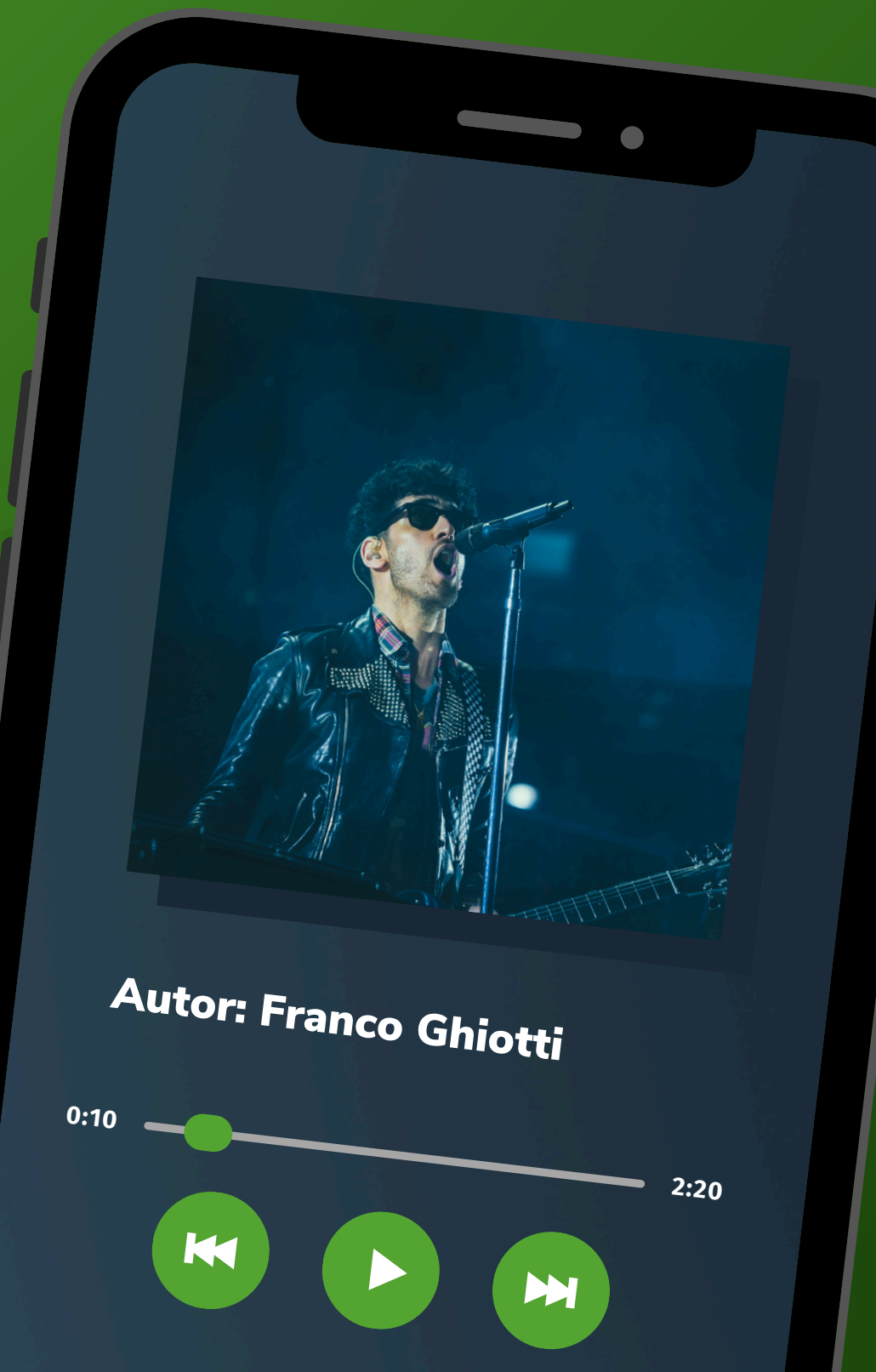
Primer Entrega Data Science II: Machine Learning para la Ciencia de Datos

Comisión 60895



Contenido

Descripción del Proyecto

Descripción de la Temática

Para el presente trabajo se tomaron datos de Kaggle, que a su vez fueron tomados de la API de Spotify. Estos datos contienen atributos descriptivos de canciones dentro de un álbum más un valor de popularidad del álbum.
El objetivo del estudio es poder predecir la popularidad de un álbum en base a diferentes atributos de algunas de sus canciones. Este estudio podría ser utilizado por artistas y discográficas para poder producir canciones con cierta popularidad target. Además, las plataformas musicales (como Spotify, iTunes, YouTube, etc) podrían beneficiarse de esta información para mejorar sus herramientas de promoción.

Base de Datos

La base de datos cuenta con 160.000 registros y 45 atributos. A continuación, se presenta una lista de los atributos:

- **ID: Variable de identificación de cada registro**
- **Name: Nombre del álbum**
- **Release Date: Fecha en que se presento el álbum**
- **Artists: Artistas del álbum**
- **Total Tracks: Numero total de canciones en el álbum**
- **Name: Nombre de la canción**
- **Duration: Duracion de la canción en ms (milisegundos)**
- **Danceability: Atributo que describe que tan bailable es la canción. Su valor esta en el rango de [0,1], donde '0' es menos bailable y '1' es lo mas bailable.**
- **Energy: Atributo que describe la intensidad de la canción. Su valor esta en el rango de [0,1], siendo '1' el valor para canciones mas energéticas**
- **Key: Tonalidad en que esta escrita la canción. Esta variable es un entero entre [0,11], donde la tonalidad Do = 0, D#=1 y asi hasta Si=11.**
- **Mode: Atributo que indica la Modalidad de la canción. Toma valor "0" si la modalidad es Menor y "1" si la modalidad es Mayor**
- **Speechiness: Atributo que detecta la presencia de discursos dentro de una canción. Su valor esta en el rango de [0,1]. Cuanto mas exclusivamente hablada sea la canción, mas cercano a 1 sera el valor de este atributo.**
- **Acousticness: Atributo que detecta si una canción es acústica. Su valor esta en el rango de [0,1]. Valores cercanos a 1 representan canciones acústicas**
- **Instrumentalness: Atributo que detecta la ausencia de voz en la canción. Su valor esta en el rango de [0,1]. Valores cercanos a 1 representan canciones completamente instrumentales.**
- **Liveness: Atributo que detecta la presencia de una audiencia en la canción. Su valor esta en el rango de [0,1]. Valores cercanos a 1 representan canciones en vivo.**
- **Valence: Atributo que busca describir la positividad musical de la canción. Su valor esta en el rango de [0,1]. Valores cercanos a 1 representan canciones mas "positivas".**
- **Tempo: Tempo estimado de la canción en pulsaciones por minuto (beats per minute BPM).**
- **Time Signature: Marca de tiempo de la canción (es la forma de especificar cuantas pulsaciones hay en un compas).**
- **Popularity: VARIABLE TARGET. Valor de popularidad del Album. Su valor esta en el rango de [0,100]**

Vale aclarar que los atributos Name – Duration – Danceability – Energy – Key – Mode – Speechiness – Acousticness – Instrulmentalness – Liveness – Valence – Tempo – Time Signature están presentes para 3 de las canciones del álbum. Si se observa la base de datos, cada uno de estos atributos aparece dividido en tres columnas, cada uno representando una canción del álbum. Por ejemplo, el atributo Name aparece como name0, name1 y name2. Es asi como se llega al valor total de 45 atributos.

Data Wrangling

Se describen a continuación las etapas del pipeline de limpieza de datos

3

Setear columna ID como index

Este paso baja el numero de columnas y genera un valor de índice que es único para cada dato.

Eliminar registros duplicados

Se encontró que, de todos los registros presentes en la base de datos, el 54.8% son duplicados. Por eso se decide eliminar los valores duplicados. Al hacer esto, se pasa a tener un total de 72.357 registros únicos.

Correr función ReduccionFeatures

Como se comento anteriormente, la base de datos cuenta con los valores descriptivos de las 3 primeras canciones del álbum. Pero algunos álbumes cuentan con solo 1 o 2 canciones. En estos casos, algunos atributos vienen con valores NAN.

Para poder reducir la cantidad de atributos, se decide crear una función que resume la información de las canciones del álbum en un solo valor por atributo. Las variables a las que se les aplica la reducción son las descriptivas:

- *Duration*
- *Danceability*
- *Energy*
- *Speechiness*
- *Acousticness*
- *Instrumentalness*
- *Liveness*
- *Valence*
- *Tempo*
- *Time Signature*

Hay 2 variables descriptivas mas que son Key y Mode. Pero estas variables, si bien toman valores numéricos, son categóricas (su valor es entero porque referencia a una categoria). Lo mismo sucede con la variable Name.

Se crea una función que reduce la cantidad de features descriptivos. Como se ve en la data, los registros con 3 o mas canciones poseen 3 columnas por feature descriptivo (ejemplo, danceability aparecerá como dance_0, dance_1 y dance_2). Para reducir esto a solo una columna por Feature, se toma el promedio de los valores de cada columna.

En el caso especial de las variables Name, Key y Mode, se toma el valor de la primer cancion.

Una vez aplicada la función, el data set pasa a tener 18 columnas.

Generar variable Target para clasificación

La variable target en este trabajo es la Popularidad, que como se explico anteriormente es un valor continuo que toma valores del 1 al 100. Para poder realizar algoritmos de

clasificación, se puede transformar la variable target Popularity en clases. Se separaran entonces los valores de la variable en 5 clases:

- Clase 1 - Popularidad baja: Valores del 1 al 20
- Clase 2 - Popularidad media-baja: Valores del 21 al 40 •
- Clase 3 - Popularidad media: Valores del 41 al 60
- Clase 4 - Popularidad media-alta: Valores del 61 al 80
- Clase 5 - Popularidad alta: Valores del 81 al 100

Llenado de valores nulos
Se verifica si en este punto existen valores nulos en el data set. Al realizar el análisis el resultado es el siguiente

```
t_key0              61
t_mode0             61
t_dur0               0
t_dance0           103
t_energy0          103
t_speech0          103
t_acous0           103
t_ins0             103
t_live0            103
t_val0             103
t_tempo0           103
t_sig0             103
popularity           0
ClasePopularidad     0
dtype: int64
```

Para poder rellenar estos valores, se utiliza la moda de cada atributo, tomando como referencia los valores del set de entrenamiento.

Pasaje de atributo Duration a segundos

El atributo Duration viene dado en milisegundos, lo cual hace que sus valores sean del orden de 105. Para evitar valores de esa magnitud, se pasan los datos a segundos, obteniendo datos en el orden de 102.

Eliminar atributo Time Signature
Los atributos Tempo y Time Signature son muy similares, ya que ambos hacen referencia a las BPM de las canciones. Por lo tanto, para no repetir información y simplificar el problema, se elimina el atributo Time Signature

Eliminar registros con Tempo igual a cero

El atributo Tempo marca los BPM de una canción. Haciendo un análisis de los datos, se encontraron registros con Tempo = 0, lo cual no es un dato coherente (no existe una canción con cero BPM). Por lo tanto, se eliminan estos valores ya que son erróneos.

Eliminar registros con Duration mayor a 360 segundos

En promedio, una canción dura alrededor de 3 minutos. Haciendo un análisis de la variable Duration, se encontraron valores mucho mayores a este valor, habiendo por ejemplo un registro con duración de 180 minutos.
Se decide utiliza run cut-off de 6 minutos, eliminando cualquier registro que tenga una duración mayor.

Normalizacion de atributos descriptivos

Se realiza una normalización de algunos atributos descriptivos utilizando el método de BoxCox. Con esto, se logra normalizar la distribución de las variables y eliminar también outliers (ver imagen de ejemplo).

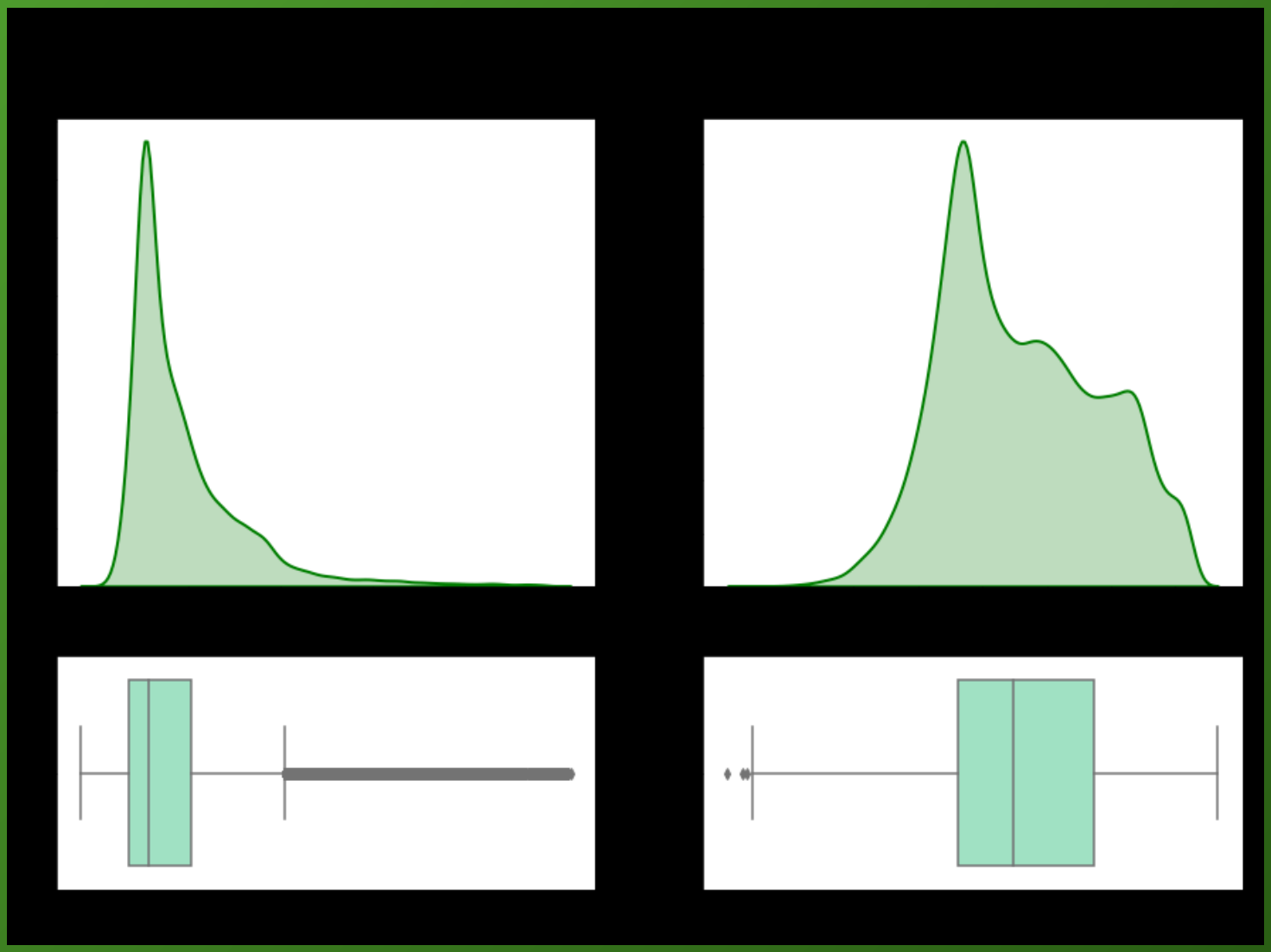


Imagen 1: Aplicación de Box Cox sobre variable Liveliness. Notar como se logra eliminar outliers

Entrenamiento de Algoritmos

Para este análisis, se aplicaron algoritmos tanto de clasificación como de regresión.

Algoritmos de Clasificación

En este caso se utiliza la variable creada ClasePopularidad. Al analizar la distribución de las clases, se observa que no están balanceadas, ya que las clases 4 y 5 (popularidades media/alta y alta) tienen muy bajos datos.

Clases	Popularidad	Cantidad de Registros
1 - Baja	0 - 20	17.771
2 - Media/Baja	21 - 40	19.894
3 - Media	41 - 60	14.161
4 - Media/Alta	61 - 80	3635
5 - Alta	81 - 100	82

Por esto, se utiliza como métrica para la evaluación de los algoritmos el F1 Score. Además, se realiza el entrenamiento no solo con el set original sino también utilizando la metodología SMOTE para poder balancear los datos.

Los algoritmos de clasificación utilizados fueron Arbol de Decision y Random Forest. A continuación se presentan los resultados obtendios:

Algoritmo	F1 Score	Parámetro Optimización
Árbol de Decisión	36.8%	Poda = 7
Random Forest	38.6%	N_Estimators = 130
Dummy Classifier	18.4%	Popularidad = 2

Algoritmo	F1 Score
Árbol de Decisión	53.1%
Random Forest	36%

Algoritmos de Regresión

En el caso del problema de regresión, se utiliza la variable target original Popularity. Se utilizaron tres algoritmos, que fueron:

- **Regresion Lineal: Se optimizo el grado del polinomio del algoritmo**
- **Random Forest Regresor**

LASSO:

Para el análisis de performance de los algoritmos, se utilizo la métrica de R2. A continuación se muestran los resultados obtenidos:

Algoritmo	R_2	Parametro Optimizacion
Regresion Lineal/Polinomica	0.081	Grado polinomio = 3
LASSO	0.005	-
Random Forest Regressor	0.097	N_Estimators = 140

Neural Nets:

Si bien en el curso no vimos este algoritmo, con las herramientas aprendidas pudimos enseñarnos la teoría detrás de las redes neurales para luego aplicarla mediante las funciones que ofrece el paquete scikit-learn.

Corrimos cuatro redes neurales. Dos de ellas para resolver un problema de clasificación y dos de ellas para resolver un problema de regresión. Asimismo, para cada problema (tanto de regresión como de clasificación) entrenamos las redes con dos variantes de datos.

La primera variante entrenó las redes neurales con los mismos datos que se usaron para los algoritmos anteriormente mencionados.

La segunda variante, gracias a una sugerencia de nuestro tutor utilizó un conjunto de datos reducido a solamente 4 dimensiones mediante PCA.

La implementación de las redes fue sorprendentemente fácil una vez que entendimos la matemática detrás de las mismas. A continuación, los resultados para regresión y clasificación:

Problemas de clasificación:

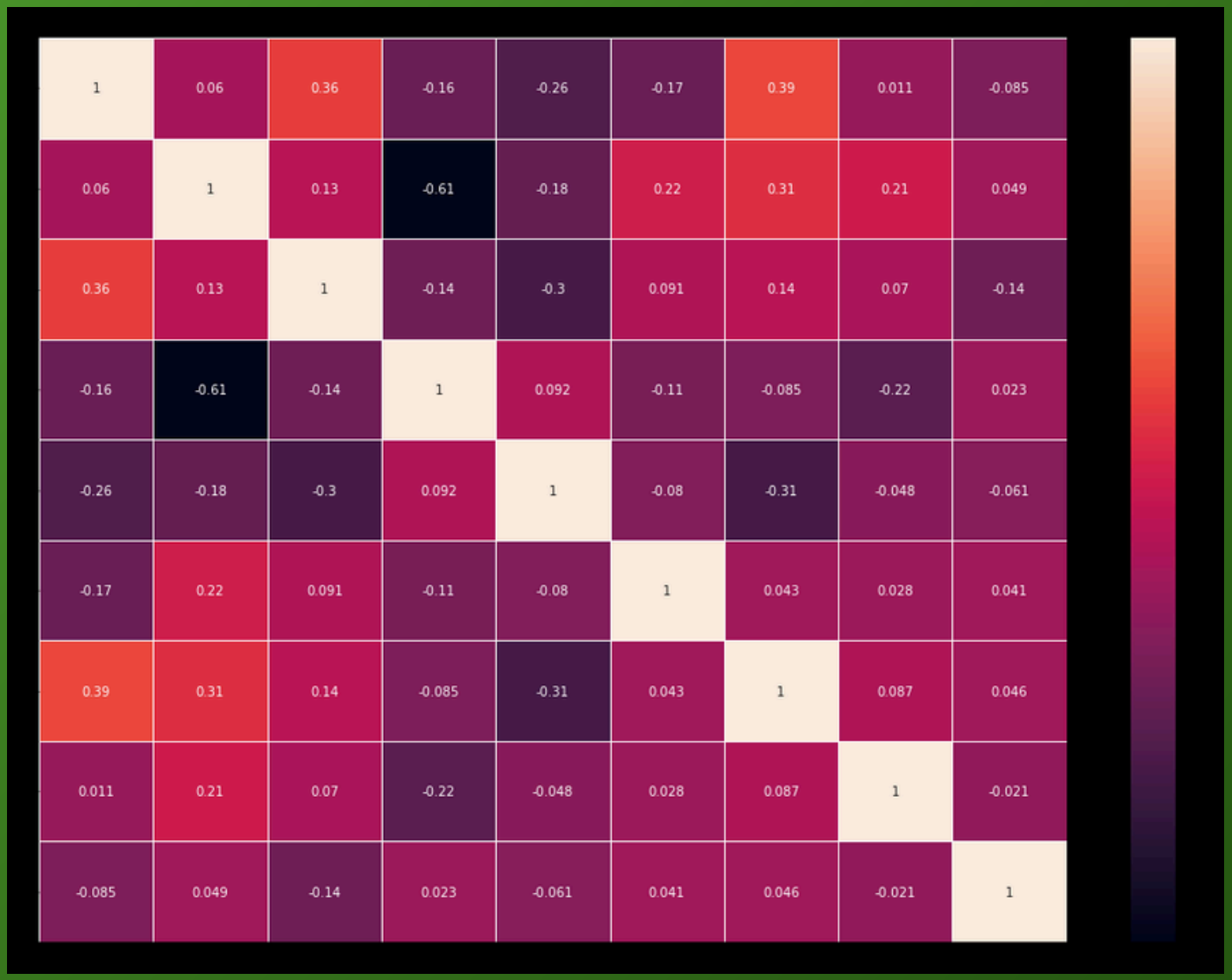
Red Neural	F1 Score	Parámetros elegidos
Red 1 con datos completos	38.35%	Hidden layers: (16,16) Activation: 'relu'
Red 2 con PCA	33.37%	Hidden layers: (10,10,10,10,10) Activation: 'relu'

Problemas de regresión:

Red Neural	R^2	Parámetros elegidos
Red 1 con datos completos	0.0804	Hidden layers : (16,16) Activation: 'relu'
Red 2 con PCA	0.0401	Hidden layers: (10,10,10,10,10) Activation: 'relu'

Conclusiones

Observando los resultados, podemos ver que estos son mejores para los problemas de clasificación que en los de regresión, pero aún así distan de tener un fuerte poder predictivo. Nuestra principal hipótesis acerca de este fenómeno tiene que ver con la muy baja correlación de las variables explicativas con la variable de salida. Desde muchas perspectivas podría argumentarse que se trata de variables esencialmente independientes o no correlacionadas, por lo que luego es difícil para cualquier tipo de modelo (incluso para las redes neurales que son bastante generales y no lineales) generar predicciones precisas. Abajo puede verse un heatmap ilustrando la situación.



Investigaciones futuras podrían intentar recopilar otro tipo de datos de las canciones para determinar cuáles son efectivamente los verdaderos predictores del éxito de un tema musical.