



Ranking de Música de Spotify

Predicción de popularidad de canciones utilizando modelos de Deep Learning

Integrantes:

1. Isidora Jara Muñoz
2. Néstor Manríquez Figueroa
3. Daniel Reyes Arias

Profesor:

Francisco Plaza Vega



0:00

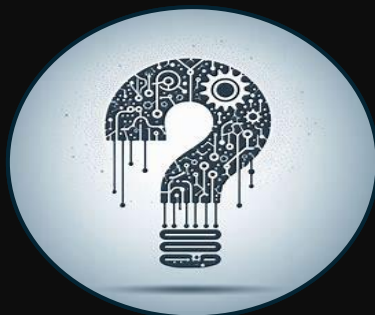
15:40



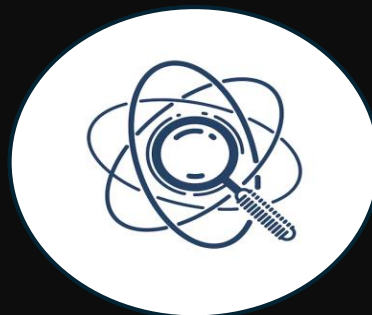
Índice



Objetivos



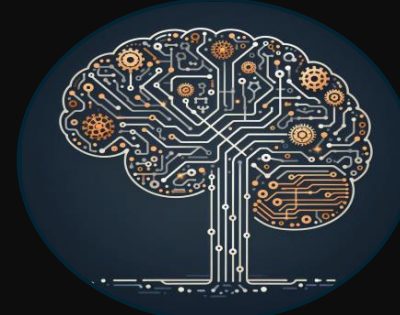
Implementación



Entrenamiento



Validación



Conclusiones



Objetivos

General:

Desarrollar un modelo de Deep Learning para predecir qué canciones serán las más populares en la plataforma de streaming de Spotify.

Específicos:

1. Recopilar y preparar los datos.
2. Realizar un análisis exploratorio de los datos.
3. Desarrollar y entrenar un modelo de Deep Learning.
4. Evaluar el modelo.



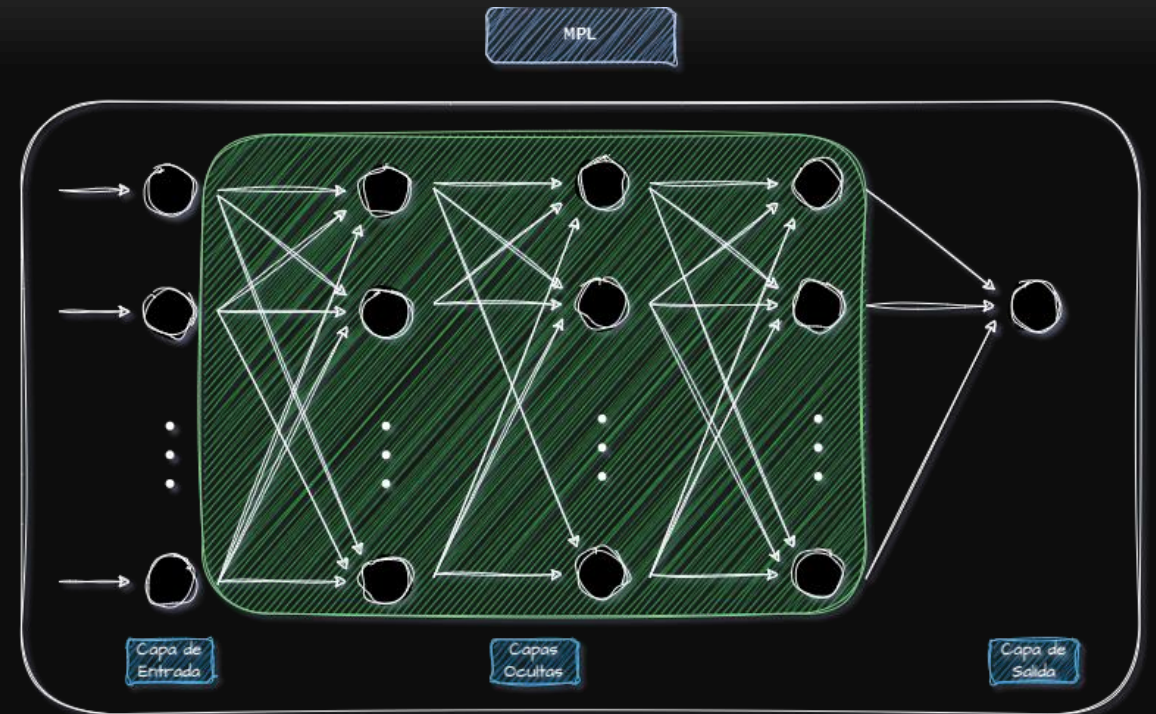
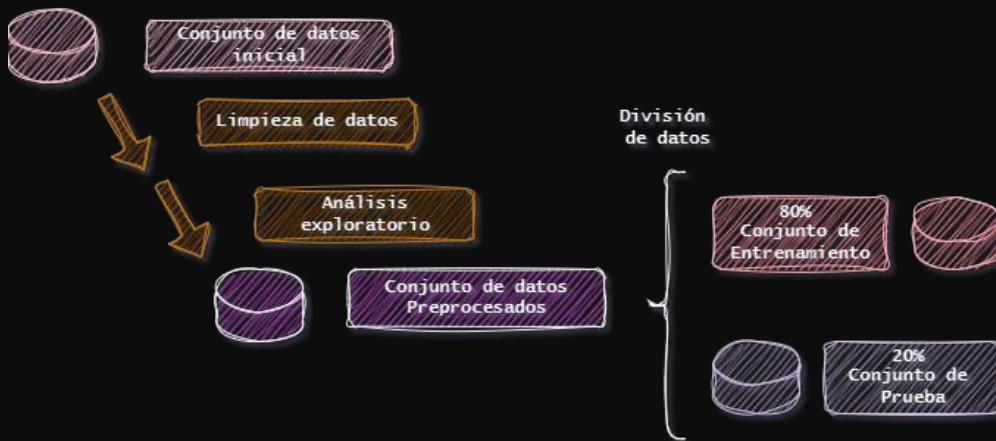
Descripción de datos

Variable	Descripción
País	Indica si la canción es global o de uno de los 34 países donde Spotify opera.
Popularidad	Puntuación basada en días en el Top 200 y su posición diaria.
Top_Max	Posición más alta alcanzada en el top en 1401 días de datos.
Título	Nombre de la canción.
Artista	Nombre del artista. (Artistas Agrupados)
Formato	Publicada como sencillo o parte de un álbum.
Explícito	Indica si tiene advertencia de contenido explícito.
Bailable	Adecuación para bailar (0.0 a 1.0).
Energía	Intensidad y actividad de la canción (0.0 a 1.0).
Tono	Tono general de la pista (0 = Do, 1 = Do#, etc.).
Volumen	Sonoridad en decibelios (dB).
Modo	Modalidad (1 = mayor, 0 = menor).
Hablado	Presencia de palabras habladas (0.0 a 1.0).
Acústico	Confianza en si la pista es acústica (0.0 a 1.0).
Instrumental	Probabilidad de que la pista sea instrumental (0.0 a 1.0).
Valencia	Positividad musical de la pista (0.0 a 1.0).
BPM	Tempo en pulsos por minuto.
Duración_ms	Duración de la pista en milisegundos.
Género	Género predominante del artista.
Días_Lanzamiento	Días desde el lanzamiento hasta la fecha de referencia.
Continente	Nombre del continente al cual pertenece el país.



Implementación

General



Plataforma	GPU	Procesador	RAM
Rstudio	No	Ryzen 5	12 GB

Entrenamiento

Modelo 1.12

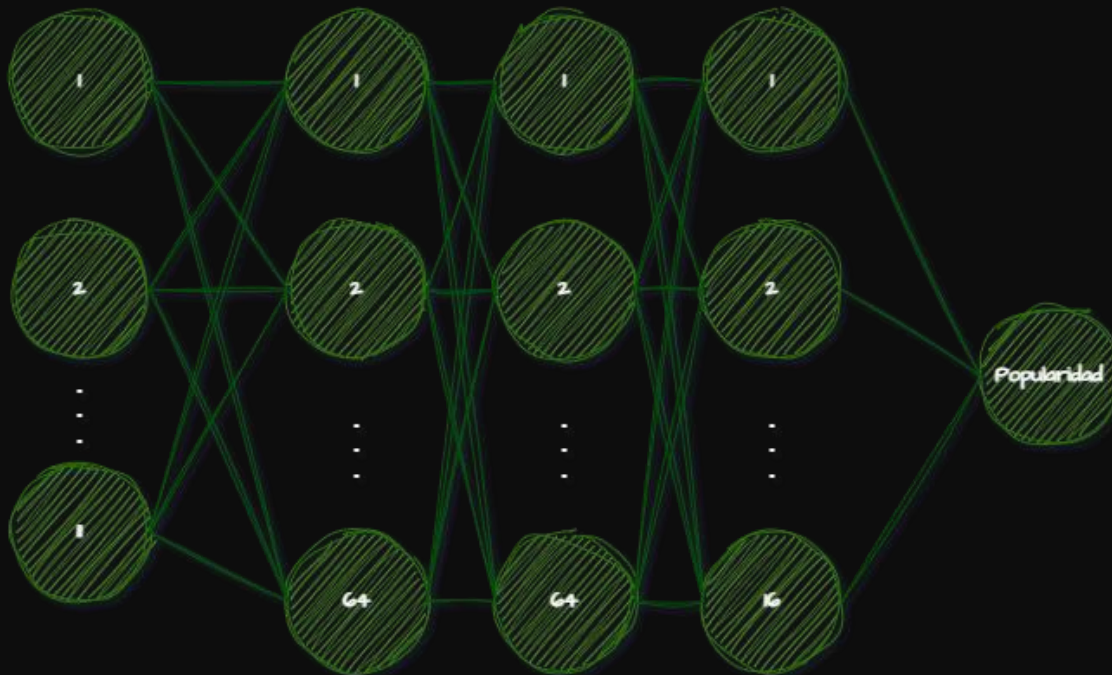
Capa de Entrada

Capa Oculta 1

Capa Oculta 2

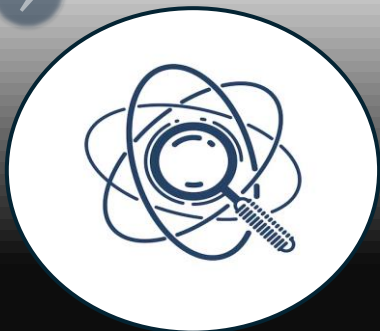
Capa Oculta 3

Capa de Salida



Top_Max, Artista, País, Explícito, Formato,
BPM, Valencia, Género, Popularidad

```
build_and_compile_model12 <- function(norm) {  
  model <- keras_model_sequential() %>%  
    norm() %>%  
    layer_dense(64, activation = 'relu') %>% #Capa oculta 1  
    layer_dense(64, activation = 'relu') %>% #Capa oculta 2  
    layer_dense(16, activation = 'relu') %>% #Capa oculta 3  
    layer_dense(1) #Una capa Dense lineal de salida única.  
  
  model %>% compile( #compile configura el procedimiento de entrenamiento  
    loss = 'mean_absolute_error', #función de pérdida  
    optimizer = optimizer_adam(0.001), #optimizador  
    metrics = c('mean_squared_error') #agrego métricas  
  )  
  
  model  
}  
  
tiempo_entrenamiento12 <- system.time({  
  history12 <- modelo12 %>% fit(  
    as.matrix(entrenamiento_x3),  
    as.matrix(entrenamiento_y3),  
    validation_split = 0.2,  
    epochs = 100  
  )  
})
```



Entrenamiento

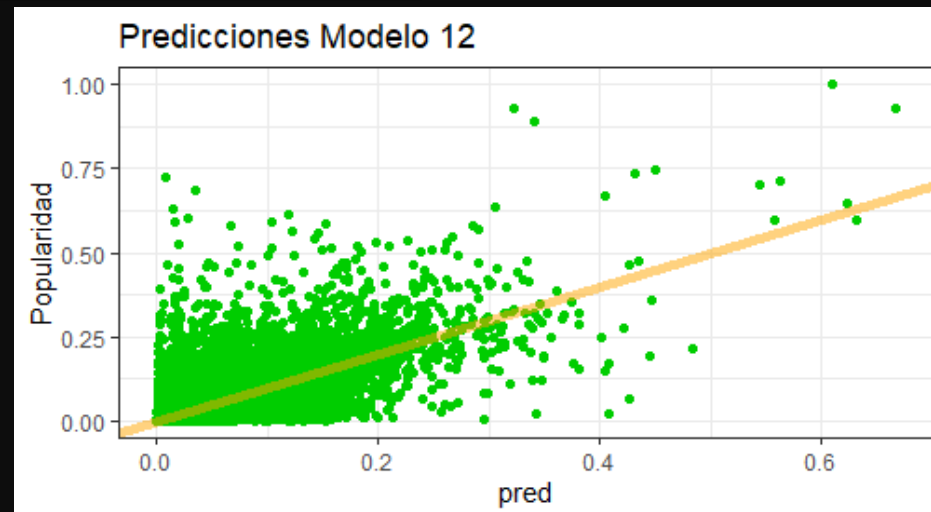
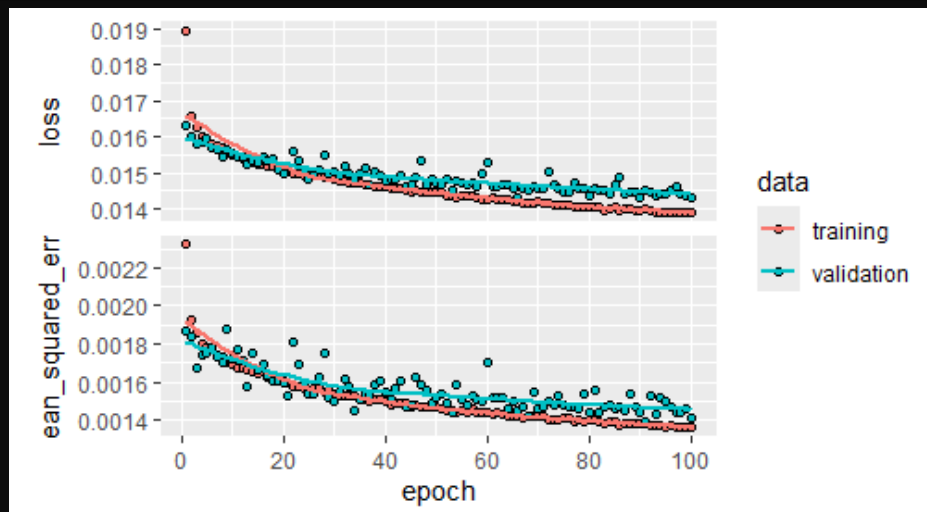
Modelo 1.12

Layer (type)	Output Shape	Param #	Trainable
normalization (Normalization)	(None, 11)	23	Y
dense_3 (Dense)	(None, 64)	768	Y
dense_2 (Dense)	(None, 64)	4160	Y
dense_1 (Dense)	(None, 16)	1040	Y
dense (Dense)	(None, 1)	17	Y

Total params: 6008 (23.47 KB)
Trainable params: 5985 (23.38 KB)
Non-trainable params: 23 (96.00 Byte)

Modelo12
loss 0.014364036
mean_squared_error 0.001423892

Modelo12.elapsed
17.83383



Entrenamiento

Modelo 2.13

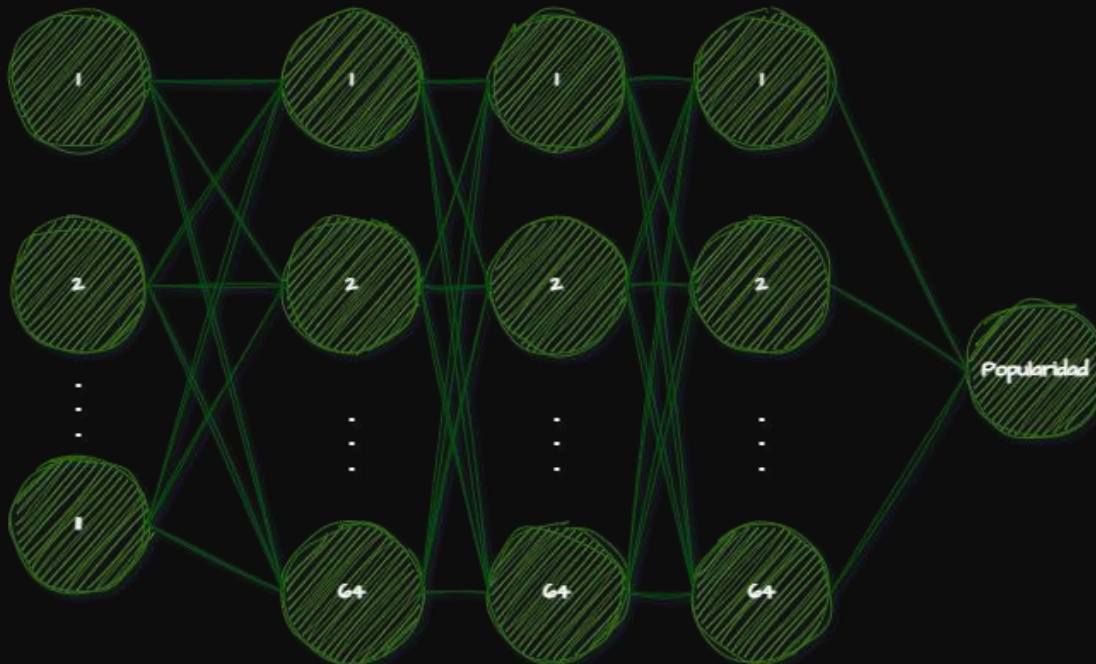
Capa de Entrada

Capa Oculta 1

Capa Oculta 2

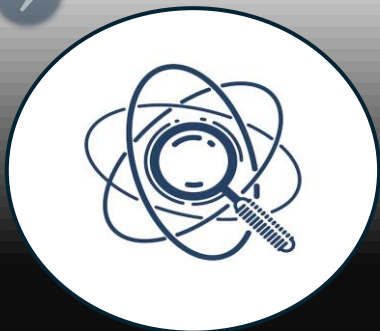
Capa Oculta 3

Capa de Salida



Top_Max,Artista,País,Explícito,Formato,
BPM,VaIencia,Género,Popularidad

```
build_and_compile_model13 <- function(norm) {  
  model <- keras_model_sequential() %>%  
    norm() %>%  
    layer_dense(64, activation = 'relu') %>% #Capa oculta 1  
    layer_dense(64, activation = 'relu') %>% #Capa oculta 2  
    layer_dense(64, activation = 'relu') %>% #Capa oculta 3  
    layer_dense(1) #Una capa Dense lineal de salida única.  
  
  model %>% compile( #compile configura el procedimiento de entrenamiento  
    loss = 'mean_absolute_error', #función de pérdida  
    optimizer = optimizer_adam(0.001), #optimizador  
    metrics = c('mean_squared_error') #agrego métricas  
  )  
  
  model  
}  
  
tiempo_entrenamiento13 <- system.time({  
  history13 <- modelo13 %>% fit(  
    as.matrix(entrenamiento_x3),  
    as.matrix(entrenamiento_y3),  
    validation_split = 0.2,  
    epochs = 300  
  )  
})
```

Entrenamiento

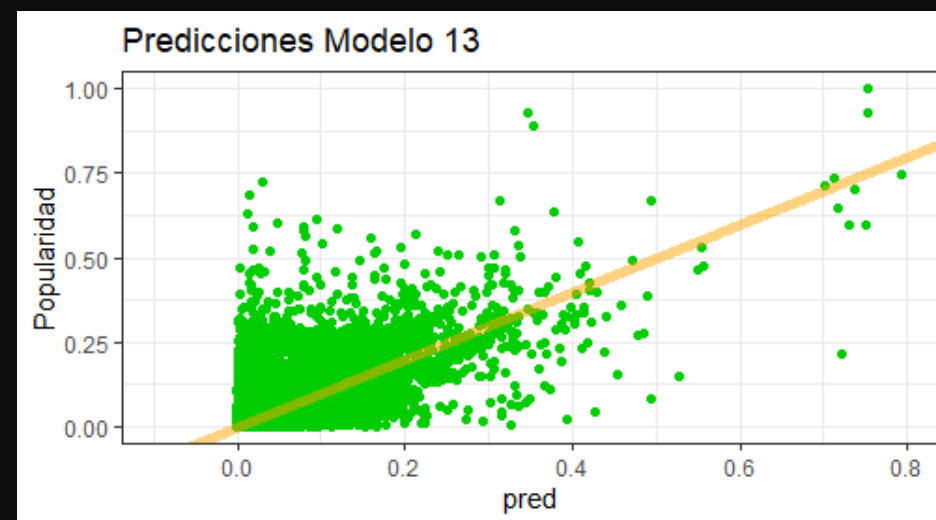
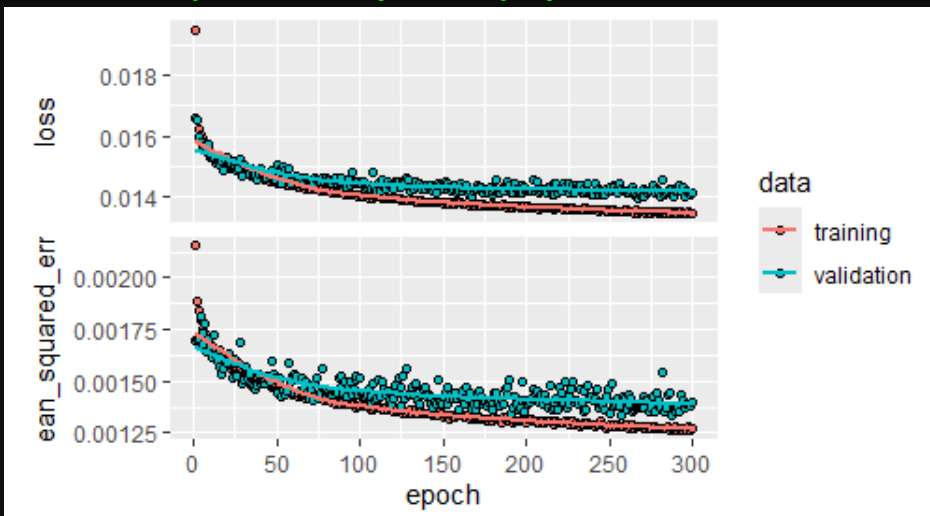
Modelo 2.13

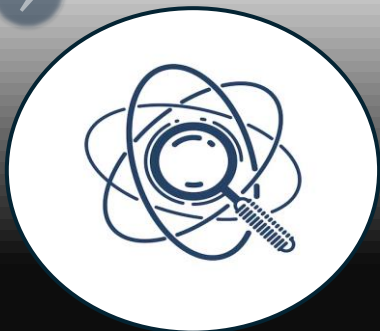
Layer (type)	Output Shape	Param #	Trainable
normalization (Normalization)	(None, 11)	23	Y
dense_7 (Dense)	(None, 64)	768	Y
dense_6 (Dense)	(None, 64)	4160	Y
dense_5 (Dense)	(None, 64)	4160	Y
dense_4 (Dense)	(None, 1)	65	Y

Modelo13
loss 0.01424418
mean_squared_error 0.00143131

Modelo13.elapsed
60.11383

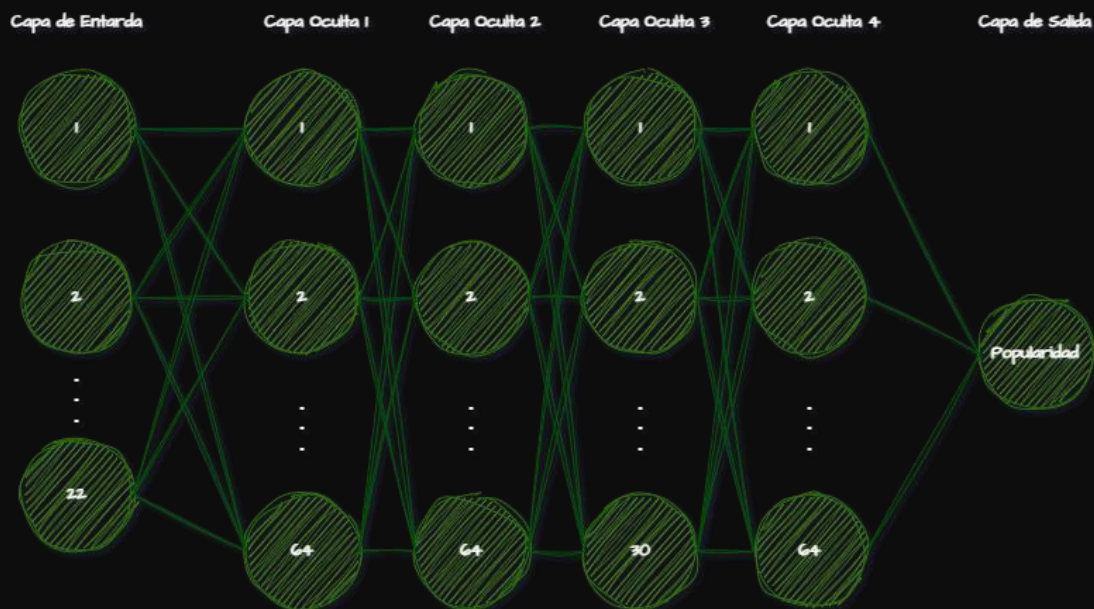
Total params: 9176 (35.85 KB)
Trainable params: 9153 (35.75 KB)
Non-trainable params: 23 (96.00 Byte)





Entrenamiento

Modelo 3.10



Top_Max, Bailable, Energía, Tono, Volumen, Modo, Hablado, Acústico, Instrumental, Valencia, BPM, Duración_ms, Formato, Días_Lanzamiento, Artista, Continente, País, Género, Explícito

```
build_and_compile_model10 <- function(norm) {  
  model <- keras_model_sequential() %>%  
    norm() %>%  
    layer_dense(64, activation = 'relu') %>% #Capa oculta 1  
    layer_dense(64, activation = 'relu') %>% #Capa oculta 2  
    layer_dense(30, activation = 'relu') %>% #Capa oculta 3  
    layer_dense(64, activation = 'relu') %>% #Capa oculta 4  
    layer_dense(1) #Una capa Dense lineal de salida única.  
  
  model %>% compile( #compile configura el procedimiento de entrenamiento  
    loss = 'mean_absolute_error', #función de pérdida  
    optimizer = optimizer_adam(0.001), #optimizador  
    metrics = c('mean_squared_error') #agrego métricas  
  )  
  
  model  
}  
  
tiempo_entrenamiento10 <- system.time({  
  history10 <- modelo10 %>% fit(  
    as.matrix(entrenamiento_x),  
    as.matrix(entrenamiento_y),  
    validation_split = 0.2,  
    epochs = 300  
  )  
})
```



Entrenamiento

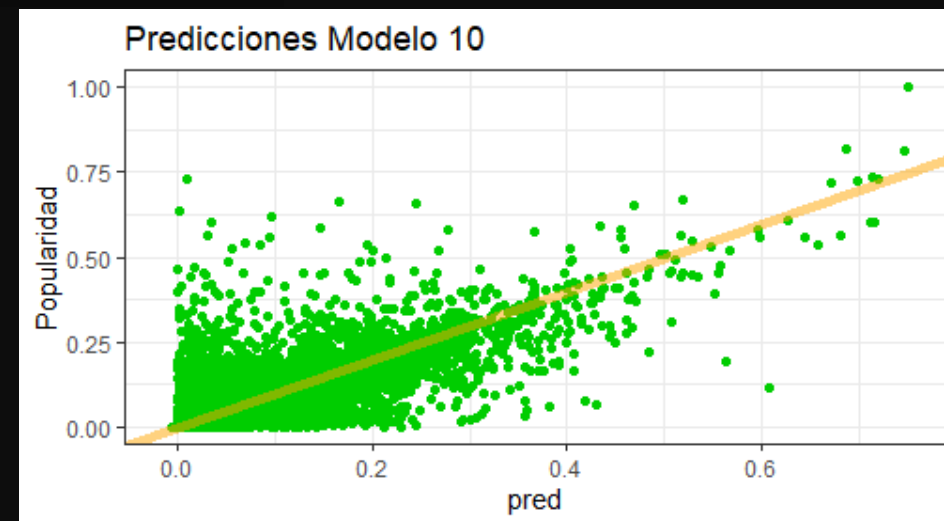
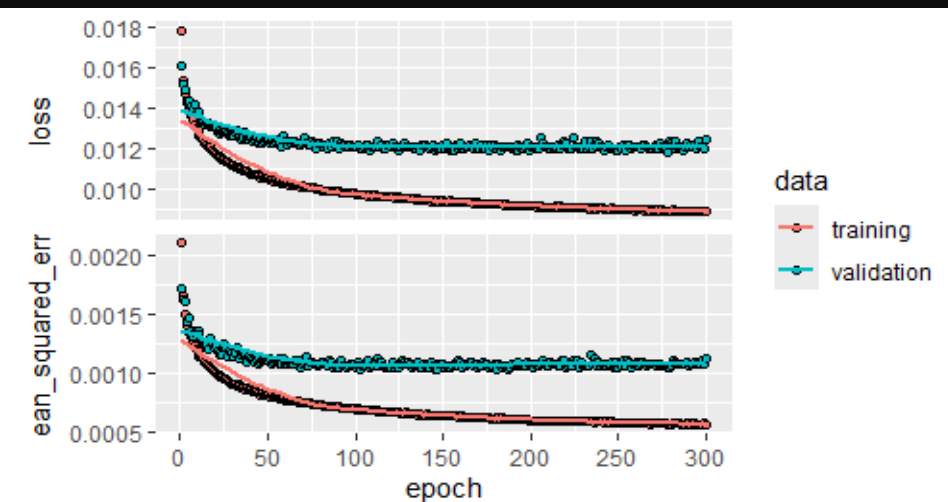
Modelo 3.10

Layer (type)	Output Shape	Param #	Trainable
normalization_1 (Normalization)	(None, 22)	45	Y
dense_17 (Dense)	(None, 64)	1472	Y
dense_16 (Dense)	(None, 64)	4160	Y
dense_15 (Dense)	(None, 30)	1950	Y
dense_14 (Dense)	(None, 64)	1984	Y
dense_13 (Dense)	(None, 1)	65	Y

Total params: 9676 (37.80 KB)
Trainable params: 9631 (37.62 KB)
Non-trainable params: 45 (184.00 Byte)

Modelo10
loss 0.012071264
mean_squared_error 0.001072746

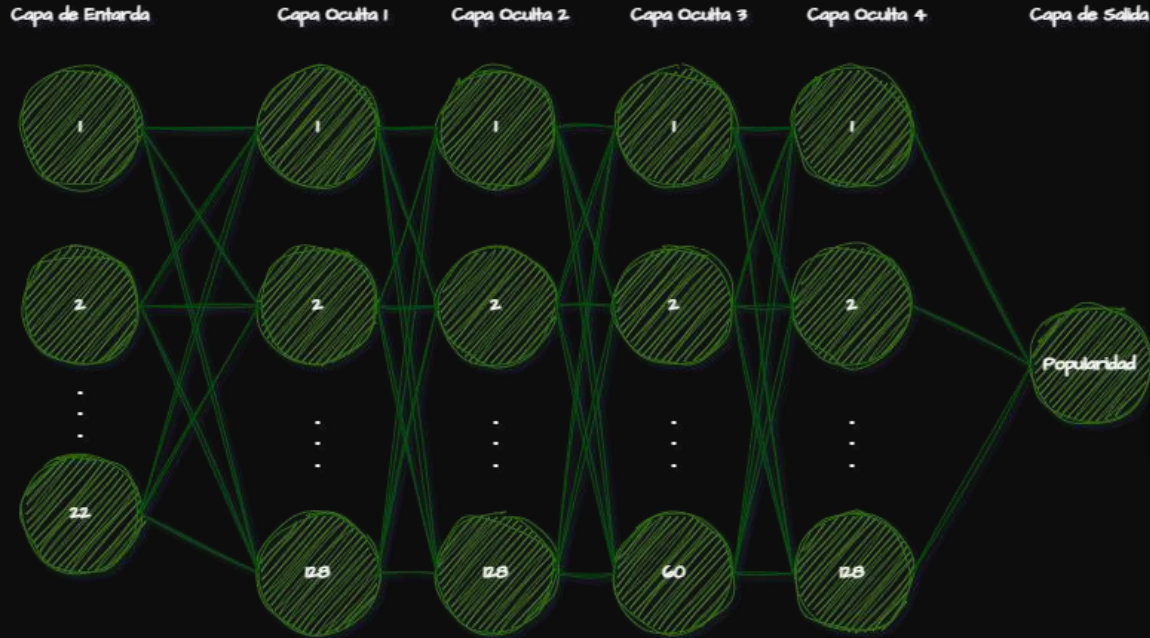
Modelo10.elapsed
64.63467



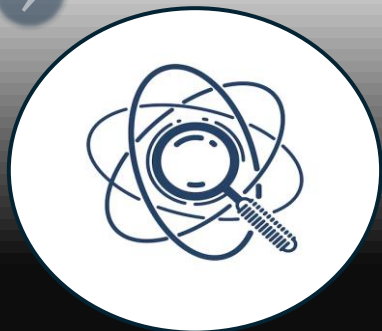
Entrenamiento

Modelo 4.11

```
build_and_compile_model11 <- function(norm) {  
  model <- keras_model_sequential() %>%  
    norm() %>%  
    layer_dense(128, activation = 'relu') %>% #Capa oculta 1  
    layer_dense(128, activation = 'relu') %>% #Capa oculta 2  
    layer_dense(60, activation = 'relu') %>% #Capa oculta 3  
    layer_dense(128, activation = 'relu') %>% #Capa oculta 4  
    layer_dense(1) #Una capa Dense lineal de salida única.  
  
  model %>% compile( #compile configura el procedimiento de entrenamiento  
    loss = 'mean_absolute_error', #función de pérdida  
    optimizer = optimizer_adam(0.001), #optimizador  
    metrics = c('mean_squared_error') #agrego métricas  
  )  
  
  model  
}  
  
tiempo_entrenamiento11 <- system.time({  
  history11 <- modelo11 %>% fit(  
    as.matrix(entrenamiento_x),  
    as.matrix(entrenamiento_y),  
    validation_split = 0.2,  
    epochs = 300  
  )  
})
```



Top_Max, Bailable, Energía, Tono, Volumen, Modo, Hablado, Acústico, Instrumental, Valencia, BPM, Duración_ms, Formato, Días_Lanzamiento, Artista, Continente, País, Género, Explícito



Entrenamiento

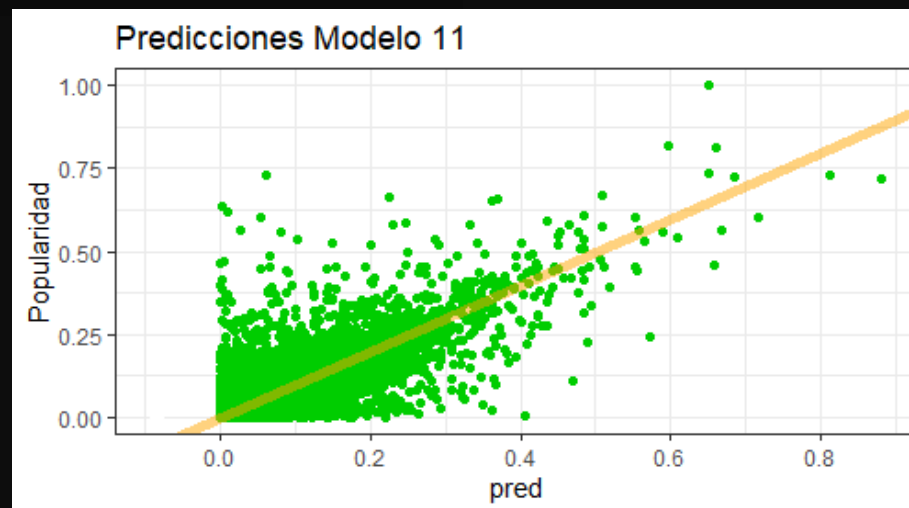
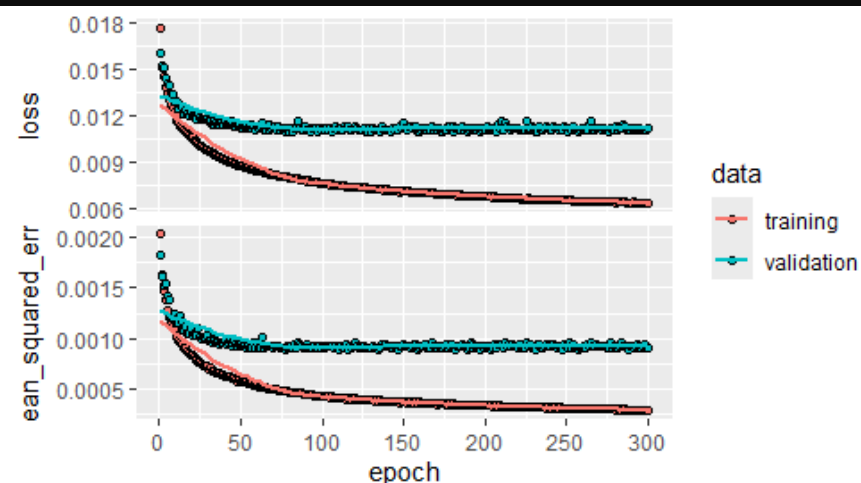
Modelo 4.11

Layer (type)	Output Shape	Param #	Trainable
normalization_1 (Normalization)	(None, 22)	45	Y
dense_22 (Dense)	(None, 128)	2944	Y
dense_21 (Dense)	(None, 128)	16512	Y
dense_20 (Dense)	(None, 60)	7740	Y
dense_19 (Dense)	(None, 128)	7808	Y
dense_18 (Dense)	(None, 1)	129	Y

Total params: 35178 (137.42 KB)
Trainable params: 35133 (137.24 KB)
Non-trainable params: 45 (184.00 Byte)

Modelo11
loss 0.0113630416
mean_squared_error 0.0009511932

Modelo11.elapsed
65.77617





Validación

Modelo	Error Absoluto Medio	Error Cuadrático Medio	Tiempo de Computo (min)	Épocas
1.12	0.01436	0.00142	17.83	100
2.13	0.01424	0.00143	60.11	300
3.10	0.01207	0.00107	64.63	300
4.11	0.01136	0.00095	65.77	300



Conclusiones

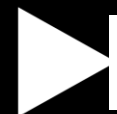
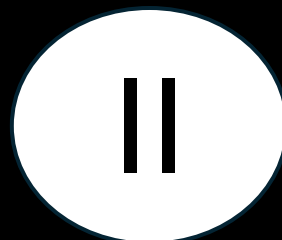
- Ampliación del Dataset: Incluir más datos, en lo posible en tiempo real, así como tendencias en redes sociales, letras de canciones (NLP), colaboraciones entre artistas, etc.
- Optimización y Regularización: Experimentar con diferentes técnicas de regularización, como dropout, y optimizadores avanzados como AdamW.
- De igual forma, se podría visualizar el modelo al utilizar más capas ocultas y/o épocas para ver si mejora la predicción de la popularidad.



14:01

15:40





15:40



15:40

Letras

El Fin.

Muchas gracias por su atención.