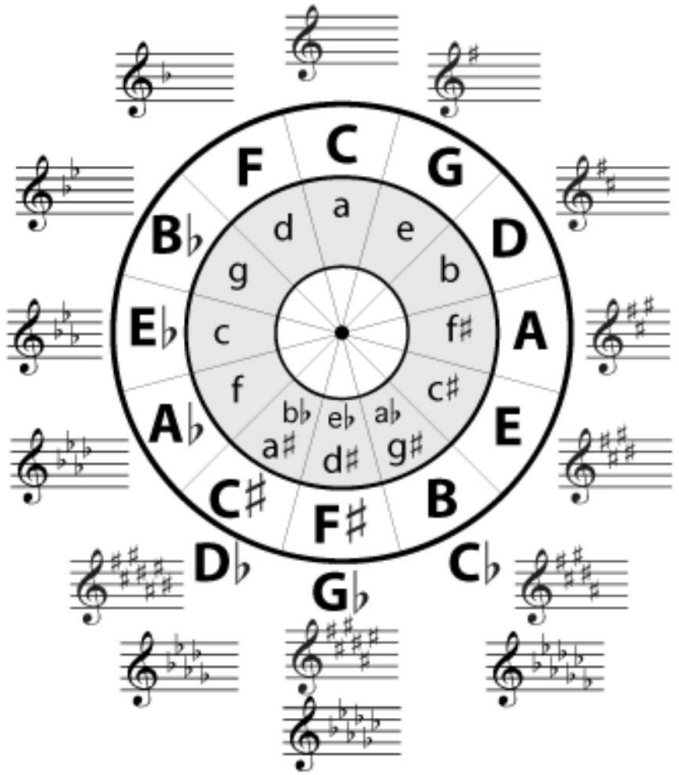
Probabilidad de tener Modalidad Mayor

Dado que algunas escalas són mucho más comunes que otras, es posible adivinar la modalidad a partir de la tonalidad. Por ejemplo, B Menor es mucho más frequente que B Mayor. Por lo tanto, si una canción está en B, es más probable que la modalidad sea Menor.

A partir de esto hemos creado la variable *prob_major,* usando las proporciones empíricas de nuestros datos.

- `Modalidad == NA  =>  prob_major = Pr(Mayor | Tonalidad)`
- `Modalidad != NA  =>  prob_major = [1 si es Mayor, 0 si es Menor]`

Relación entre
relativo menor (minúsculas)
relativo mayor (mayúsculas)
i escala.



Imputación en dos pasos

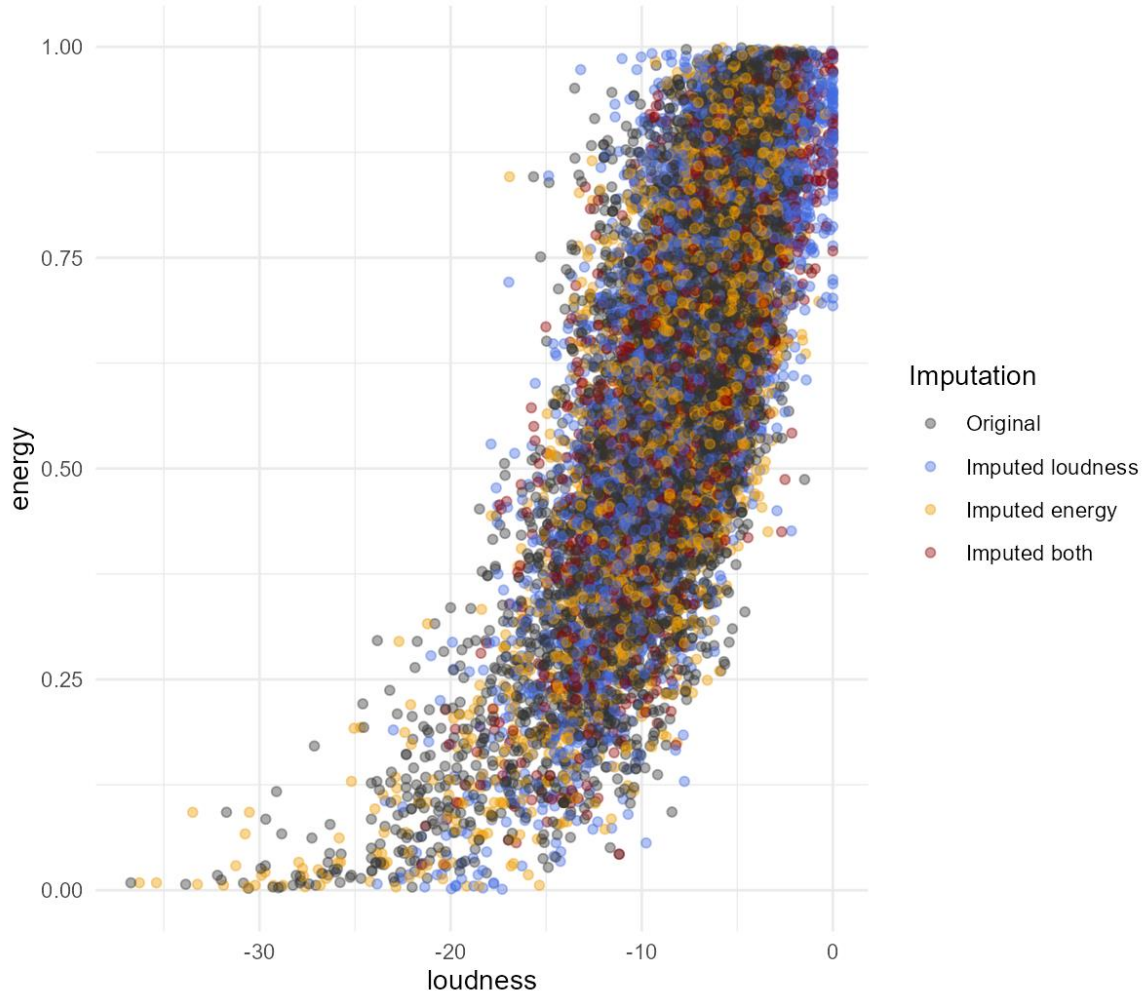
1. A partir del entendimiento del negocio i de **datos empíricos**, imputamos la modalidad.

```
key -> prob_major -> audio_mode
```

2. Imputamos del resto de valores con **MICE**, usando una matriz de predictores personalizada i diferentes métodos:

- Predictive Mean Matching (pmm)
- Ranfom Forest (rf)
- Regressión Lasso (lasso.norm)

Uno de los gráficos para verificar la imputación

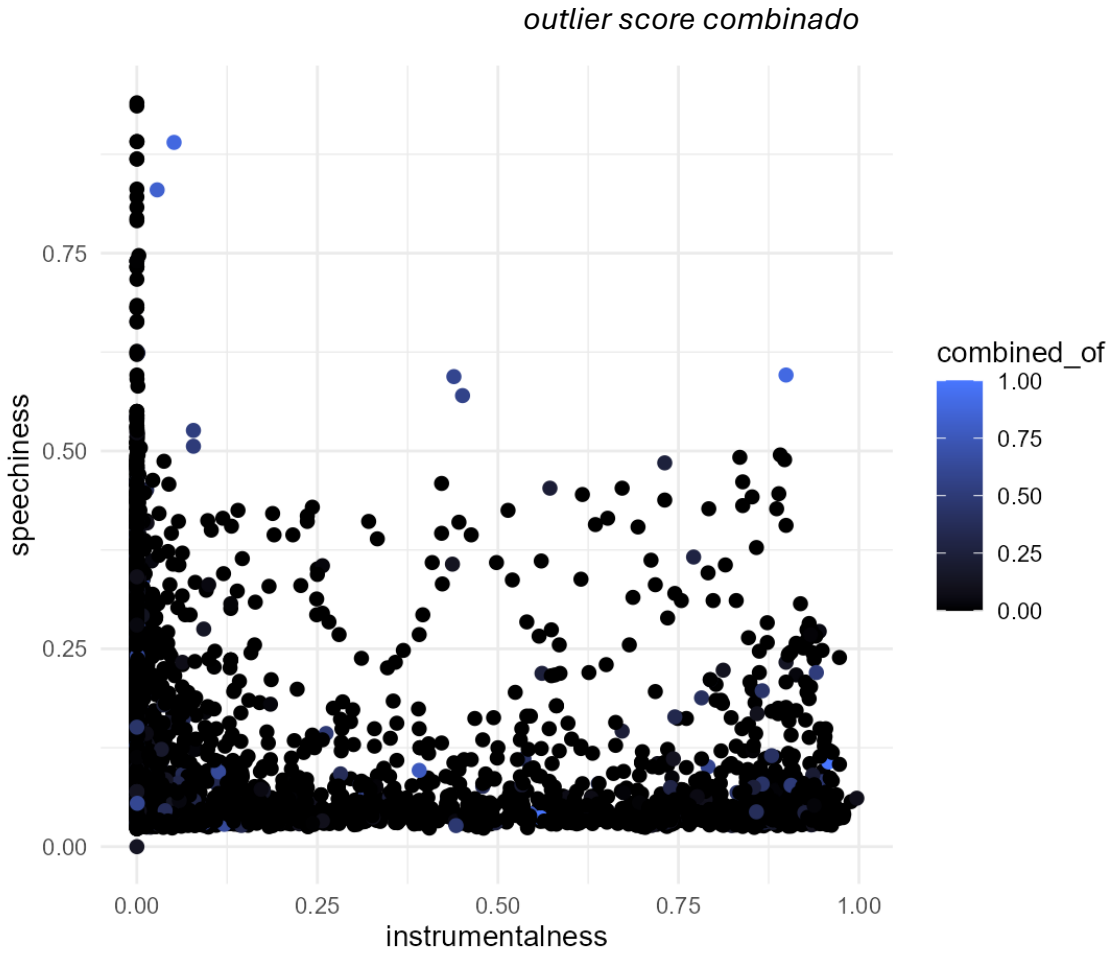


Outliers

Para aprovechar la **fuerte relación** entre variables, se ha usado **ALSO** (Attribute Wise Learning Score for Outliers) para obtener un *score* para la mayoria de las variables.

También se han detectado anomalias en la relación **instrumentalness-speechiness**, que se han calificado como outliers, pues una canción no puede ser instrumental i contener voces a la vez.

Se han combinado los dos *scores* en uno, para usarlo como **peso** para los métodos de regresión.



Associtation Rules

- Al realizar un análisis de *association rules* con nuestros datos pretendíamos identificar patrones entre elementos. Para ello primeramente debíamos transformar nuestros datos para obtener una base transaccional.
- Para ello dividimos todas las variables numéricas en 3 intervalos de igual dimensión absoluta y eliminamos variables (*prob_major* y *outlier_weight*) que anteriormente habíamos creado pero que no aportan información en este análisis. A partir de estos cambios convertimos los datos al formato adecuado, una matriz transaccional. Como resultado cada transacción (canción) queda reflejada en el formato de la derecha:
- En relación a la creación de reglas, tras múltiples combinaciones hemos decidido conservar los dos siguientes análisis:

```
{ID=1,
 liveness=Low,
 loudness=High,
 danceability=High,
 song_duration_ms=Low,
 time_signature=4,
 audio_valence=High,
 energy=Medium,
 tempo=Low,
 acousticness=Low,
 speechiness=Low,
 key=B,
 instrumentalness=Low,
 audio_mode=Major,
 song_popularity=Medium,
 duration_qual=Short}
```

-El primero se determina con un soporte y una confianza mínima de 0.05 y 0.7 respectivamente. Tras eliminar reglas redundantes resulta en un total de 4715.

```
set of 4715 rules

rule length distribution (lhs + rhs):sizes
   1    2    3    4    5
   6  154  746 1699 2110
```

-El segundo se determina con un suporte y una confianza mínima de 0.35 y 0.9 respectivamente. Tras eliminar reglas redundantes resulta en un total de 292

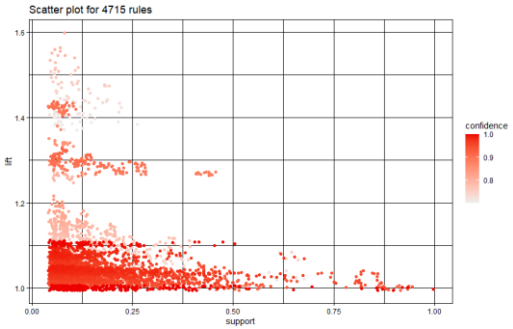
```
set of 292 rules

rule length distribution (lhs + rhs):sizes
  1   2   3   4   5
  5  53 107  94  33
```

Associtation Rules

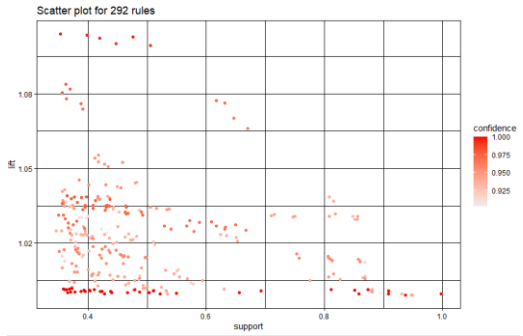
·Como hemos comentado antes, el primer análisis parte de niveles menos exigentes de soporte y confianza. Podemos observar que las reglas con mayor *lift* explican que la variable *energy* tome valores altos. Si nos fijamos, se tratan de reglas bastante evidentes cómo que canciones con: niveles de *loudness* altos, *danceability* medios, sentimientos positivos y que no son acústicas llevan asociadas un carácter energético.

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{loudness=High, danceability=Medium, audio_valence=High, acousticness=Low}	=> {energy=High}	0.07555761	0.8137988	0.09284557	1.598362	979
[2]	{audio_valence=High, tempo=Medium, acousticness=Low, duration_qual=Medium}	=> {energy=High}	0.06135680	0.7965932	0.07702400	1.564568	795
[3]	{danceability=Medium, time_signature=4, audio_valence=High, acousticness=Low}	=> {energy=High}	0.07494019	0.7959016	0.09415760	1.563210	971
[4]	{danceability=Medium, song_duration_ms=Low, audio_valence=High, acousticness=Low}	=> {energy=High}	0.07632940	0.7899361	0.09662731	1.551493	989
[5]	{danceability=Medium, audio_valence=High, acousticness=Low}	=> {energy=High}	0.07632940	0.7893057	0.09670448	1.550255	989



·Para el segundo análisis llegamos a una conclusión similar. Se trata de reglas evidentes que no se enfocan en la variable objetivo *popularity*. En este caso las reglas con mayor lift explican niveles altos de *loudness*.

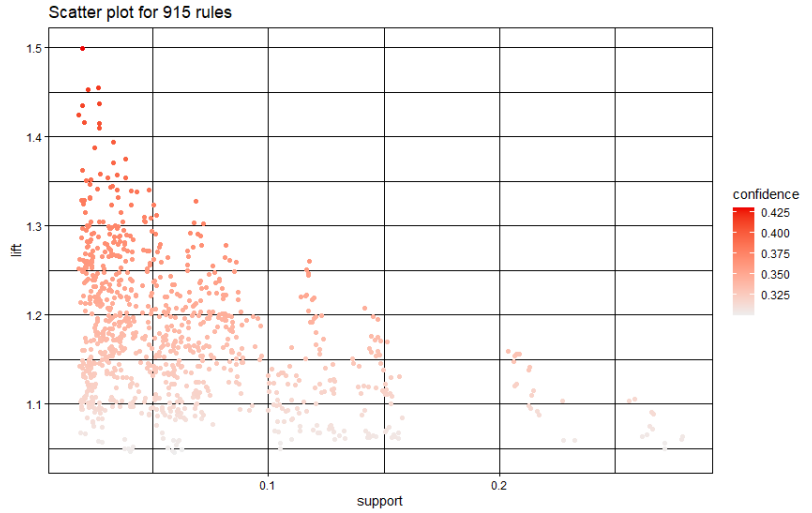
	lhs	rhs	support	confidence	coverage	lift	count
[1]	{liveness=Low, energy=High, acousticness=Low, instrumentalness=Low}	=> {loudness=High}	0.3508528	0.9962744	0.3521649	1.103593	4546
[2]	{liveness=Low, energy=High, instrumentalness=Low}	=> {loudness=High}	0.3999383	0.9961553	0.4014818	1.103461	5182
[3]	{energy=High, acousticness=Low, instrumentalness=Low}	=> {loudness=High}	0.4175349	0.9961333	0.4191557	1.103437	5410
[4]	{energy=High, instrumentalness=Low}	=> {loudness=High}	0.4745697	0.9961121	0.4764220	1.103413	6149
[5]	{energy=High, acousticness=Low}	=> {loudness=High}	0.4455507	0.9934607	0.4484834	1.100476	5773



Associtation Rules

- Para finalizar este apartado, hemos realizado también un análisis enfocado a la popularidad, nuestra variable objetivo. Para obtener reglas que explicaran la popularidad, en cualquier nivel, hemos tenido que realizar un análisis con un soporte y confianza notablemente bajos, concretamente de 0.02 y 0.3 respectivamente. Aunque se podrían encontrar reglas con mayor confianza, el soporte tendría que ser aún mucho menor.
- El resultado de este análisis nos muestra que las reglas que explican niveles altos de popularidad con *lift* alto tienden a incluir: *danceability*=High, *acousticness*=Low, *key*=Db, *instrumentalness*=Low y *loudness*=High. Aún así debemos ser cautos al usar estos resultados pues están asociados a niveles de soporte cercanos a 0.02 y valores de confianza levemente superiores a 0.4.

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{danceability=High, acousticness=Low, key=Db, instrumentalness=Low}	=>	{song_popularity=High}	0.02106969	0.4305994	0.04893108	1.498194	273
[2]	{loudness=High, danceability=High, key=Db, instrumentalness=Low}	=>	{song_popularity=High}	0.02500579	0.4169884	0.05996759	1.450838	324
[3]	{loudness=High, danceability=High, acousticness=Low, key=Db}	=>	{song_popularity=High}	0.02106969	0.4167939	0.05055183	1.450161	273
[4]	{danceability=High, time_signature=4, acousticness=Low, key=Db}	=>	{song_popularity=High}	0.02099251	0.4127466	0.05086054	1.436079	272
[5]	{danceability=High, time_signature=4, key=Db, instrumentalness=Low}	=>	{song_popularity=High}	0.02500579	0.4122137	0.06066219	1.434225	324



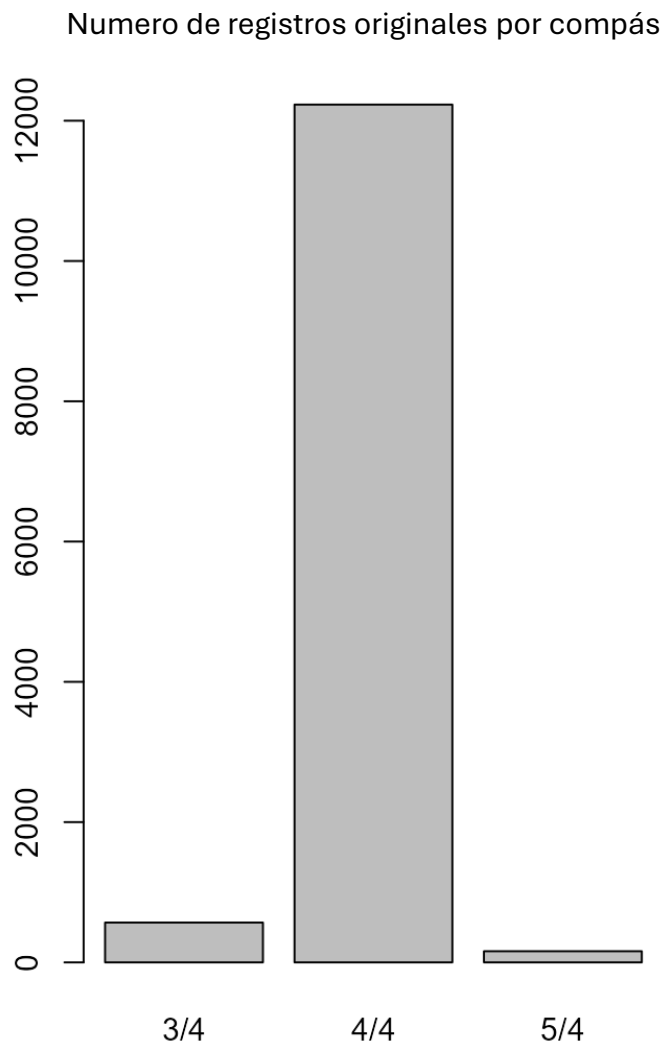
Preparación para el modelado

En general, teniendo en cuenta la cantidad de datos i el KPI usado, hemos decidido dividir los datos entre 70% de entrenamiento i 30% de *testing*.

La base de datos está **altamente desbalanceada** según el compás (*time_signature*). Por lo tanto, al dividir los datos entre *training* i *testing*, es necesario que en todas las divisiones estén presentes todos los compases.

Para aquellos métodos que necesiten datos balanceados, se ha creado una base de datos con *oversampling*. En particular, se ha usado una técnica inspirada en **SMOTE** (Synthetic Minority Over-sampling Technique), que interpola las variables numéricas entre observaciones cercanas.

La base de datos sobremuestrada contiene 12.000 registros por cada compás.



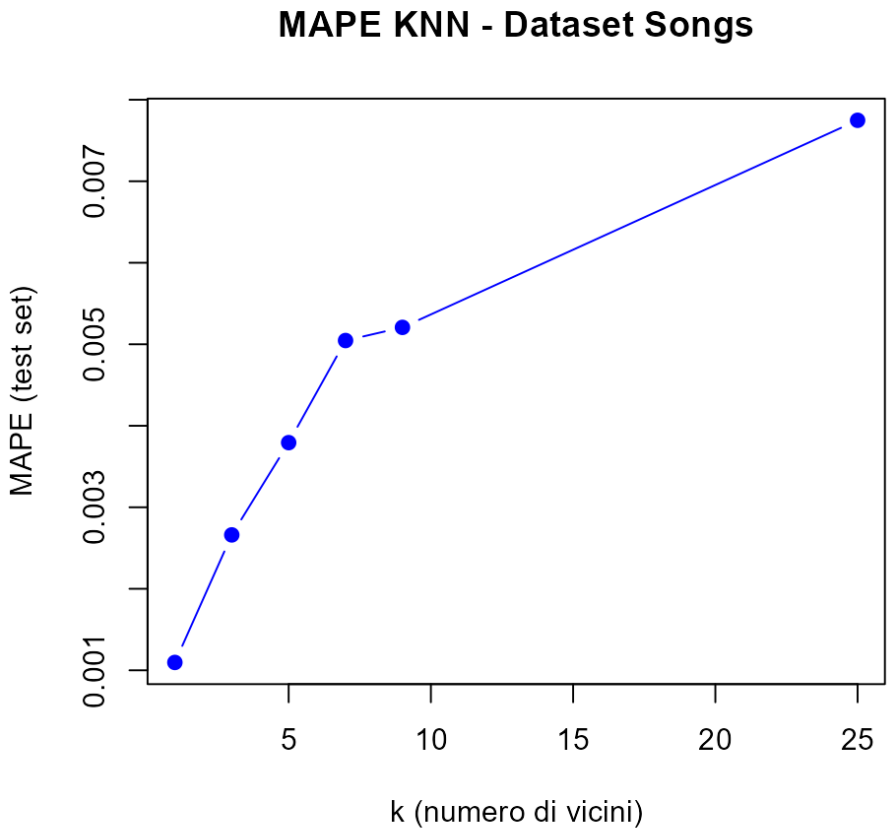
KNN

El primer método aplicado fue el de los K Vecinos Más Cercanos (KNN) en un contexto de regresión. Inicialmente, el conjunto de datos depurado se dividió en un conjunto de entrenamiento y otro de prueba para obtener resultados fiables y evaluar la capacidad del modelo para generalizar a datos no vistos. Para garantizar un buen rendimiento del KNN, todas las características se convirtieron a formato numérico y se estandarizaron, ya que las diferencias de escala podrían afectar significativamente el cálculo de las distancias.

La selección y ponderación de características se realizó mediante un método propio (pero efectivo). Se eliminaron dos variables.

Posteriormente, se entrenó el modelo y se utilizó para realizar predicciones, las cuales se evaluaron mediante el indicador clave de rendimiento (KPI) del proyecto, el **Error Porcentual Absoluto Medio** (MAPE), junto con la **correlación** entre los valores predichos y reales.

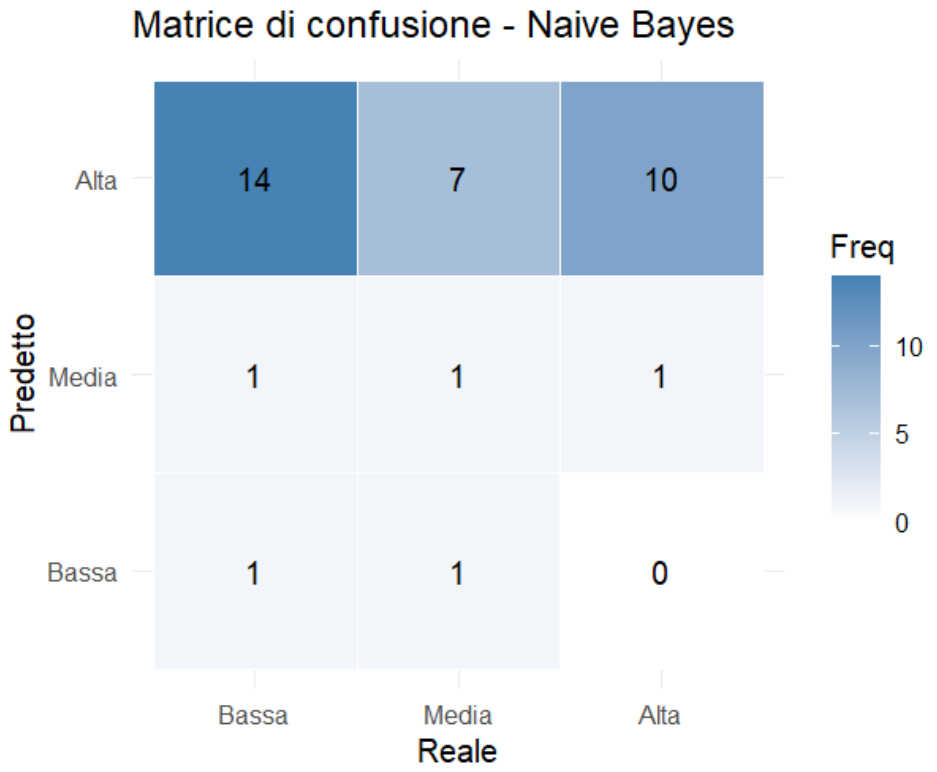
Finalmente, se optimizó el parámetro k (número de vecinos) en función de los resultados del MAPE, obteniéndose el mejor rendimiento para la **k = 1**. Esto significa que el KNN es muy eficiente computacionalmente. Los resultados para el conjunto de prueba de Kaggle fueron buenos: ocupamos el primer lugar al momento de redactar este informe.



Naive Bayes

A primera vista, el uso del algoritmo Naive Bayes parecía un enfoque forzado para predecir la popularidad de las canciones, ya que Naive Bayes está diseñado para variables categóricas, mientras que nuestra variable objetivo (la popularidad de las canciones) es continua y numérica. Para que el método fuera aplicable, transformamos la variable objetivo en tres categorías discretas que representan niveles de popularidad bajo, medio y alto. Sin embargo, tras aplicar el modelo Naive Bayes, los resultados fueron insatisfactorios: las predicciones estaban fuertemente sesgadas hacia la categoría "alta", lo que generó un resultado desequilibrado. Además, la precisión derivada de la matriz de confusión correspondiente fue bastante baja, lo que indica que Naive Bayes no es adecuado para este problema de regresión.

Accuracy: 0.33333



Conclusiones y proximos pasos

Conclusiones clave

- Tratamiento de calidad robusto: sin missings relevantes; outliers identificados con ALSO + lógica musical.
- Imputación en dos pasos validada visualmente → estabilidad en variables clave.
- EDA: dos variables eliminadas; correlaciones apoyan selección final de predictores.
- Feature Engineering mejora consistencia (prob_major, outlier_weight).
- Association Rules revela patrones, pero target continuo limita reglas sobre popularidad.
- KNN (k=1) obtiene mejor MAPE y buena correlación → método eficaz.
- Naive Bayes inadecuado → sesgo hacia “high” + baja accuracy.

Próximos pasos

- Probar modelos más avanzados: RF, Gradient Boosting, árboles.
- Evaluar reducción de dimensionalidad (PCA/UMAP) para contrastar si la variabilidad del dataset puede representarse en menos componentes sin perder precisión.
- Aplicar validación cruzada para evaluar estabilidad del MAPE.
- Desarrollar prototipo de sistema de scoring para simular escenarios.
- Diseñar pipeline reproducible + prototipo dashboard (futuro deployment).