# Pruebas para que el modelo pronostique distinto de 35

# Tabla de pruebas a realizar

|  |  |  |
| --- | --- | --- |
| % | Rango de tiempo | |
| 30% | El original | Salida sea distinto de 35 |
| 40% | El original | Salida sea distinto de 35 |
| 20% | El original | Salida sea distinto de 35 |

El modelo base es el de la combinación 20% + El rango de tiempo original

Quedándonos el resto de combinaciones a realizar + el reentrenamiento del modelo y la prueba del modelo original solo para ver cómo reacciona a esas fechas.

# Prueba de variar el % que toma

Toma el 30% final de los datos en vez del 20% para test

Evaluando eso con el modelo sin reentrenar obtenemos en 25/10/2023

Que sigue dando todo 35 +algunos decimales.

Orden de menos a mas en tiempo

Evaluate on test data

16/16 [==============================] - 4s 13ms/step - loss: 0.0634

test loss 0.06340067088603973

# Prueba de variar el % que toma reentrenamiento

Mismo que el anterior

Ahora con reentrenamiento del 70% de los datos en vez del 80%

Evaluate on test data

16/16 [==============================] - 0s 13ms/step - loss: 0.0635

test loss 0.06345754861831665

Se obtuvo un valor ligeramente menor de 35 siendo 34.99 pero ninguno valor en la salida >=36

Orden de menos a mas en tiempo

Prediciendo el X\_test nos genera datos de salida

Si evaluamos los datos de salida nunca es >=36

# Prueba de variar la Salida para que sea distinto de 35

Ahora con reentrenamiento de

train\_df,test\_df = dataset[1:800], dataset[800:1250]

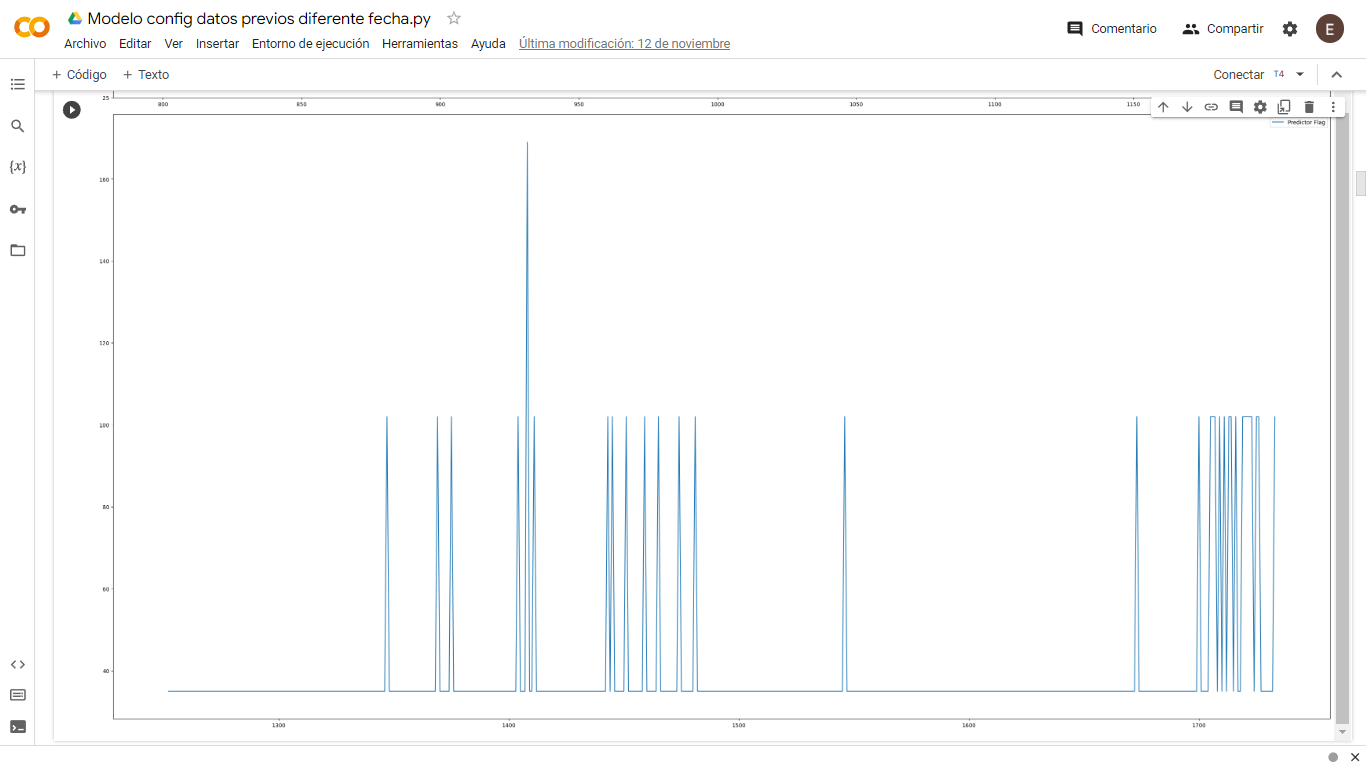
train = train\_df

train\_df2 = dataset[1250:]

Básicamente agarramos los datos de test de 800-1250

Total de test 450

Donde hay más pico



Evaluo sin reentreno

Evaluate on test data

13/13 [==============================] - 4s 17ms/step - loss: 0.1194

test loss 0.11942112445831299

Evaluo con reentreno

Evaluate on test data

13/13 [==============================] - 0s 13ms/step - loss: 0.1194

test loss 0.11943856626749039

Prediciendo el X\_test nos genera datos de salida

Si evaluamos los datos de salida nunca es >=36

# Prueba 2 de variar la Salida para que sea distinto de 35

Ahora con reentrenamiento de

train\_df,test\_df = dataset[1:950], dataset[950:1250]

train = train\_df

train\_df2 = dataset[1250:1670]

train2 = train\_df2

test\_df2 = dataset[1670:]

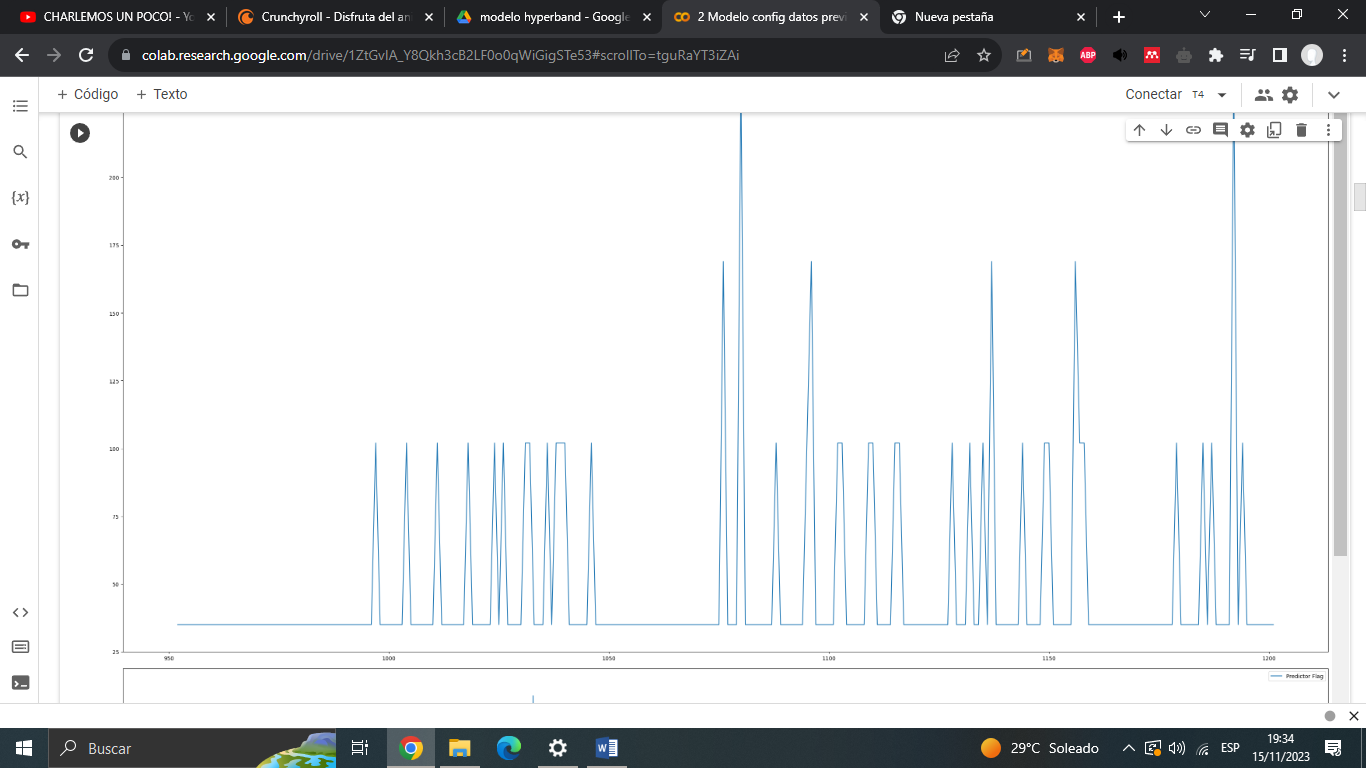
test\_df2 = test\_df2

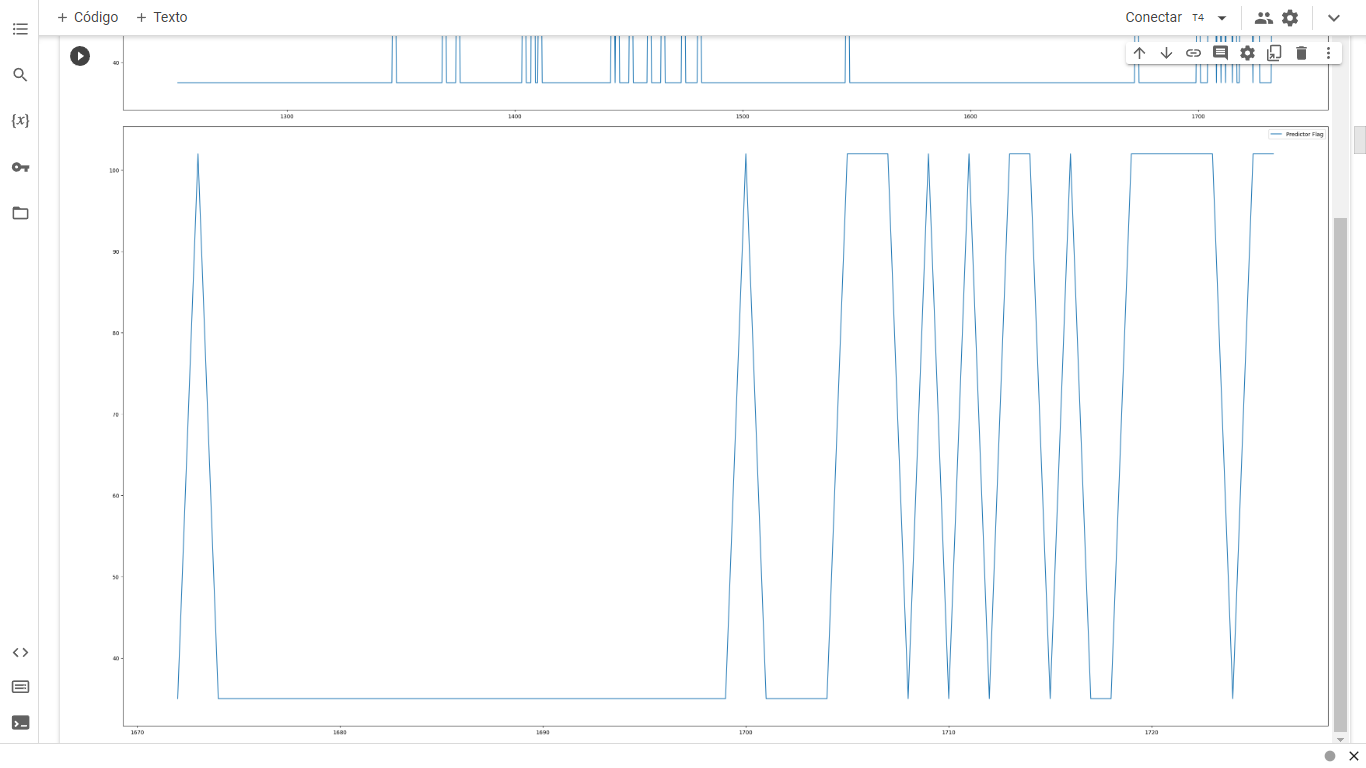
esta vez train esta dividido en 2 partes y test tambien

Donde hay más pico

dataset[950:1200].plot( y='Predictor Flag', use\_index=True)

dataset[1670:1725].plot( y='Predictor Flag', use\_index=True)





Evaluo sin reentreno

Evaluate on test data

10/10 [==============================] - 1s 24ms/step - loss: 0.1882

test loss 0.18815618753433228

Evaluo con reentreno

Evaluate on test data

10/10 [==============================] - 0s 13ms/step - loss: 0.1925

test loss 0.1925068348646164

Archivo:

2 Modelo config datos previos diferente fecha.py

Prediciendo el X\_test nos genera datos de salida

Si evaluamos los datos de salida nunca es >=36

# Prueba 3 de variar la Salida para que sea distinto de 35

Train está dividido en 3 partes para que cuando entrene aprenda de las zonas donde hay más picos de vistas, test también tiene pico de datos

Se quitó en 1250:1670 que solo tenía 3.3% de datos >= 35

No se tomaron en cuenta zonas donde hay solo 35 en predictor flag en un largo lapso de tiempo

train\_df,test\_df = dataset[0:100], dataset[950:1250]

train = train\_df

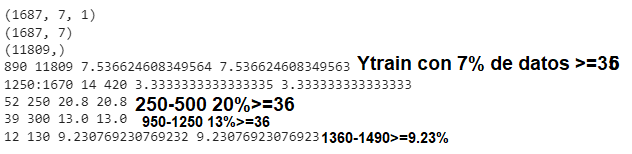
train\_df2 = dataset[250:500]

train\_df3 = dataset[1360:1490]#dataset[1250:1670]

train2 = train\_df2

test\_df2 = dataset[1670:]

test\_df2 = test\_df2



Resultado sin reentrenar

Evaluate on test data

8/8 [==============================] - 1s 13ms/step - loss: 0.2631

test loss 0.263082891702652

Resultado con reentrenamiento

Evaluate on test data

10/10 [==============================] - 0s 13ms/step - loss: 0.1925

test loss 0.1925068348646164

Si evaluamos los datos de salida nunca es >=36

# Prueba 4 Hypertuner de variar la Salida para que sea distinto de 35

No se usó el mejor modelo anterior

Se probó pasar por hypertuner de hyperband para ver si generaba una salida >=36

Estos modelos si se guardaron

train\_df,test\_df = dataset[1:950], dataset[950:1250]

train = train\_df

train\_df2 = dataset[1250:1670]

train2 = train\_df2

test\_df2 = dataset[1670:]

test\_df2 = test\_df2

max\_trials 50- epochs 100

Resultado del mejor modelo

Trial 0072 summary

Hyperparameters:

units: 288

lr: 0.0016406586383193105

dropout: 0.1

recurrent\_dropout: 0.4

batch\_size: 64

tuner/epochs: 50

tuner/initial\_epoch: 17

tuner/bracket: 2

tuner/round: 2

tuner/trial\_id: 0067

Score: 0.18816892802715302

results\_summary <class 'NoneType'>

Prediciendo el X\_test nos genera datos de salida

Si evaluamos los datos de salida nunca es >=36

# Prueba 5 Hypertuner de variar la Salida para que sea distinto de 35

No se usó el mejor modelo anterior

Con estos datos

train\_df,test\_df = dataset[1:950], dataset[950:1250]

train = train\_df

train\_df2 = dataset[1250:1670]

train2 = train\_df2

test\_df2 = dataset[1670:]

test\_df2 = test\_df2

MAPE-max\_trials100- epochs200-aleatorio

Misma configuracion de datos train test que el anterior pero con mas épocas

Salida del mejor modelo

Trial 0234 summary

Hyperparameters:

units: 160

lr: 0.0007479781936524693

dropout: 0.30000000000000004

recurrent\_dropout: 0.2

batch\_size: 32

tuner/epochs: 100

tuner/initial\_epoch: 34

tuner/bracket: 2

tuner/round: 2

tuner/trial\_id: 0230

Score: 0.18816962838172913

results\_summary <class 'NoneType'>

Prediciendo el X\_test nos genera datos de salida

Si evaluamos los datos de salida nunca es >=36

# Prueba 6 Hypertuner de variar la Salida para que sea distinto de 35

No se usó el mejor modelo anterior

Con estos datos

train\_df,test\_df = dataset[0:100], dataset[950:1250]

train = train\_df

train\_df2 = dataset[250:500]

train\_df3 = dataset[1360:1490]#dataset[1250:1670]

train2 = train\_df2

test\_df2 = dataset[1670:]

test\_df2 = test\_df2

Cambios de dataset[1250:1670] a dataset[1360:1490] para que train tenga mayor % de valores >=36

Resultado del mejor modelo

Trial 0245 summary

Hyperparameters:

units: 192

lr: 0.0028156208463983876

dropout: 0.0

recurrent\_dropout: 0.0

batch\_size: 32

tuner/epochs: 100

tuner/initial\_epoch: 34

tuner/bracket: 1

tuner/round: 1

tuner/trial\_id: 0240

Score: 0.1882011443376541

results\_summary <class 'NoneType'>

Prediciendo el X\_test nos genera datos de salida

Si evaluamos los datos de salida nunca es >=36

# Sobre actualizar el anteproyecto

El anteproyecto presentado hacía referencia a un modelo o modelos que funcione sobre las paginas institucionales, eventualmente sabemos que solo fue sobre 1 pagina (tum transmedia).

* Sitio Institucional de la U.Na.M. disponible en <https://unam.edu.ar/>
* Sitio de la editorial Universitaria disponible en <https://editorial.unam.edu.ar/>
* Portal de acceso al contenido generado por la radio y la televisión de la Universidad disponible en <https://transmedia.unam.edu.ar/>

Orden de tesis copiado de los “ejemplos”

capitulo 1-introduccion

capitulo 2-marco teorico

si mi solución usa RNN GRU la explicación de esto debe de estar en el marco teórico o va en la solución propuesta,

capitulo 3-descripccion del problema

### debería de hacer una entrevista a la gente que me dio acceso al google analytics? Marcelo Puerta, mencionar el orgramigrama y cuestiones por el estilo?

capitulo 4- materiales y métodos (este capitulo aparece en la tesis de bayco), un capitulo para hablar sobre

“se describe la metodología que se ha utilizado como guía para el desarrollo del trabajo y las herramientas software utilizadas durante el proceso”

capitulo 4- solucion propuesta

todas las cuestiones referente a hypertuner, obtención de datos

capitulo 5- prueba experimental

se especifican las conclusiones obtenidas a partir del trabajo realizado y los posibles trabajos futuros

Capítulo 6 se especifican las conclusiones obtenidas a partir del trabajo realizado y los posibles trabajos futuros

Anexos

Bibliografia