Descripción breve

- El presente informe aborda la exploración de los datos para su entendimiento dentro del contexto organizacional, la preparación de los datos para poder utilizarlos como entrada para modelos predictivos, el análisis preliminar de selección de modelos relevantes para responder a la pregunta, el desarrollo y calibración de modelos y la visualización de resultados .

MINI PROYECTO

Equipo 12 – Deep Learning

-Christian Beraún Chamorro

- D’sharlie Sánchez Rozo

- William Alexander Morales Valera

INTEGRANTES

# **INTRODUCCIÓN**

El presente miniproyecto se basa en las características musicales de una canción y si estas influyen o son determinantes respecto del año de publicación o lanzamiento. En resumen, se busca responder a la pregunta ¿Existe una relación entre las características musicales de una canción y el año en que fue publicada/lanzada? Una respuesta positiva a esta pregunta revelaría una profunda comprensión de la naturaleza de una composición musical y, lo que es más importante, esta comprensión se demostraría matemáticamente.

En este proyecto, se utilizan datos reales para predecir el año en que fue lanzada una canción a partir de sus características del timbre en la grabación. En total, son 90 atributos predictores: los 12 primeros corresponden al timbre promedio y los 78 siguientes a la covarianza. Originalmente, este tipo de datos fue recolectado en un proyecto llamado Million Song Dataset de la Universidad de Columbia (http://millionsongdataset.com/). Los datos para trabajar tienen los mismos predictores pero con canciones no consideradas en la base de datos original.

En general, se deben reportar los resultados del modelamiento predictivo siguiendo los pasos que se muestran a continuación:

* Exploración de los datos para su entendimiento dentro del contexto organizacional.
* Preparación de los datos para poder utilizarlos como entrada para modelos predictivos.
* Análisis preliminar de selección de modelos relevantes para responder a la pregunta.
* Desarrollo y calibración de modelos.
* Visualización de resultados.

**DATASETS PROPORCIONADOS**

* sampleSubmission.csv
* testReg.csv
* trainReg.csv

Disponibles en: https://www.kaggle.com/competitions/music-year-prediction/data

# **PARTE I: EXPLORACIÓN DE LOS DATOS Y CONTEXTO ORGANIZACIONAL**

Para empezar, necesitamos mapear la descripción, cantidad de variables y calidad de los datos de ambos datasets, los cuales se realizó a continuación:

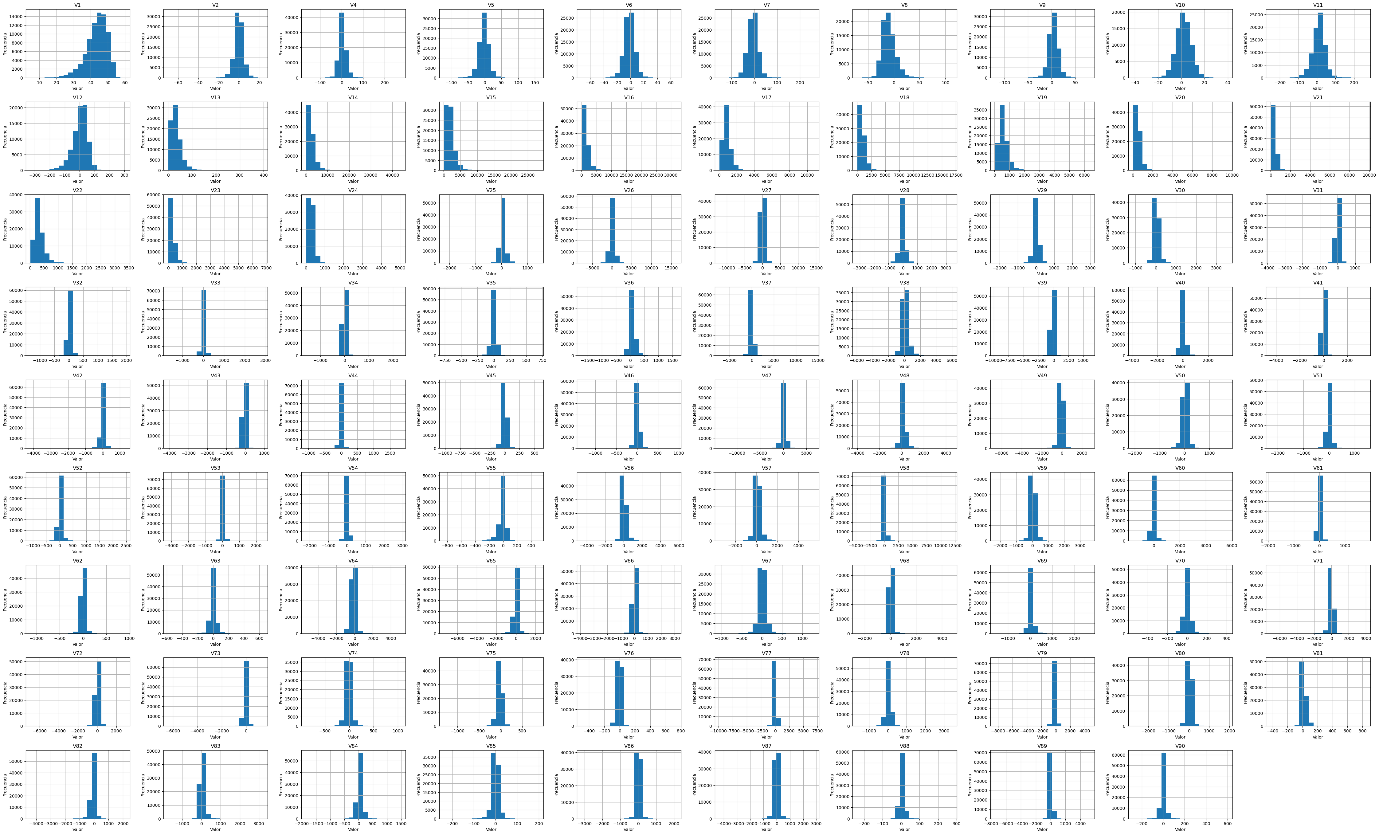
Train Data: 77779 rows × 92 columns

 ***Tabla 1. Análisis Descriptivo de los datos***

AÑADIR MÁS ANÁLISIS SOBRE EL CUADRO ANTERIOR

Siendo que son 91 variables, se revisó en el notebook de una forma más amplia los histogramas, a efectos de observar el comportamiento y distribución de las variables.

**Gráfico n.° 1 – Histogramas**



Luego se realizó un análisis de la distribución de la variable a predecir:

**Gráfico n.° 2 – Distribución de la variable a predecir**

Gráfico, Histograma

Descripción generada automáticamente

Como podemos observar, existe una distribución mayor para los años más cercanos a la fecha; ello podría deberse a que los datos se encuentran disponibles en diversas plataformas streaming de música siendo que estas surgieron en los últimos 20 años, tal como lo precisa en la página web de los datasets:

* SecondHandSongs dataset -> cover songs
* musiXmatch dataset -> lyrics
* Last.fm dataset -> song-level tags and similarity
* Taste Profile subset -> user data
* thisismyjam-to-MSD mapping -> more user data
* tagtraum genre annotations -> genre labels
* Top MAGD dataset -> more genre labels

Ahora bien, tengamos en cuenta este dato importante "los 12 primeros (V1 A V12) corresponden al timbre promedio y los 78 siguientes a la covarianza (V13 a V90)". Analizaremos la correlación que existe entre las variables predictoras con la variable de interés:

**Gráfico n.° 3 – Matriz de Correlación**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

AÑADIR MÁS ANÁLISIS DEL GRÁFICO ANTERIOR

# **PARTE II: PREPARACIÓN DE LOS DATOS PARA MODELOS PREDICTIVOS**

Como pudimos observar, las variables predictoras van de V1 a V90, pero se debe eliminar V3 y ID, siendo que no agregan valor al modelo, por ser valores vacíos o el índice de los registros. Ahora procedemos a analizar la dispersión de los datos

**Gráfico n.° 4 – Determinación de outliers**

Gráfico

Descripción generada automáticamente

Ahora bien, si observamos las variables V14 en adelante, podemos ver que hay mucha variabilidad. Mientras que de V1 a V12, no hay mucha.

**Gráfico n.° 5 – Boxplot de las variables V1 a V12**

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

Considerando todo lo anterior, se vio por conveniente crear variables que capturen los datos de V1 a V12 y de V13 a V90, usando métricas como suma, promedio, mediana, producto, entre otras. Conforme se detalla a continuación:



AÑADIR MÁS ANÁLISIS DEL CÓDIGO ANTERIOR

**Gráfico n.° 6 – Correlación con “Y” de las nuevas variables**

Gráfico, Gráfico en cascada

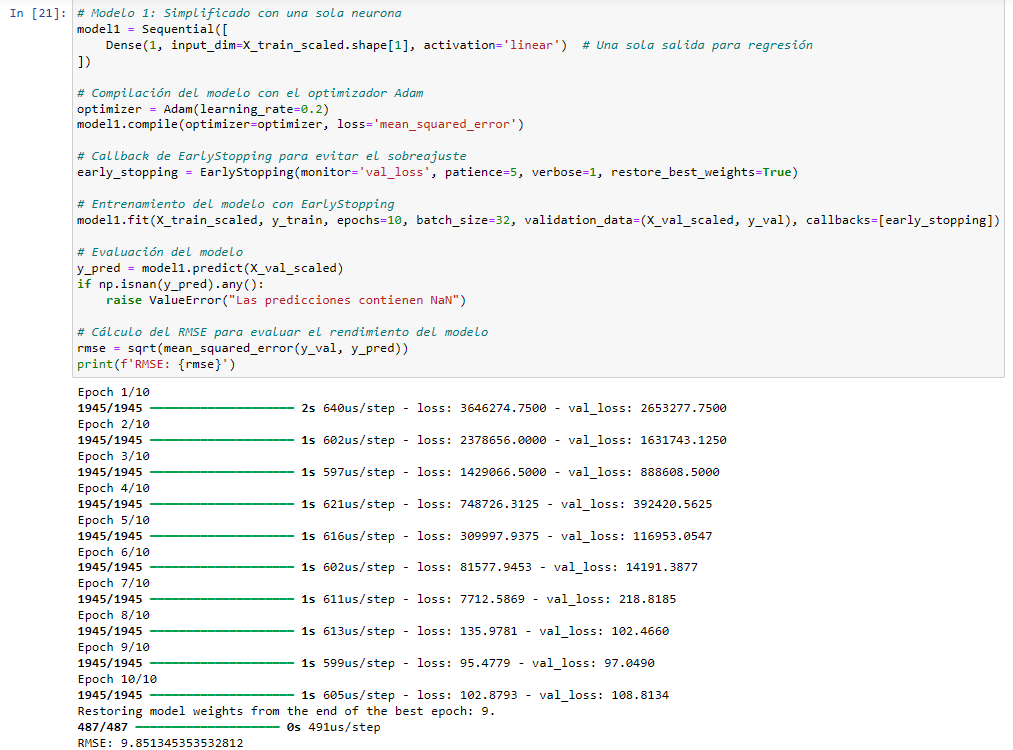
Descripción generada automáticamente

Aquí podemos observar que fue una buena idea agregar estas características, especialmente aquellas que tienen una correlación más alta del 0.02, toda vez que permiten capturar los datos de los dos grupos indicados.

AÑADIR MÁS ANÁLISIS DEL GRÁFICO ANTERIOR

# **PARTE III: ANÁLISIS PRELIMINAR DE SELECCIÓN DE MODELOS RELEVANTES**

### \*Modelo 1: Una sola neurona \*



### \*Modelo 2: Múltiples Capas \*

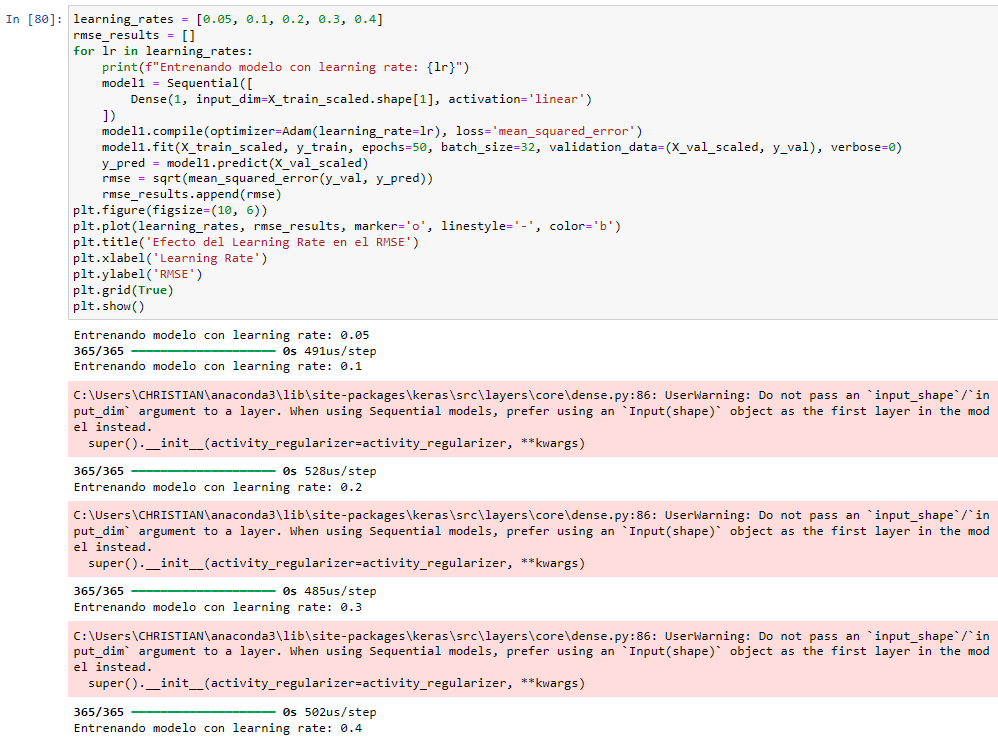
Texto

Descripción generada automáticamente

Cabe precisar que existirá un modelo 3, pero será resultante de la calibración del modelo 2 lo cual se observará en las siguientes páginas.

# **PARTE IV: DESARROLLO Y CALIBRACIÓN DE MODELOS**

Se realizó la calibración de ambos modelos, por ejemplo a nivel de learning rate:



**Gráfico n.° 7 – Calibración Learning Rate Modelo 1**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

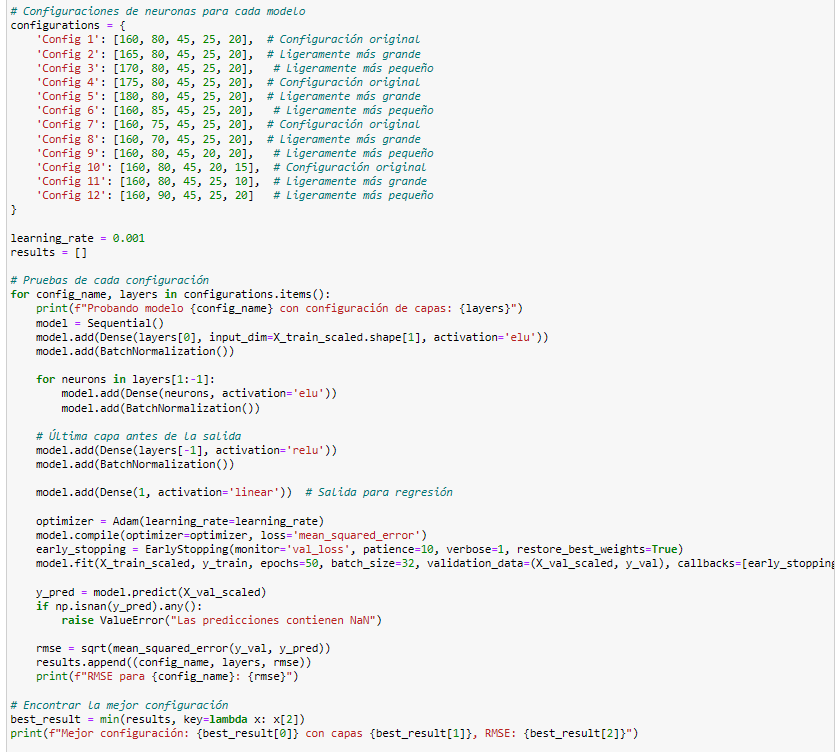
Y para el modelo 2 con capas ocultas (múltiples capas):

**Gráfico n.° 78– Calibración Learning Rate Modelo 2**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Como podemos observar, en ambos casos el óptimo del Leargnin Rate es 0.001; ahora procedemos calibrar aún más el modelo 2:



El código anterior fue diseñado para explorar la configuración óptima de una red neuronal multicapa. Una vez establecida la estructura básica, se procedió a calibrar ciertos parámetros y a incorporar diversas técnicas para mejorar el rendimiento y la generalización del modelo. Entre estas técnicas se incluyeron el uso de dropout y normalización por lotes (batch normalization) para regularizar el modelo, la implementación de paradas tempranas (early stopping) para evitar el sobreajuste, la optimización mediante el algoritmo Adam, y la selección cuidadosa de las funciones de activación. Estas estrategias fueron aplicadas con el objetivo de afinar el modelo y alcanzar los mejores resultados posibles en términos de precisión y eficiencia durante el entrenamiento.

En el modelo final calibrado (Modelo 3), tenemos lo siguiente:

Texto

Descripción generada automáticamente

**Sobre el modelo:**

* **Dense:** Representa una capa densamente conectada (o completamente conectada). Cada neurona en una capa densa recibe entrada de todas las neuronas de la capa anterior, lo cual es una característica de las redes neuronales tradicionales.
* **La primera capa densa con 360 neuronas**, donde input\_dim especifica el número de características de entrada del modelo. La función de activación 'relu' (Rectified Linear Unit) es comúnmente utilizada por su eficiencia y efectividad en redes neuronales profundas.
* **Dropout(0.3):** Una técnica de regularización donde aleatoriamente se "apagan" un porcentaje de las neuronas durante el entrenamiento para prevenir el sobreajuste. 0.3 significa que el 30% de las neuronas en la capa anterior se desactivarán aleatoriamente en cada paso durante el entrenamiento.
* **BatchNormalization():** Normaliza las activaciones de la capa anterior al restar la media del batch y dividir por la desviación estándar, lo que ayuda a mejorar la velocidad, rendimiento y estabilidad del entrenamiento.

Considerando que tiene normalización, aplicación técnicas y ajustes, se considero a este modelo 2 calibrado, como el modelo 3 “Múltiples capas Adam y calibración de hiperparámetros”

Siendo el mejor modelo obtenido, se generaron los predicts

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Hito:

* RMSE EN LAB LOCAL - 8.41185
* RMSE EN KAGGLE - 8.64070
* RANKING EN EL PUBLIC LEADERBOARD 5 DE 21 EQUIPOS.
* DIFERENCIA CON EL RANKING 1: 8.64070 - 8.4901 = 0.1505

Fuente: Public Leaderboard de Kaggle: <https://www.kaggle.com/competitions/music-year-prediction/leaderboard>, con información al 29.04.2024 a las 4:15 am.

# **PARTE V: VISUALIZACIÓN DE RESULTADOS**

A continuación se presentan los resultados obtenidos por cada tipo de modelo empleado:

**Gráfico n.° X– Comparación de modelos**

Gráfico, Gráfico de barras

Descripción generada automáticamente

Como podemos observar, inicialmente daba la impresión de que el modelo de una sola neurona sería mejor que de múltiples capas, pero a lo largo de las calibraciones realizadas, empleando batchnormalization, técnica de dropout y búsqueda intensiva de parámetros óptimos que incluye el learning rate, se obtuvieron los siguientes resultados:

* 1. Modelo 1: Una sola Neurona / RMSE OBTENIDO: 9.85
  2. Modelo 2: Múltiples capas / RMSE OBTENIDO: 13.86
  3. Modelo 3: Múltiples capas Adam y calibración de hiperparámetros / RMSE OBTENIDO: 8.411

En ese sentido, el modelo 3 al tener el mejor RMSE, fue utilizado para generar las predicciones y así obtener el puntaje de 8.64070, con lo cual nuestro equipo se colocó en el ranking 5 de la competencia.

***Tabla 2. Modelos y los resultados obtenidos***

|  |  |  |
| --- | --- | --- |
| Detalle del Modelo | RMSE JupyterNotebook | Puntuación Kaggle |
| Modelo 01: Red neuronal – Una sola neurona | 9.85 | 10.11 |
| Modelo 02: Red neuronal – Multicapa | 13.86 | 15.86 |
| Modelo 03: Red neuronal – Multicapa Adam calibrado | 8.411 | 8.64 |

Recomendaríamos utilizar el modelo de redes neuronales Adam, calibrando los parámetros de número de capas, tasa de aprendizaje, neuronas por capa, tipo de función de activación, implementando estrategias de batchnormalization, dropout y earlystopping, a efectos de mejorar aún más el modelo.

Ahora bien, respondiendo a la pregunta de interés... sí, es posible siendo que las variables indican que existiría una relación entre las características musicales de una canción y el año en que fue publicada/lanzada.

**Anexos.**

Anexo 1. Resultado en Competencia (pdf )

Anexo 2. HTML Notebook

Anexo 3. Notebook en Jupyter