

PROYECTO

RECOMENDACIÓN DE CANCIONES UTILIZANDO UNA ANN

PROBLEMA

El objetivo del problema estuvo centrado en una predicción binaria de canciones, se quiere predecir si una canción con x características que puede ser de agrado para un agente Y. Esto tratará de imitar el modelo de recomendación de canciones nuevas que plataformas de música como Spotify, Apple music.

ORIGEN DATASET

Extraido de la API de Spotify

VARIABLES INDEPENDIENTES

Métricas que describen a la canción: duración, calidad de la canción, existencia de voces, score de instrumentos, timbre de la canción.

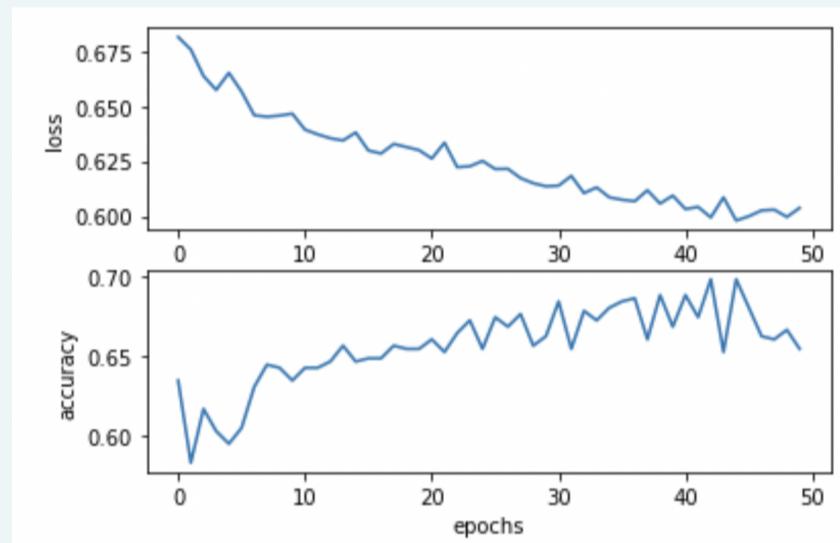
VARIABLE DEPENDIENTE

Target: Si la canción fue de agrado para el agente.

EXPERIMENTACIÓN

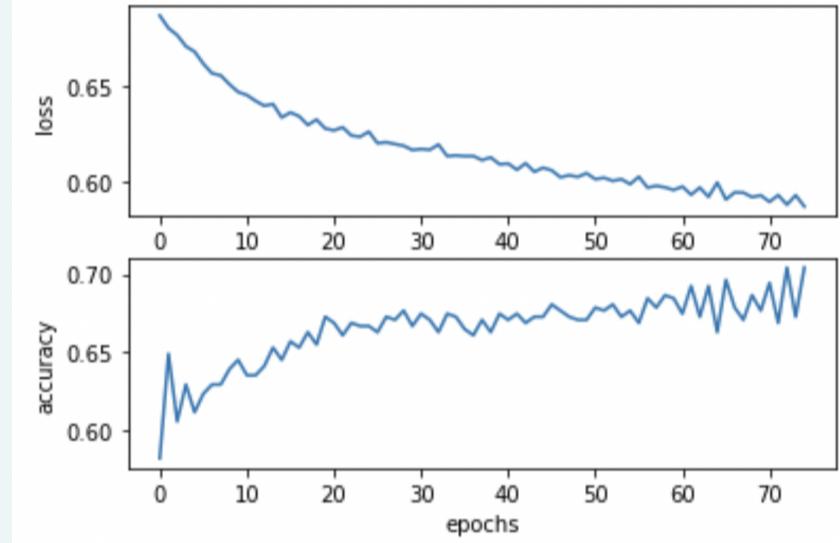
Para la etapa de experimentación se usaron 4 arquitecturas, a continuación se muestran los resultados del mejor experimento por arquitectura.

ARQUITECTURA 1



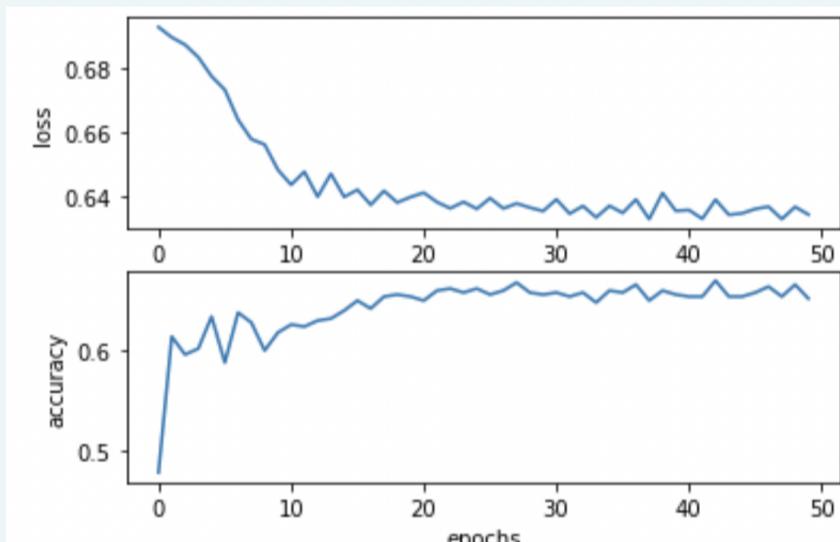
Error y exactitud en set de validación durante entrenamiento en arquitectura 1

ARQUITECTURA 2



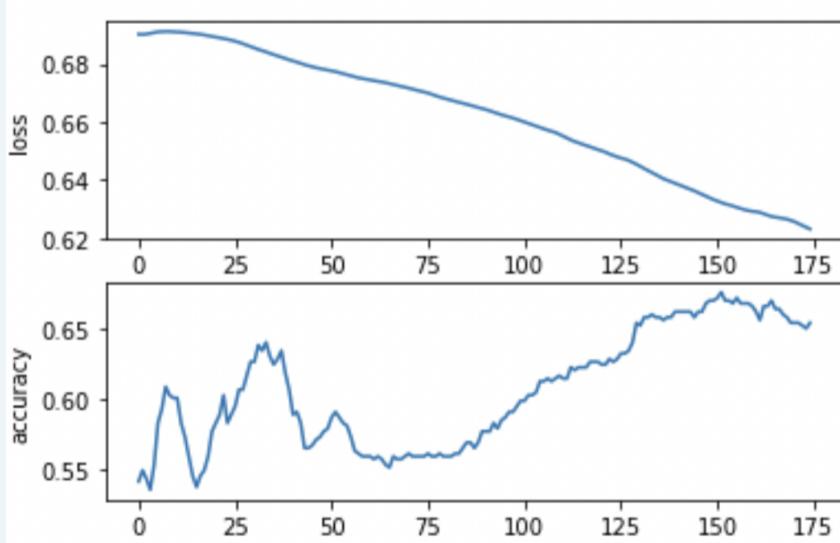
Error y exactitud en set de validación durante entrenamiento en arquitectura 2

ARQUITECTURA 3



Error y exactitud en set de validación durante entrenamiento en arquitectura 3

ARQUITECTURA 4



Error y exactitud en set de validación durante entrenamiento en arquitectura 4 (ganadora)

CONCLUSIONES

Con el último experimento se logró tener una exactitud del 80% sobre el set de validación. Bajo la arquitectura ganadora descrita a continuación:

- 3 capas ocultas, 1 capa con 128 neuronas, la segunda compuesta por 64 neuronas y 32 neuronas en la tercera, ambas activadas por la función relu .
- 2 capas de normalización del output continuas a las salidas de las capas densas
- 1 capa de dropout para castigar el sobre ajuste al desactivar el 10% de las salidas de la última capa oculta
- 1 capa densa de 1 neurona con activación sigmoid

Se logró disminuir el error a 0.14, teniendo una exactitud del 92% en el entrenamiento y de 80% en el set de validación.

Basados en el resultado podemos concluir que el modelo puede tomar canciones aleatorias y clasificarlas para ser mostradas y estas deberían de ser aceptadas o queridas por el agente el 80% de las veces.

STATISTICAL

DETECCIÓN DE ENFERMEDADES EN EL OJO CON CNN

PROBLEMA

La detección temprana de enfermedades en los ojos es crucial para conseguir una visión duradera a través de los años, es por esto que para esta parte se desarrolló una Red Neuronal Convolutinal la cual nos permitirá clasificar entre 5 tipos de enfermedades en los ojos, esto basado en imágenes médicas de la retina.

ORIGEN DATASET

Competencia Retinal Disease Classification de Kaggle.

VARIABLES INDEPENDIENTES

Imagenes de 150x150 pixeles a color

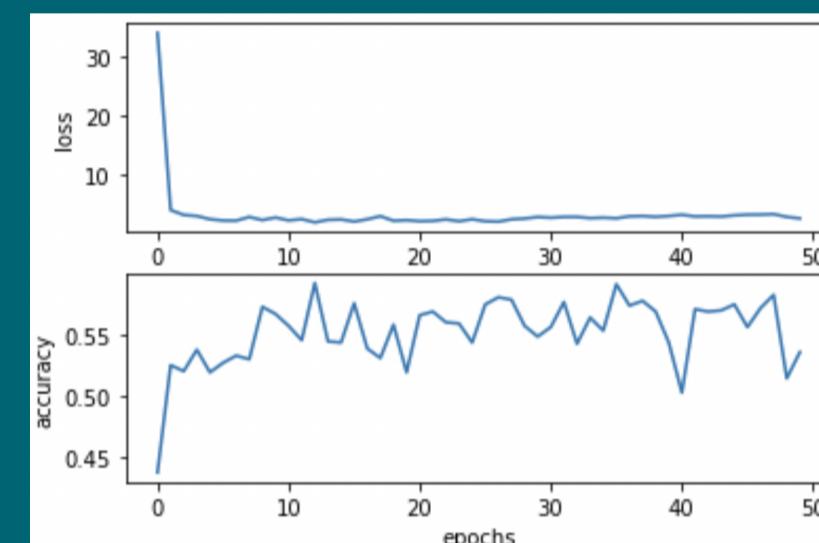
VARIABLE DEPENDIENTE

El tipo de enfermedad (Diabetic Retinopathy, Age-related Macular Degeneration, Macular Hole, Retinal Dystrophy, Myopia)

EXPERIMENTACIÓN

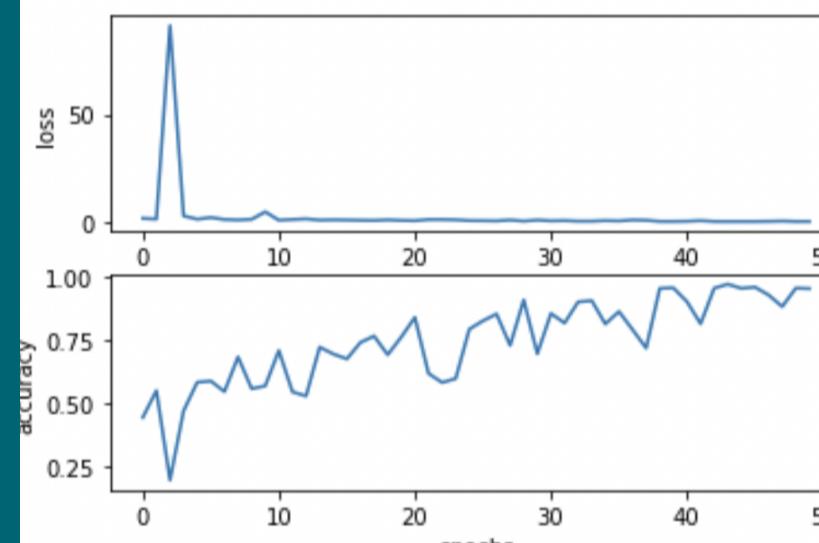
Para la experimentación se utilizaron básicamente dos arquitecturas sobre las cuales se hicieron 3 experimentos en total. A continuación se muestran las gráficas del entrenamiento de los dos mejores experimentos en su etapa de validación.

ARQUITECTURA 1



Error y exactitud en set de validación durante entrenamiento en arquitectura 1

ARQUITECTURA 2

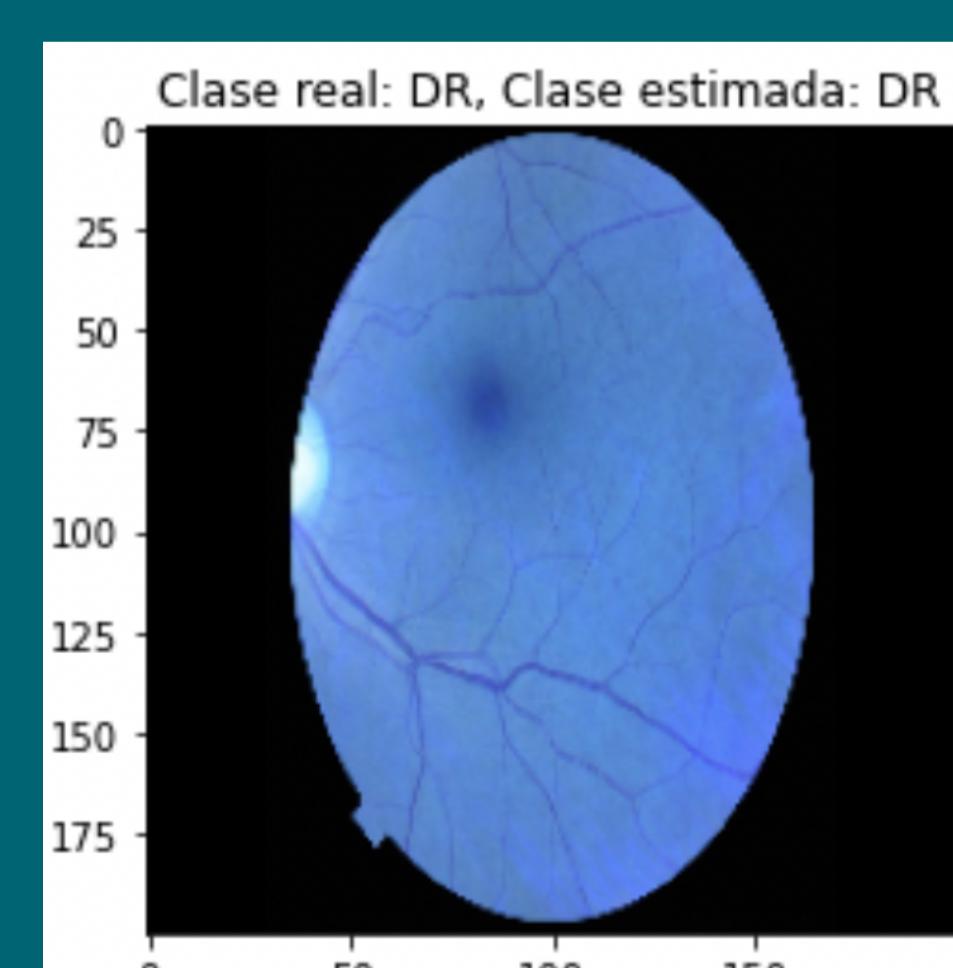


Error y exactitud en set de validación durante entrenamiento en arquitectura 2 (ganadora)

CONCLUSIONES

Para validar la generalización del modelo se evaluó este en un set de datos antes no vistos, en el cual alcanzó una exactitud de 96%, es decir que la clasificación de las 5 enfermedades antes mencionadas es una tarea confiable al utilizar este modelo. La arquitectura responsable de este resultado está descrita a continuación:

- 1 capa convolucional con 16 filtros de 3x3 activada por relu, 1 capa de maxpooling con filtro de 2x2
- 2 capas densas de 32 neuronas activadas por relu, estas capas seguidas de capas de normalización
- 1 capa de Dropout al 10%.
- 1 capa densa de 5 neuronas activada por función sigmoid



LEARNING II

PREDICCIÓN DEL PRECIO DE BITCOIN CON RNN

PROBLEMA

La predicción del precio de las criptomonedas es un tema de interés en estos días debido a su relevante oportunidad de sacar beneficio del trading. Para esta parte utilizaremos una Red Neuronal Recurrente para así poder estimar el precio del bitcoin en el tiempo. Para lograr esto, alimentaremos a nuestro modelo con observaciones que están compuestas por el precio de cierre de Bitcoin en los últimos 5 días pasados y la idea es poder predecir el precio de cierre el día siguiente.

ORIGEN DATASET

Fuente original Yahoo Finance

VARIABLES INDEPENDIENTES

Precio de cierre del Bitcoin en los últimos 5 días.

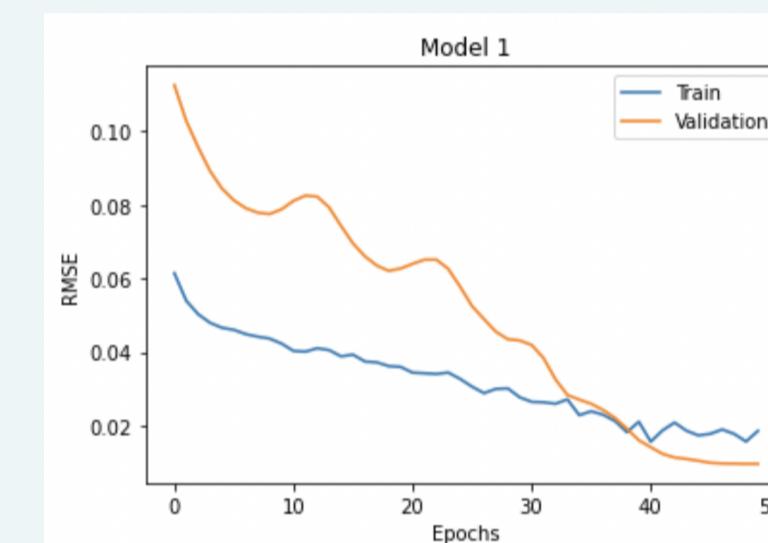
VARIABLE DEPENDIENTE

El precio de cierre del Bitcoin en el día número 6.

EXPERIMENTACIÓN

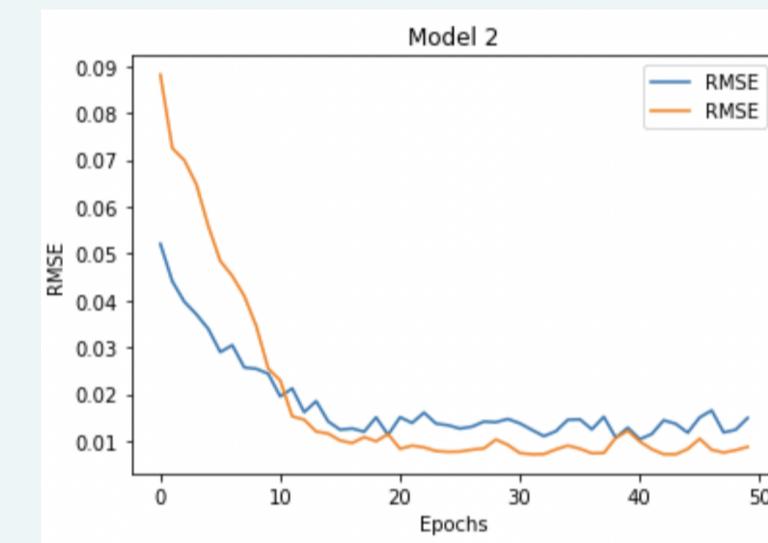
Para la etapa de experimentación se usaron 2 arquitecturas, a continuación se muestran los resultados del mejor experimento por arquitectura.

ARQUITECTURA 1



RMSE en set de entrenamiento y validación durante entrenamiento en arquitectura 1

ARQUITECTURA 2

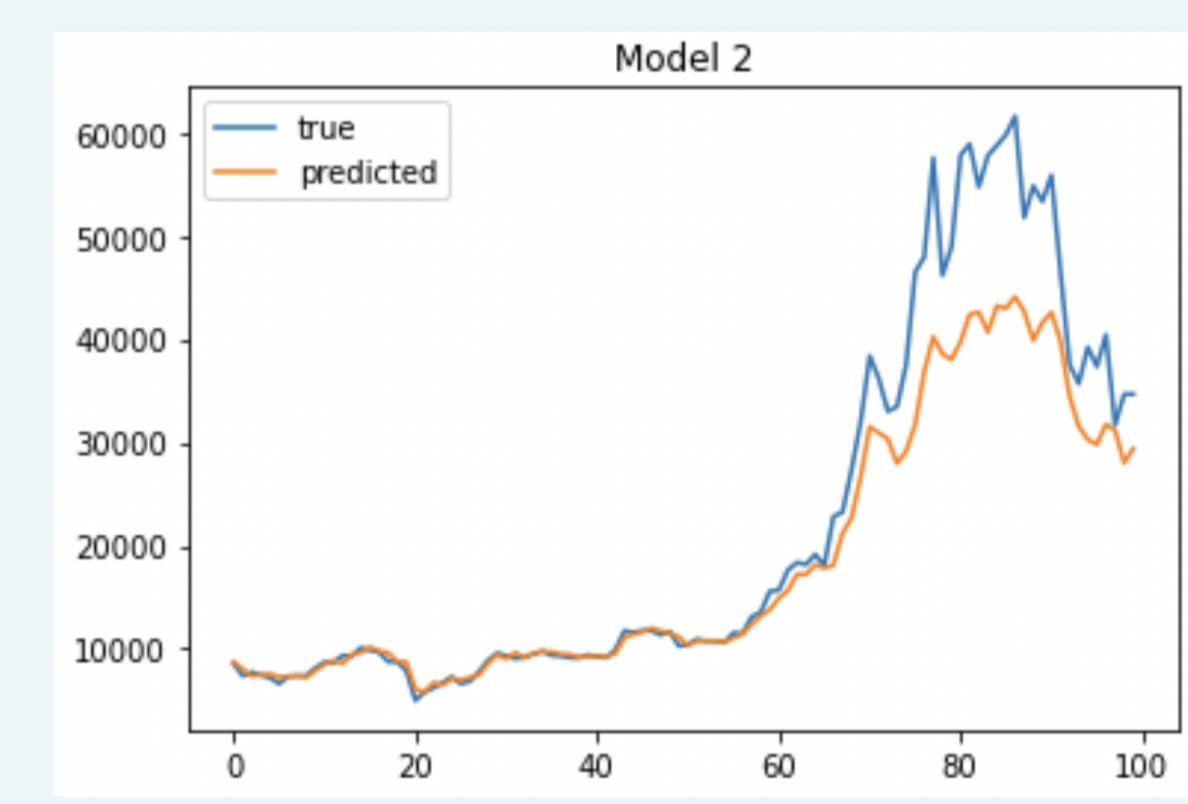


RMSE en set de entrenamiento y validación durante entrenamiento en arquitectura 2 (ganadora)

CONCLUSIONES

Se logró disminuir el error a 0.14, teniendo una exactitud del 92% en el entrenamiento y de 80% en el set de validación.

A continuación se muestra una gráfica del modelo con data de validación y es interesante como al principio bajo un comportamiento normal se comporta bien el modelo, pero cuando se llega a los puntos donde es desordenado no logra llegar a predecir el precio exacto, pero si el comportamiento.



Proyección del precio del bitcoin utilizando el modelo vs el precio real.

Este resultado se obtuvo de entrenar el modelo 2 con ventanas de 5 días hacia atrás y tratando de predecir únicamente un día adelante, el error se redujo hasta llegar a una estacionalidad en donde ya no disminuya el error solamente oscilaba en un mínimo de RMSE de 0.0011.

Esto bajo la arquitectura ganadora descrita a continuación:

- 1 capa LSTM de 5 neuronas, 1 capa densa con 32 neuronas, la segunda densa compuesta por 16 neuronas, ambas activadas por la función relu.
- 1 capa de dropout para castigar el sobre ajuste al desactivar el 20% de las salidas de la última capa oculta
- 1 capa densa de 1 neurona activación lineal