Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Inteligencia Artificial
Recibido: 05/02/2023 | Aceptado: 13/03/2023 | Publicado: 30/03/2023

Identificadores persistentes:
ARK: ark:/42411/s11/a87
PURL: 42411/s11/a87

Uso de las redes neuronales para determinar la calificación de una aplicación publicada en Google Play Store

Use of neural networks to determine the rating of an application published in the Google Play Store

Rudy Roberto Tito Durand[1], Marcelo Andre Guevara Gutierrez[2], Jeampier Anderson Moran Fuño[3], Edsel Yael Alvan Ventura[4]

[1] Universidad Nacional de San Agustín. Arequipa, Perú. rtitod@unsa.edu.pe
[2] Universidad Nacional de San Agustín. Arequipa, Perú. mguevarag@unsa.edu.pe
[3] Universidad Nacional de San Agustín. Arequipa, Perú. jmoran@unsa.edu.pe
[4] Universidad Nacional de San Agustín. Arequipa, Perú. ealvan@unsa.edu.pe

[*] Autor para correspondencia: jmoran@unsa.edu.pe

Resumen

La inteligencia artificial es la combinación de algoritmos escritos en forma de código computacional con el fin de que se ejecuten en una computadora para emular comportamientos similares a la inteligencia humana. En este trabajo, se buscará el uso de las redes neuronales, las cuales son parte de la inteligencia artificial y, nos permiten solucionar problemas de predicción. Se modificará un dataset basado en las descargas de aplicaciones de Google Play Store y sus características y se procesará la información para obtener información de salida.

Palabras clave: redes neuronales, numpy, flujo tensorial, valores perdidos, valores atípicos

Abstract

Artificial intelligence is the combination of algorithms written in the form of computer code in order to be executed on a computer to emulate behaviors similar to human intelligence. In this work, the use of neural networks will be sought, which are part of artificial intelligence and allow us to solve prediction problems. A dataset based on Google Play Store app downloads and their features will be modified and the information processed to obtain output information.

Keywords: *neural networks, numpy,tensorflow, missing values, outliers.*

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

Introducción

La inteligencia artificial es la combinación de algoritmos escritos como código computacional con el fin de que las computadoras en donde se ejecuten estos códigos ejecuten comportamientos similares a la inteligencia humana. Desde su invención , ha tenido muchas aplicaciones. Las aplicaciones que se le pueden dar a la Inteligencia Artificial han estado incrementando en casi todos los dominios de la vida humana desde los sistemas de recomendación para compras, gestión de inventarios y aspectos logísticos hasta la solución de problemas de salud [1].

Existe un gran número de problemas dentro de la ingeniería en los cuales se puede aplicar Inteligencia Artificial pero debido a la naturaleza de los datos con los que se trabaja no se puede tratar de las formas tradicionales. Esto tardó ya que recién en los años 70 entendieron el hecho de que era imposible el manejo de las millones de combinaciones posibles necesarias para poder evaluar situaciones reales, esto debido al infinito número de perturbaciones que pueden existir dentro de una sola situación [2]. Las redes neuronales son sistemas de procesamiento de información, que a su vez son reconocidas como un paradigma matemático de computación. Son ampliamente utilizadas en diversos ambientes teóricos y prácticos, son un conjunto de unidades llamadas neuronas artificiales conectadas [3].

Como un problema relacionado con los aspectos logísticos está relacionado con el desarrollo de software para móviles y si una empresa que inicia puede saber si una aplicación va a ser exitosa o no podría condicionar el desarrollo, evitando pérdidas económicas o buscando mayores ganancias al redefinir el proyecto. Estudios realizados por diferentes organizaciones demuestran que la tasa de éxito de los proyectos está alrededor del 32%, y de proyectos fallidos alrededor del 24% [4]. A través de este proyecto se busca predecir a partir de factores vinculados al software desarrollado , que calificación se puede obtener al ser puesto al público, a través de Google Play, que es la tienda de aplicaciones creada por Google donde puedes encontrar juegos, películas, música, libros y más, la cual está disponible para cualquier dispositivo móvil que cuente con sistema operativo Android [5], aunque hay estudios que aclaran que la calificación de una aplicación no necesariamente indica que será la más popular [6]. Para lograr tal fin, se usará una de las herramientas de la inteligencia artificial la cual son las redes neuronales, porque son un método de resolver

problemas, por ejemplo, el mapeo autoorganizado suelen ser utilizado como herramienta para la predicción de tendencias y como clasificador de conjuntos de datos. [7]. Las cuales consisten en unidades de procesamiento interconectadas de manera densa, llamadas neuronas; una de sus características más importantes es la capacidad del aprendizaje adaptativo mediante datos dados, además pueden ser combinadas con otras herramientas como la lógica difusa, los algoritmos genéticos o los sistemas expertos [8]. Se buscará implantar un modelo estándar que a su vez, requerirá de sistemas de información apoyados en datos históricos, para lo cual estas redes neuronales leerán la información de un archivo CSV, el cual será tratado en una fase preliminar para eliminar o completar información errónea obtenida en la fase de recolección.

Marco Teórico

Logística

Para Ferrel, Hirt, Adriaenséns, Flores y Ramos, la logística es "una función operativa importante que comprende todas las actividades necesarias para la obtención y administración de materias primas y componentes, así como el manejo de los productos terminados, su empaque y su distribución a los clientes"[9]

Redes neuronales

Las redes neuronales simulan la estructura y el comportamiento del cerebro, usan lo que se conoce como procesos de aprendizaje para dar solución a los problemas para los que fueron programadas; son un conjunto de algoritmos matemáticos que encuentran las relaciones no lineales entre conjuntos de datos; suelen ser utilizadas como herramientas para la predicción de tendencias y como clasificadoras de conjuntos de datos[18]. Se denominan neuronales porque están basadas en el funcionamiento de una neurona biológica cuando procesa información[10].

Data Cleaning

Se conoce como al proceso que se encarga de corregir los errores en los datos, se convierte por tanto en un mecanismo necesario para que las estadísticas, los informes y en última instancia las decisiones

que se tomen por los directivos sean confiables, pues en la medida en que esté garantizada la calidad de los datos, así mismo habrá seguridad y fiabilidad en las acciones posteriores que se produzcan a partir de su análisis.[11]

Data Set

Un Dataset no es más que un conjunto de datos tabulados en cualquier sistema de almacenamiento de datos estructurado. El término hace referencia a una única base de datos de origen, la cual se puede relacionar con otras, cada columna del Dataset representa una variable y cada fila corresponde a cualquier dato que estemos tratando[12]

Red Neuronal Secuencial

Una red neuronal sequential es un tipo de modelo de red neuronal que se conforma por capas de neuronas, cada capa se agrega una después de la otra. La metodología usada durante la construcción del modelo es paso a paso y trabajando en una sola capa en un momento determinado.

Herramientas

Colab Research

Es una herramienta para escribir y ejecutar código Python en la nube.de Google. También es posible incluir texto enriquecido, “links” e imágenes. En caso de necesitar altas prestaciones de cómputo, el entorno permite configurar algunas propiedades del equipo sobre el que se ejecuta el código.[13]

Python

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario “traducirlo” a lenguaje máquina.[15]

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

TensorFlow

Tensor Flow es una biblioteca de software de código abierto para computación numérica, que utiliza gráficos de flujo de datos. Los nodos en las gráficas representan operaciones matemáticas, mientras que los bordes de las gráficas representan las matrices de datos multidimensionales (tensores) comunicadas entre ellos.

Tensor Flow es una gran plataforma para construir y entrenar redes neuronales, que permiten detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos.[16]

Keras

Keras es una biblioteca que funciona a nivel de modelo: proporciona bloques modulares sobre los que se pueden desarrollar modelos complejos de aprendizaje profundo. A diferencia de los frameworks, este software de código abierto no se utiliza para operaciones sencillas de bajo nivel, sino que utiliza las bibliotecas de los frameworks de aprendizaje automático vinculadas, que en cierto modo actúan como un motor de backend para Keras.[17]

Métodos y Metodología computacional

Se usó para el desarrollo de este detector de sitios web fraudulentos usamos el modelo CRISP-DM, la cual comprende de seis fases: análisis del problema, análisis de Datos, preparación de los Datos, modelado, evaluación y explotación [10].

- La fuente de información fue un Dataset.csv, esto fue sacado de internet con datos reales, para poder construir un árbol de decisión utilizando las librerías *sklearn*, y se hizo uso de las funciones que nos brinda.
- Análisis del problema: Identificamos qué tipos de clasificación tendrán las páginas que se van a probar, viendo que pueden ser tres, y también identificamos la información necesaria para poder hacer esta clasificación.
- Análisis de Datos: con el *dataset* que obtuvimos, vimos que cantidad de información tiene, y las variables lingüísticas las clasificamos en números para su uso correcto con la librería que se trabajaría.
- Preparación de Datos: Se hizo una limpieza del *dataset*, con las funciones que nos brinda la libre, como eliminación de campos vacíos, datos irrelevantes.

- Modelado: Seleccionamos la técnica adecuada para poder hacer la clasificación.
- Evaluación: Tuvimos que corroborar efectivamente que el modelo escogido se ajuste a lo que estamos buscando, en este caso poder clasificar los diferentes sitios web.

Resultados y discusión

Resultados y Discusión

Análisis del Problema

Los datos de las aplicaciones de Play Store tienen un enorme potencial para impulsar a las empresas para la creación de aplicaciones exitosas.

Por lo cual pretendemos realizar una red neuronal que permita la predicción de ratings usando como datos de entrenamiento una fuente de datos provenientes de un dataset recolectado de las descargas de la Google Play Store, este consiste en datos extraídos de la web de 10,000 aplicaciones de Play Store.

Análisis de los Datos

Fase de data cleaning:

El dataset posee 13 variables (columnas) las cuales son: App, Category, Rating, Reviews, Size, Installs, Type, Price, Content Rating, Genres, Last Updated, Current Ver y Android Ver. Se cuenta con 10841 filas.

Luego de la descarga del dataset se procedió al análisis para eliminar los valores missing.

Fase de Transformación:

En la fase de transformación, en la sección Genre se detectó que existían datos múltiples valores en la variable, pues esta variable proveía mas de un valor en sus celdas, por lo que se decidió dividir la cadena en dos tokens usando el símbolo “;” y quedándonos con el primer token obtenido. En el caso de la variable “Content Rating”,”Category”, se decidió conservar cada uno de sus datos para la codificación. Mientras que para las variables “Last Updated”, se procedió a la transformación de sus valores restando la fecha indicada en la fila y el año de publicación aproximado del dataset. La variable

"Android Ver", se transformó en un número de tipo float, debido a que el orden de precedencia de las versiones esta determinado por la posición que ocupa. Con respecto a la variable "Size", se quitó y transformó cada valor por el sufijo que tuvo al final. Por ejemplo si el valor es "233k", entonces esto indicaría que debemos multiplicar 233*0.001, mientras que si se tenía "233M", se conservaría igual, en síntesis, se transformó y se puso en una misma unidad de medida en la variable "Size". Con respecto a la variable "Installs" se transformó a números enteros y se les redimensiona a un valor general que es el valor actual entre 100,000. Cabe destacar que todos los valores "Varies with Device" en las filas de "Android Ver", "Size", entre otras, fueron eliminadas pues no representan una gran cantidad de datos y se decidió eliminarlas debido a la falta de datos que este valor provee.

Fase de eliminación de Outliers:

Para la fase de eliminación de outliers se procedió a identificar los cuartiles de los datos que fueron convertidos en la anterior fase. El método usado para la eliminación de outliers que se usó fue la eliminación por el valor máximo y mínimo dados por el Método Caja y Bigotes.

En el cual se determina que el valor máximo y minimo estan dados en la siguiente formula:

$$máximo = q3+(1.5*iqr)$$

$$mínimo = q1+(1.5*iqr)$$

Donde "q1" es el cuartil que divide a los datos en 25% y 75%. Mientras que "q3", representa el tercer cuartil representa que los valores por debajo de ese valor son el 75% de los datos. Con respecto a "iqr", es la diferencia entre el tercer cuartil y el primero. Los valores de máximo y mínimo son los valores que nos ayudarán a identificar los outliers de nuestra gráfica de Caja y Bigotes. Se procedió a reemplazar(en algunos casos eliminar) a los valores outliers con la media del total de valores, de esta manera no afectamos al balance de los datos y por lo tanto se conservó los demás atributos de esas filas para el entrenamiento de nuestro modelo Secuencial de red Neuronal.

Fase de creación de red neuronal:

- En esta fase se identifican cada uno de los valores como "Categórico" o "Continuo", por lo tanto las variables "Content Rating","Category" se consideran categóricos mientras que las variables "Android

Ver", "Installs","Reviews" son considerados valores continuos, pues los valores pueden incrementar o disminuir, pero no se mantienen en un rango fijo de valores a los cuales es posible clasificarlos.

Luego de este análisis, se procedió a categorizar todos los valores como una variable en específico, es decir todos los valores categóricos han sido puestos como características propias de la Aplicación. De este modo si hay las características categóricas A,B y C, se le asigna este tipo de codificación a sus valores:

Nombre de la Aplicación	A	B	C
App A	1	0	0

La tabla anterior representa que la aplicación "App A" tiene la característica A, y no tiene las características B y C. Esto beneficia al modelo Secuencial, pues los valores altos o secuenciales 1,2,3,4,5 pueden afectar a los pesos de cada neurona. Por lo tanto este modelo de codificación tendrá más precisión a la hora de entrenar nuestro modelo de red neuronal.

A) Preparación de los datos

1. Los valores missings se trataron asi:

- A continuación se muestran la tabla de variables que contienen valores missing y sus respectivos porcentajes:

	Total	%
Rating	1474	13.597
Current Ver	8	0.074
Android Ver	3	0.028
Type	1	0.009
Content Rating	1	0.009

- Luego de eliminar los valores missing en la variable objetivo "Rating", es muestra los valores missing restantes:

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

DATAFRAME REMOVIDO

	Total	%
Current Ver	4	0.043
Android Ver	3	0.032
Content Rating	1	0.011

- Se eliminan los otros missing Values
- Se observa las filas anteriores con respecto las nuevas filas sin valores missing:

```
Original: (10841, 13)
Clean dataframe: (9360, 13)
```

2. **Se procede a un análisis de las columnas a utilizar**

- 'App' : No se utilizara porque es el nombre
- 'Category': Si se utilizara
- 'Rating': Es nuestro OUTPUT
- 'Reviews': Si se utilizara
- 'Size': Si se utilizara
- 'Installs': Si se utiliza
- 'Type': No se utilizara porque la mayoría son valores Free
- 'Price': No se utilizara porque la mayoría son valores 0
- 'Content Rating': Si se utilizara
- 'Genres': No se utilizara porque era redundante respecto a Category.
- 'Last Updated': Se utilizara
- 'Current Ver': No se utilizará porque es un valor subjetivo dependiendo del creador de la App.
- 'Android Ver': Se utilizara

En este sentido las únicas variables a usar son: Reviews, Category, Rating,Size, Installs, Content Rating, Last Updated, Current Ver, Android Ver.

3. **Se arreglan los datos de algunas columnas**

- Al ya haber realizado la obtención de los datos, se deben analizar cuáles requieren de una transformación para que sean útiles.

- Para arreglar los datos del campo Size debemos convertir todos sus datos a la misma base, en este caso serían kilobytes, para ello se asigna un valor para poder convertir los datos a esta base, y esta acción se repite en todas los registros de la tabla eliminando el carácter y transformando el dato en un float y reemplazandolo en la tabla..

```
print("DF-SHAPE:",df.shape)
M = 1
k = 0.001
varies_devices=0
for index, row in df.iterrows():
  if(row["Size"][-1] == "M"):
    r = row["Size"][:-1]
    s = float(r)
    s = s*M
    row["Size"] = s*M
  elif(row["Size"][-1] == "k"):
    r = row["Size"][:-1]
    s = float(r)
    s = s*k
    row["Size"] = s*k
```

- Ahora también existe un valor no numérico “varies in device” que básicamente indica que el tamaño depende del dispositivo en el cual se instale la aplicación evaluada, debido a que no debemos indagar el peso de la aplicación en otra fuente, debemos descartar estos y para eso tenemos un else dentro del bucle que contabiliza cuantas veces se repite este valor en el campo Size de la tabla

```
else:
  varies_devices+=1
  pass#sacar o rescatar los "varies with device"
df.at[index,"Size"] = s
  #df.set_value(index,'Size',s)
```

- Después de haber evaluado cada fila de la tabla se nos muestra la siguiente información y los datos ya registrados como floats de 64 bits

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

```
#plt.show()
print("Varies with device: ", varies_devices)
df["Size"].describe()
```

```
DF-SHAPE: (9360, 13)
Varies with device:  1637
count    9360.000000
mean       23.143466
std        23.245147
min         0.008500
25%         5.500000
50%        15.000000
75%        33.000000
max       100.000000
Name: Size, dtype: float64
```

- Como ya se ha evaluado toda la tabla ya tenemos una vista más limpia de los datos de el campo Size con datos numéricos y ya retirados los valores “varies in device”

```
df["Size"].head(10)
```

```
0    19.0
1    14.0
2     8.7
3    25.0
4     2.8
5     5.6
6    19.0
7    29.0
8    33.0
9     3.1
Name: Size, dtype: float64
```

- Para arreglar los datos del campo Installs, lo que se requiere es eliminar el signo “+”, además que en algunos casos se están trabajando con números demasiado grandes, por lo que convendría reducirlos para no ocupar tanto espacio por el tamaño de los números

```
array(['10,000+', '500,000+', '5,000,000+', '50,000,000+', '100,000+',
       '50,000+', '1,000,000+', '10,000,000+', '5,000+', '100,000,000+',
       '1,000,000,000+', '1,000+', '500,000,000+', '100+', '500+', '10+',
       '5+', '50+', '1+'], dtype=object)
```

- Para lo cual se ha tomado como base el valor de 100.000 para poder reducir el tamaño de los valores originales. logrando así reducir espacio, y al retirar el símbolo “+” se establecen como floats de 64 bits

```
for i, row in df.iterrows():
  tmp = row["Installs"].replace("+","")
  tmp = tmp.replace(",","")
  df.at[i,"Installs"] = str(float(tmp)/100000)

df["Installs"] = df["Installs"].astype(float)
```

- Como ya se ha evaluado toda la tabla ya tenemos una vista más limpia de los datos de el campo Installs con datos numéricos más manejables

```
0       0.1
1       5.0
2      50.0
3     500.0
4       1.0
5       0.5
6       0.5
7      10.0
8      10.0
9       0.1
10     10.0
11     10.0
Name: Installs, dtype: float64
```

- Ahora en el caso del campo Reviews, lo único que se hace es establecer el tipo del campo como floats ya que los datos originales de por sí son manejables y no requieren mayor conversion

```
df_test = df.copy()
df_test["Reviews"] = df_test["Reviews"].astype(float)
```

- Aquí se ve ya convertido a float los datos iniciales

Reviews
159.0
967.0
87510.0
215644.0
967.0

- Para el caso de Genres el obstáculo que tenemos es que algunos registros poseen más de un géneros y no podemos clasificar las fusiones de géneros para realizar análisis, afortunadamente cuando es más de un género en el registro están separados por un “;” por lo que este será el límite para separar dichas fusiones obteniendo todos los valores diferentes dentro de la tabla.

```
df_test["Genres"].unique()
```

```
array(['Art & Design', 'Art & Design;Pretend Play',
       'Art & Design;Creativity', 'Auto & Vehicles', 'Beauty',
       'Books & Reference', 'Business', 'Comics', 'Comics;Creativity',
       'Communication', 'Dating', 'Education;Education', 'Education',
       'Education;Creativity', 'Education;Music & Video',
       'Education;Action & Adventure', 'Education;Pretend Play',
       'Education;Brain Games', 'Entertainment',
       'Entertainment;Music & Video', 'Entertainment;Brain Games',
       'Entertainment;Creativity', 'Events', 'Finance', 'Food & Drink',
       'Health & Fitness', 'House & Home', 'Libraries & Demo',
       'Lifestyle', 'Lifestyle;Pretend Play',
       'Adventure;Action & Adventure', 'Arcade', 'Casual', 'Card',
       'Casual;Pretend Play', 'Action', 'Strategy', 'Puzzle', 'Sports',
       'Music', 'Word', 'Racing', 'Casual;Creativity',
       'Casual;Action & Adventure', 'Simulation', 'Adventure', 'Board',
       'Trivia', 'Role Playing', 'Simulation;Education',
       'Action;Action & Adventure', 'Casual;Brain Games',
       'Simulation;Action & Adventure', 'Educational;Creativity',
       'Puzzle;Brain Games', 'Educational;Education', 'Card;Brain Games',
       'Educational;Brain Games', 'Educational;Pretend Play',
       'Entertainment;Education', 'Casual;Education',
```

- Luego de hallar todos los valores diferentes pasamos a separar las fusiones en elementos individuales y así obtenemos los géneros únicos

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

```
[ ]  df_test["Genres"]=df_test["Genres"].str.split(';').str[0]
```

```
[ ]  df_test["Genres"].unique()
```

```
array(['Art & Design', 'Auto & Vehicles', 'Beauty', 'Books & Reference',
       'Business', 'Comics', 'Communication', 'Dating', 'Education',
       'Entertainment', 'Events', 'Finance', 'Food & Drink',
       'Health & Fitness', 'House & Home', 'Libraries & Demo',
       'Lifestyle', 'Adventure', 'Arcade', 'Casual', 'Card', 'Action',
       'Strategy', 'Puzzle', 'Sports', 'Music', 'Word', 'Racing',
       'Simulation', 'Board', 'Trivia', 'Role Playing', 'Educational',
       'Music & Audio', 'Video Players & Editors', 'Medical', 'Social',
       'Shopping', 'Photography', 'Travel & Local', 'Tools',
       'Personalization', 'Productivity', 'Parenting', 'Weather',
       'News & Magazines', 'Maps & Navigation', 'Casino'], dtype=object)
```

- Ya habiendo obtenido los géneros individuales ya se puede realizar un conteo de registros coincidentes con cada género.

```
df_test.groupby("Genres")["App"].nunique()
```

```
Genres
Action                304
Adventure              78
Arcade                186
Art & Design           61
Auto & Vehicles        73
Beauty                 42
Board                  57
Books & Reference     171
Business              263
Card                   46
Casino                 37
Casual                217
Comics                 54
Communication         258
Dating                134
Education             498
Educational            93
Entertainment         502
Events                 45
Finance               302
Food & Drink           94
Health & Fitness      246
House & Home           62
Libraries & Demo       63
Lifestyle             302
Maps & Navigation     118
Medical               291
```

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

- Para el campo Last Update ya que se esta trabajando con fechas debemos convertir la cadena de caracteres de toda la tabla a un valor de tipo Date, y como el formato en el que se presentan en estas cadenas es en casi toda la tabla es solo separar los valores y registrarlos con el tipo Date

```
print("CURRENT DAY : ",current_date)
date_pub = current_date - timedelta(days=365*4)#FUE HACE 4 AÑOS

# date_now = datetime.today().date()#BUSCAR LA FECHA DE PUBLICACION DEL DATASET
date = date_pub
NO = 0
import math
for i, row in df_test.iterrows():
  tmp = row["Last Updated"]
  try:
    format = datetime.strptime(tmp,"%B %d, %Y").date()
    days = (date_pub-format).days
    df_test.at[i,"Last Updated"] = days
  except:
    df_test.at[i,"Last Updated"] = "NO"
    print("No")
    NO +=1
condition = df_test.loc[df_test["Last Updated"] == "NO"]
df_test.drop(condition.index, inplace=True)
print("Total Exceptions: ",condition.shape,"->", NO)
print("Total Success: ", df_test.shape)
```

- Ya para trabajar es más sencillo trabajar con datos numéricos que con fechas cambiaremos el significado del campo, pasando de registrar la fecha de la última actualización, se contarán los días desde la última actualización de cada aplicación de la tabla hasta el dia de hoy y pasa de ser de tipo Date a tipo float

```
[ ]  #CONVIRTIENDO A FLOAT-> PARA MAS ADELANTE USARLO EN RED NEURONAL
     df_test["Last Updated"] = df_test["Last Updated"].astype(float)
```

```
[ ]  df = df_test
     df["Last Updated"].head(30)
     df.rename(columns = {'Last Updated':'Days passed',}, inplace = True)
```

- Por último para arreglar los datos de Android ver, hay caracteres que obstaculizan el análisis, la cadena “and up”, “nan” y “NO”, lo primero se debe retirar dichos caracteres

```
for i, row in df_test.iterrows():
  val = row["Android Ver"]
  if(val == "Varies with device" or val=="nan"):
    tmp = "NO"
    total_out+=1
  else:
    tmp = val.replace("and up","")
    total_succ += 1
  df_test.at[i,"Android Ver"] = tmp
print("#Success: ", total_succ)
print("#Fails: ", total_out)
print("#Total", df_test.shape)
```

```
#Success:  8041
#Fails:  1319
#Total (9360, 13)
```

- Luego tenemos algunos rangos por ejemplo “4.0 - 5.0” en el cual se nos muestran 2 valores, en este caso se toma el primer valor osea el menor, así obtenemos todos los valores diferentes y realizamos un conteo

```
for i, row in df_test.iterrows():
  val = row["Android Ver"]
  if val != "NO":
    val = val[0:3]
    df_test.at[i,"Android Ver"] = val
```

```
df_test.groupby("Android Ver")["App"].nunique()
```

```
Android Ver
1.0       2
1.5      15
1.6      87
2.0      34
2.1     112
2.2     203
2.3     780
3.0     201
3.1       8
```

- Ya por ultimo convertimos estos valores de las versiones a floats

```
arr= np.array(df_test["Android Ver"].unique())
arr = arr[arr != "NO"]
another = np.array([])
for item in arr:
  another = np.append(another, float(item))
arr.astype(float)
print(arr,another)
mean = np.mean(another)
print("mean: ", mean)
```

```
['4.0' '4.2' '4.4' '2.3' '3.0' '4.1' '2.2' '5.0' '6.0' '1.6' '1.5' '2.1'
 '7.0' '4.3' '2.0' '3.2' '5.1' '7.1' '8.0' '3.1' '1.0'] [4.  4.2 4.4 2.3 3.  4.1 2.2 5.  6.  1.6 1.5 2.1 7.  4.3 2.  3.2 5.1 7.1
 8.  3.1 1. ]
```

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

```
df_test["Android Ver"].astype(float)#convirtiendo a float
```

```
0       4.000
1       4.000
2       4.000
3       4.200
4       4.400
        ...
9355    4.100
9356    4.100
9357    4.100
9358    3.867
9359    3.867
Name: Android Ver, Length: 9360, dtype: float64
```

4. Los outliers se trataron asi:

En el paso anterior se arreglaron los datos con el fin de obtener coherencia en el resultado de la red neuronal. Se tiene el dataset con 13 columnas y 9360 filas

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Days passed	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159.0	19.0	0.100	Free	0	Everyone	Art & Design	223.0	1.0.0	4.000
1	Coloring book moana	ART_AND_DESIGN	3.9	967.0	14.0	5.000	Free	0	Everyone	Art & Design	215.0	2.0.0	4.000
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510.0	8.7	50.000	Free	0	Everyone	Art & Design	17.0	1.2.4	4.000
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644.0	25.0	500.000	Free	0	Teen	Art & Design	71.0	Varies with device	4.200
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967.0	2.8	1.000	Free	0	Everyone	Art & Design	69.0	1.1	4.400
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9355	FR Calculator	FAMILY	4.0	7.0	2.6	0.005	Free	0	Everyone	Education	426.0	1.0.0	4.100
9356	Syafa Maroc - FR	FAMILY	4.5	38.0	53.0	0.050	Free	0	Everyone	Education	389.0	1.48	4.100
9357	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4.0	3.6	0.001	Free	0	Everyone	Education	43.0	1.0	4.100
9358	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114.0	3.6	0.010	Free	0	Mature 17+	Books & Reference	1307.0	Varies with device	3.867
9359	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307.0	19.0	100.000	Free	0	Everyone	Lifestyle	24.0	Varies with device	3.867

9360 rows × 13 columns

Se puede observar los tipos de datos, resumidos a object(string) y float64

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

```
[32] df.dtypes

App                object
Category           object
Rating            float64
Reviews           float64
Size              float64
Installs          float64
Type               object
Price              object
Content Rating     object
Genres             object
Days passed       float64
Current Ver        object
Android Ver       float64
dtype: object
```

Se realizó una función que detecta outliers en las columnas del dataset (una observación anormal y extrema en una muestra estadística). Se calcula el valor iqr (Los Outliers son los puntos que caen a más de 1.5 veces el IQR, a partir de la caja) y se hallan límites superiores e inferiores, con los cuales se creará una estructura con los outliers y sus valores:

```
[33] def detect_outlier(df,cols):
       outlier_dict = {}
       for col in cols:
         print(col)
         q1 = np.quantile(df[col],0.25)
         q3 = np.quantile(df[col],0.75)
         iqr = q3 - q1
         upper = q3+(1.5*iqr)
         lower = q1-(1.5*iqr)
         outlier = df.loc[ ((df[col]>=upper) | (df[col]<=lower) )][[col]]
         outlier_dict[col] = [q1,q3,iqr,lower,upper,outlier.values]
       return outlier_dict
```

Función para remover outliers

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

```
def showData(column,df_outlier):
  lower = df_outlier[column]["lower"]
  upper = df_outlier[column]["upper"]
  q1 = df_outlier[column]["q1"]
  q3 = df_outlier[column]["q3"]
  iqr =  df_outlier[column]["inter_q"]
  print("q1=",q1,"\nq3=",q3,"\niqr=",iqr,"\nupper=",upper,"\nlower=",lower)
  return (lower,upper)

def remove_outliers(df,column,df_outlier,method="capping"):
  lower,upper = showData(column,df_outlier)
  new_df = df.copy()
  outliers = new_df.loc[(df[column] >= upper) | (df[column] <= lower)]
  print("METHOD= ",method)
  if(method.lower() == "capping"):
    #show boxplot before capping
    # sns.boxplot(new_df[column])
    #do capping
    new_df.loc[(new_df[column] > upper), column] = df[column].mean()#upper#PONER LA MEDIA/MEDIANA
    new_df.loc[(new_df[column] < lower), column] = df[column].mean()#lower#PONER LA MEDIA/MEDIANA
    #show after capping
    # sns.boxplot(new_df[column])
  elif(method.lower() == "remove"):
    #show boxplot before has removed
    sns.boxplot(new_df[column])
    #Removiing...
    new_df = new_df.loc[(df[column] < upper) & (df[column] > lower)]
    #showing after removing outliers
    sns.boxplot(new_df[column])

  print("#rows of Original DF: ",len(df))
  print("#rows of New DF: ",len(new_df))
  print("#Outliers: ", len(outliers), len(df)-len(new_df))
  return (new_df,outliers)
```

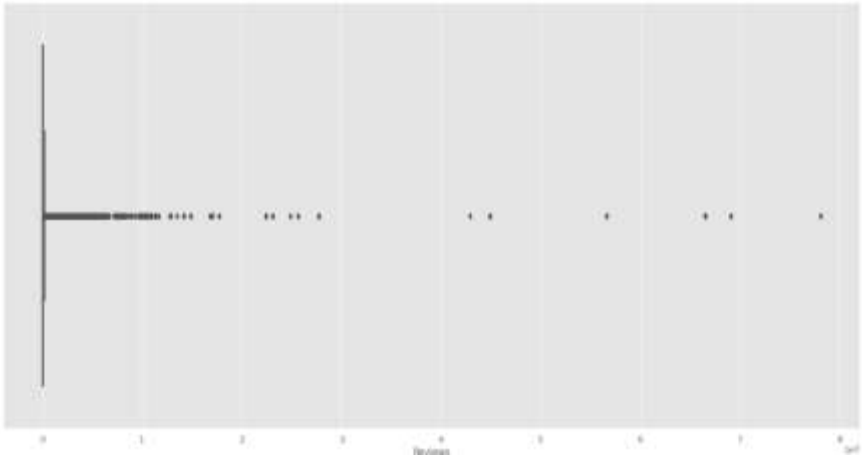
Al procesar la información con la función detect_outliers, se obtuvieron los siguientes valores

	Reviews	Size	Installs	Days passed	Android Ver
q1	186.75	5.5	0.1	25.0	3.867
q3	81627.5	33.0	50.0	313.0	4.1
inter_q	81440.75	27.5	49.9	288.0	0.233
lower	-121974.375	-35.75	-74.75	-407.0	3.5175
upper	203788.625	74.25	124.85	745.0	4.4495
outlier	[[215644.0], [224299.0], [295221.0], [271920.0...	[[77.0], [77.0], [84.0], [97.0], [76.0], [76.0...	[[500.0], [1000.0], [1000.0], [10000.0], [500....	[[1453.0], [1126.0], [1402.0], [857.0], [1073....	[[3.3], [3.0], [3.3], [3.3], [3.0], [3.3], [3....

En base a esto, se corrigen los outliers para cada una de las variables de la tabla anterior

Para reviews se grafica primero para ver la magnitud del problema

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

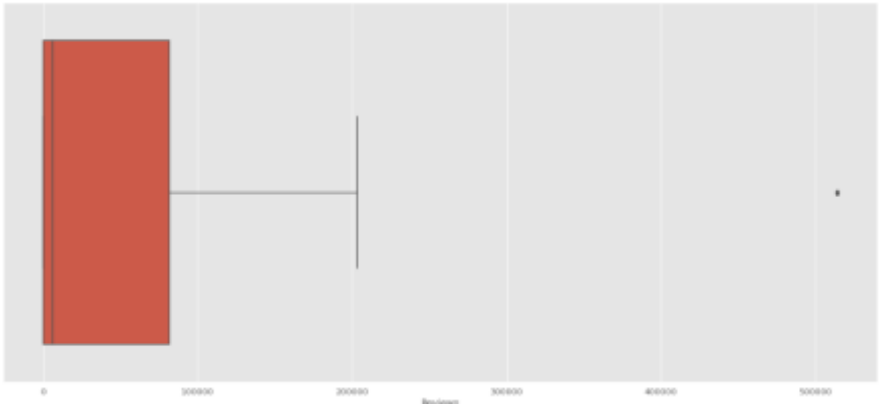


Se realiza un llamado a la función *remove_outliers*() para remover los outliers. Se ha de aclarar que se usará el método capping para tal fin:

```
df_test, outliers = remove_outliers(df_test,"Reviews",df_outlier,method="Capping")
```

```
q1= 186.75
q3= 81627.5
iqr= 81440.75
upper= 203788.625
lower= -121974.375
METHOD=  Capping
#rows of Original DF:  9360
#rows of New DF:  9360
#Outliers:  1634 0
```

Se visualiza el gráfico de cajas y bigotes:



Se visualizan los resultados:

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

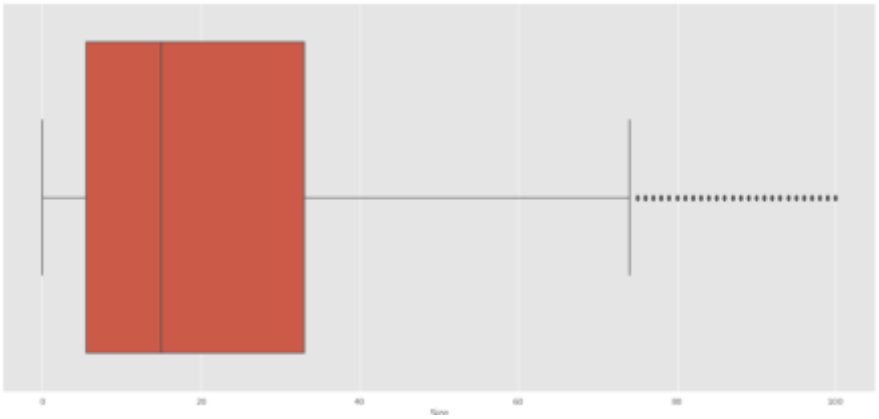
```
df_test.loc[df_test["Reviews"]>500000,"Reviews"] = df_test["Reviews"].mean()
```

```
df_test["Reviews"].describe()
```

```
count      9360.000000
mean      36645.841596
std       49036.140636
min           1.000000
25%         186.750000
50%        5955.000000
75%       81627.500000
max      203130.000000
Name: Reviews, dtype: float64
```

Para sizes tenemos lo siguiente:

Se visualiza el gráfico de cajas y bigotes donde se puede ver los puntos separados de la caja del diagrama:



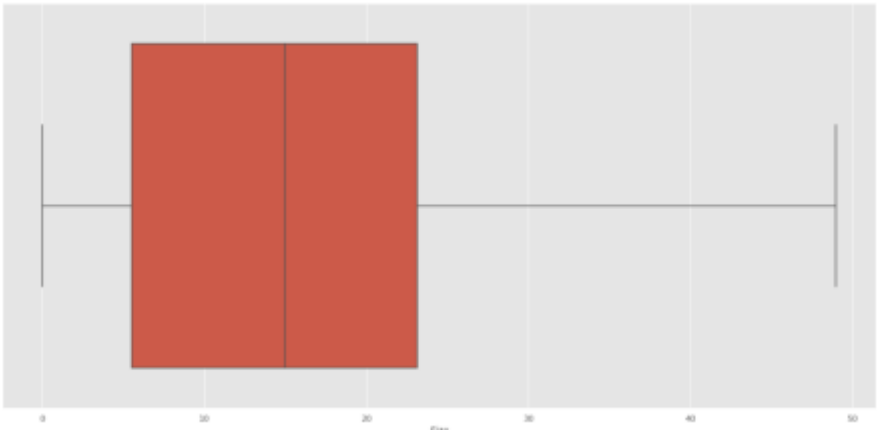
Un análisis de los datos arroja lo siguiente:

	Size	Android Ver
q1	5.5	3.867
q3	33.0	4.1
inter_q	27.5	0.233
lower	-35.75	3.5175
upper	74.25	4.4495
outlier	[[77.0], [77.0], [84.0], [97.0], [76.0], [76.0...	[[2.3], [3.0], [2.3], [2.3], [3.0], [2.3], [2....

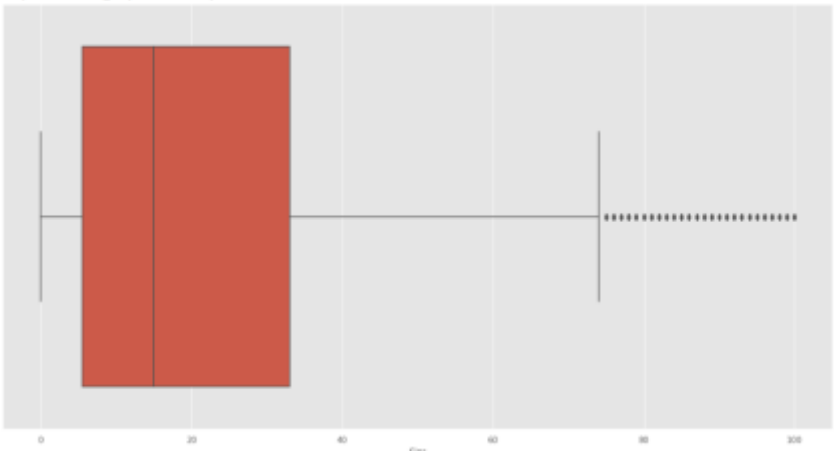
Se hace el llamado para corregir ese problema usando capping:

```
df_test, outliers = remove_outliers(df_test,"Size",df_outlier,method="Capping")

q1= 5.5
q3= 33.0
iqr= 27.5
upper= 74.25
lower= -35.75
METHOD=  Capping
#rows of Original DF:  9360
#rows of New DF:  9360
#Outliers:  490 0
```

Se resolvió el problema de outliers:



Para Installs tenemos que tiene puntos distanciados considerablemente de la caja:



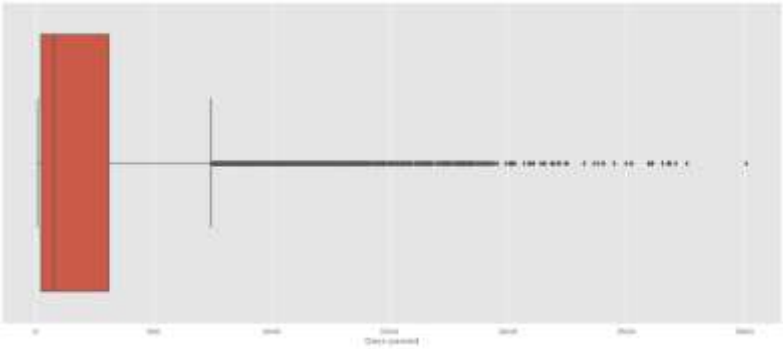
Análisis de límites de los datos y visualización de los outliers

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

	Installs
q1	0.1
q3	50.0
inter_q	49.9
lower	-74.75
upper	124.85
outlier	[[500.0], [1000.0], [1000.0], [10000.0], [500....

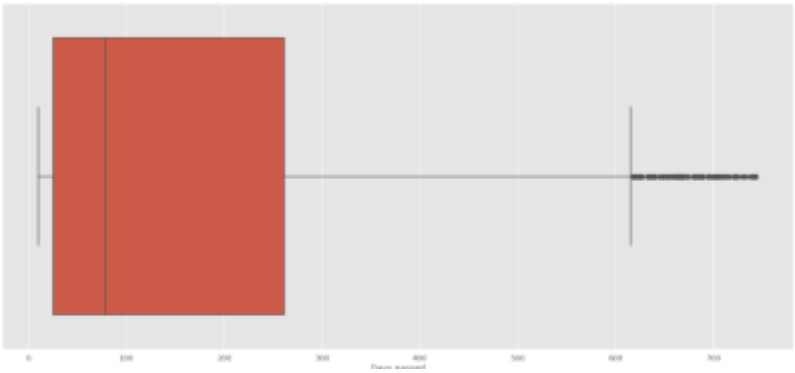
Luego de la corrección se puede ver el siguiente gráfico en función a los datos corregidos :

Para Days Passed tenemos lo siguiente:

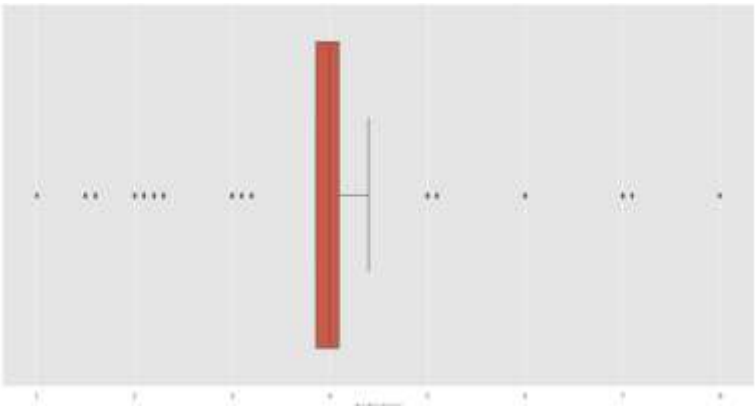


Se visualiza la corrección de los datos:

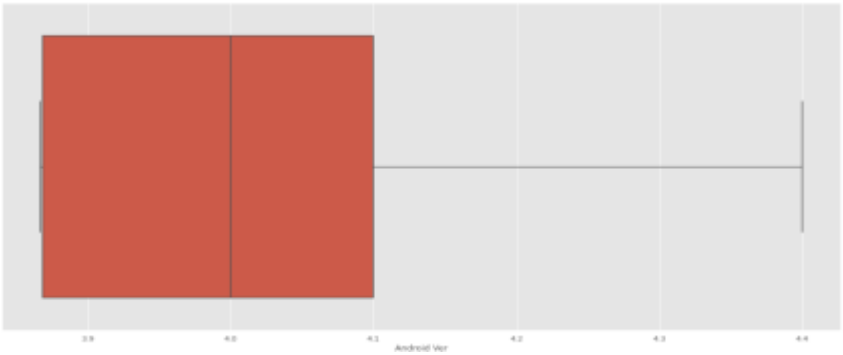
Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft



Para Android Version tenemos lo siguiente:



Finalmente se ve la corrección de los datos:



5. Redes Neuronales

- Se seguirán unos ciertos pasos

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

PASOS

- CODIFICAR Y/O CATEGORIZAR
- CONTRUIR EL MODELO
- COMPILAR
- ENTRENAR
- PREDECIR

- Ahora se removerán las columnas que no se utilizaran

REMOVE COLUMNS

```
[ ]  dataset.drop(columns=["Type","Price","Genres","Current Ver","App"],inplace=True)
```

- Luego se ordenara la tabla

Category	Reviews	Size	Installs	Content Rating	Days passed	Android Ver	Rating

- Observamos la columna “Category” esta columna.

```
df.groupby("Category").nunique()
```

AUTO_AND_VEHICLES	18	65	50	11	3	49	7
BEAUTY	14	41	30	10	3	37	5
BOOKS_AND_REFERENCE	20	139	90	14	4	143	6
BUSINESS	31	191	112	15	2	177	7
COMICS	19	55	44	10	5	39	7
COMMUNICATION	25	166	120	14	3	161	7
DATING	27	136	68	12	3	64	7
EDUCATION	13	111	57	9	4	79	6
ENTERTAINMENT	16	80	41	8	4	53	7
EVENTS	14	40	34	11	3	42	6

- Se procede a codificar “Category” y los valores que poseía esta columna

	Reviews	Size	Installs	Content Rating	Days passed	Android Ver	Rating	ART_AND_DESIGN	AUTO_AND_VEHICLES	BEAUTY	...	SPORTS	TRAVEL_AND_LOCAL	TOOLS	PERSONALIZATION	PRODUCTIVITY	PARENTING	WEATHER	VIDEO_PLAYERS	NEWS_AND_MAGAZIN
0	159.000000	19.0	0.100000	Everyone	222.0	4.000	4.1	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1	967.000000	14.0	5.000000	Everyone	214.0	4.000	3.9	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	87510.000000	8.7	50.000000	Everyone	16.0	4.000	4.7	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
3	107645.319559	25.0	10.307683	Teen	75.0	4.200	4.3	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
4	967.000000	2.8	1.000000	Everyone	58.0	4.400	4.3	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

- Se procede a codificar “Rating y los valores que poseía esta columna”

- Se procede a tener dos variables para entrenar y los datos para el testing

```
train_dataset = dataset.sample(frac=0.8,random_state=0)
test_dataset = dataset.drop(train_dataset.index)
```

Ahora se procede a normalizar

- Primero se obtienen las estadísticas

```
train_stats = train_dataset.describe()
train_stats.pop("Rating")
train_stats = train_stats.transpose()
train_stats
```

	count	mean	std	min	25%	50%	75%	max
Reviews	7488.0	36646.543287	49051.651664	1.00000	190.750000	5998.5	80909.750000	203130.0
Size	7488.0	23.383751	23.385373	0.00850	5.600000	15.0	33.000000	100.0
Installs	7488.0	12.258876	15.685476	0.00001	0.100000	5.0	22.654223	50.0
Days passed	7488.0	259.702858	396.855706	9.00000	24.000000	76.0	311.000000	3010.0
Android Ver	7488.0	4.023885	0.163303	3.86700	3.868779	4.0	4.100000	4.4
ART_AND_DESIGN	7488.0	0.006944	0.083049	0.00000	0.000000	0.0	0.000000	1.0

- Se obtienen las etiquetas del train y test (Se quita Rating porque es nuestro output)

```
train_labels = train_dataset.pop('Rating')
#En este instante, se borro Rating en train_dataset
test_labels = test_dataset.pop('Rating')
#En este instante, se borro Rating en test_dataset
```

- Se normalizan los datos (Con los datos de traint_stats)

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

```
def norm(x):
  return (x - train_stats['mean']) / train_stats['std']

normed_train_data = norm(train_dataset)
normed_test_data = norm(test_dataset)
```

- Resultado de la tabla *normed_train_data*

- Se construye el modelo y ejecuta

```
def build_model():
  model = tf.keras.Sequential([
    tf.keras.layers.Dense(32, activation='relu', input_shape=[len(train_dataset.keys())]),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(1)
  ])
  #OPTIMIZERS
  optimizer = tf.keras.optimizers.RMSprop(0.001)
  #COMPILE
  model.compile(loss='mse',
                optimizer=optimizer,
                metrics=['mae', 'mse'])
  return model
```

- Se entrena al modelo

```
history = model.fit(
  normed_train_data, train_labels,
  epochs=EPOCHS, validation_split = 0.2, verbose=0,
  callbacks=[PrintDot()])
```

- Se realiza un ejemplo de predicción

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

```
example_batch = normed_train_data[:10]
example_result = model.predict(example_batch)
example_result
```

```
array([[4.396088 ],
       [3.8018947],
       [4.0594616],
       [4.571604 ],
       [3.97579  ],
       [3.6644804],
       [2.948675 ],
       [4.6022177],
       [4.6018677],
       [4.2051067]], dtype=float32)
```

- Se imprime la historia de los 1000 epoch iterados

```
hist = pd.DataFrame(history.history)
hist['epoch'] = history.epoch
hist.tail()
```

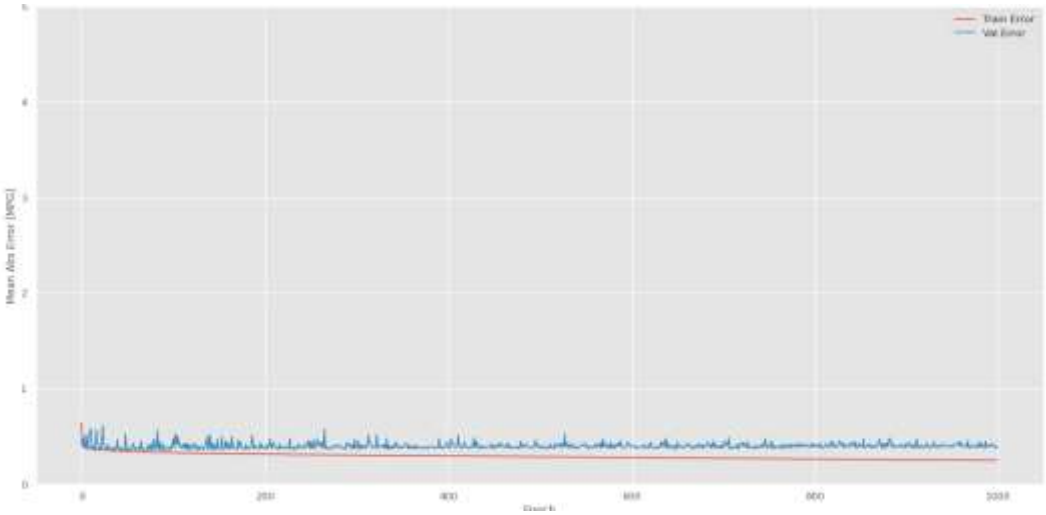
	loss	mae	mse	val_loss	val_mae	val_mse	epoch
995	0.141928	0.248724	0.141928	0.352853	0.391610	0.352853	995
996	0.143987	0.249435	0.143987	0.370625	0.404294	0.370625	996
997	0.142637	0.246242	0.142637	0.311485	0.379217	0.311485	997
998	0.143063	0.250076	0.143063	0.347423	0.391196	0.347423	998
999	0.138982	0.247082	0.138982	0.340982	0.379805	0.340982	999

- Se crea una función para ver el error de entrenamiento

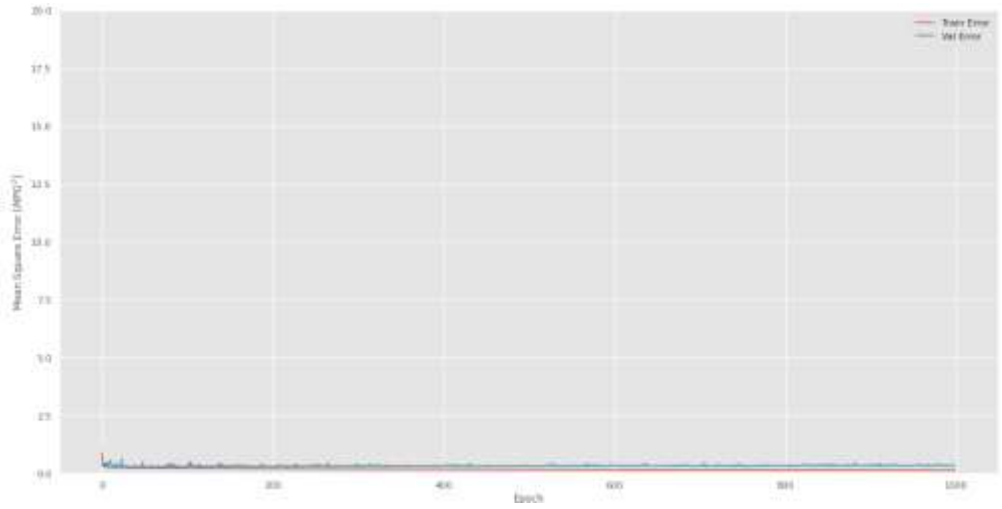
Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

```
def plot_history(history):
  hist = pd.DataFrame(history.history)
  hist['epoch'] = history.epoch

  plt.figure()
  plt.xlabel('Epoch')
  plt.ylabel('Mean Abs Error [Rating]')
  plt.plot(hist['epoch'], hist['mae'],
           label='Train Error')
  plt.plot(hist['epoch'], hist['val_mae'],
           label = 'Val Error')
  plt.ylim([0,5])
  plt.legend()

  plt.figure()
  plt.xlabel('Epoch')
  plt.ylabel('Mean Square Error [$Rating^2$]')
  plt.plot(hist['epoch'], hist['mse'],
           label='Train Error')
  plt.plot(hist['epoch'], hist['val_mse'],
           label = 'Val Error')
  plt.ylim([0,20])
  plt.legend()
  plt.show()


plot_history(history)
```

- Se grafica el error cuadrático medio y el error absoluto medio



Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

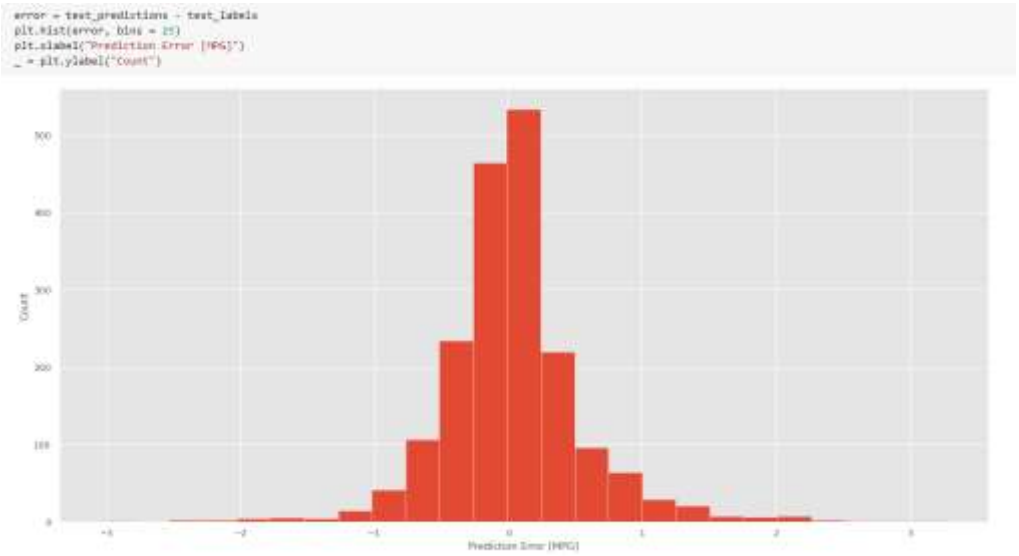


- Se analizan las predicciones

```
loss, mae, mse = model.evaluate(normed_test_data, test_labels, verbose=2)

print("Testing set Mean Abs Error: {:5.2f} MPG".format(mae))
```

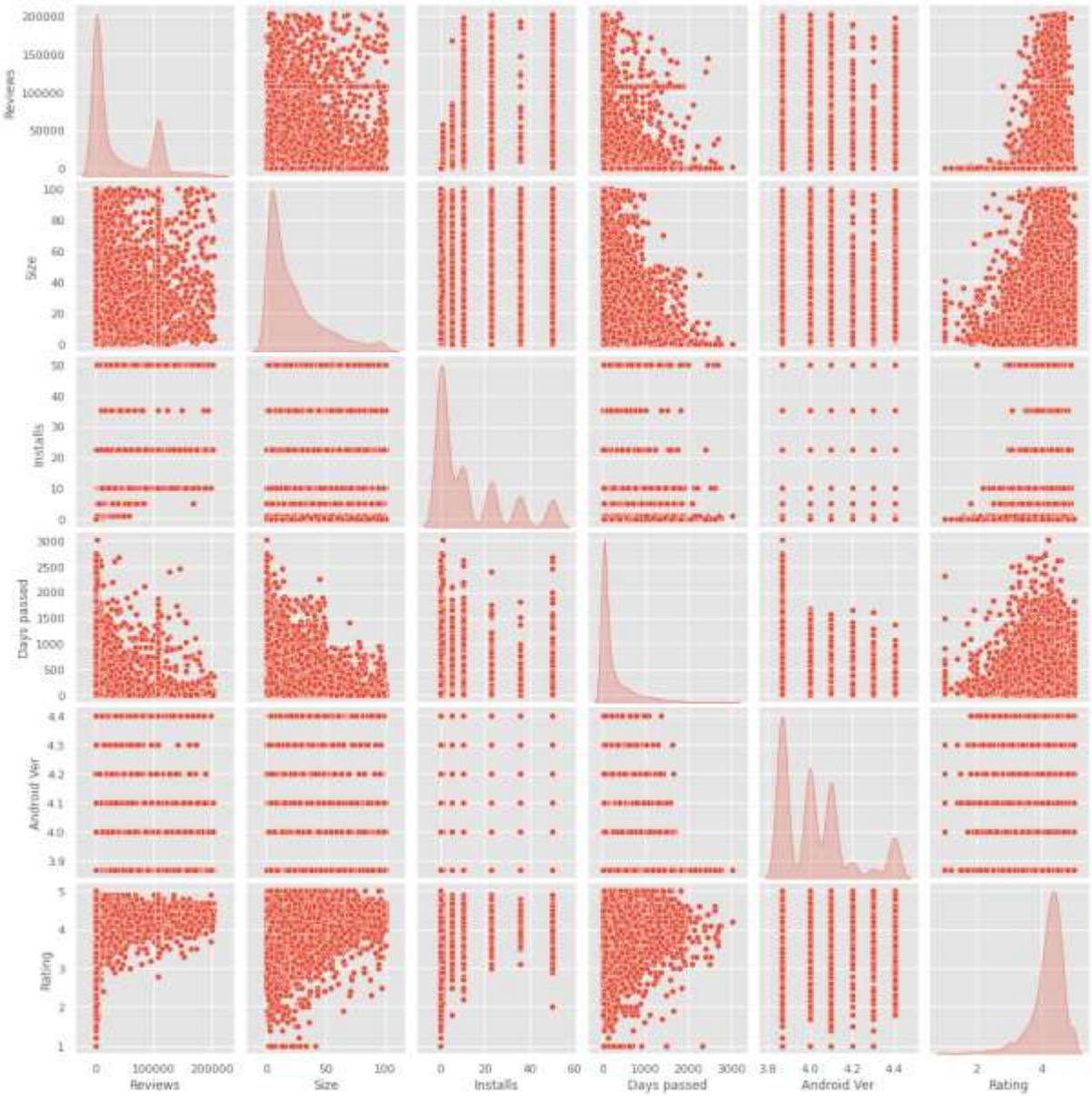
```
59/59 - 0s - loss: 0.2757 - mae: 0.3544 - mse: 0.2757 - 87ms/epoch - 1ms/step
Testing set Mean Abs Error:  0.35 MPG
```

- Gráfico en el error medio absoluto



Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

- Se usa *pairplot* que genera una gráfica de pares de variables. nos permite ver tanto la distribución de variables individuales como las relaciones entre dos variables



Evaluación

Para la evaluación de nuestros datos, se usaron dos variantes de un mismo modelo, en el cual el modelo versión 1, se implementó sin un modelo de estrategia determinado para corregir a la red neuronal con respecto al error cuadrático medio.

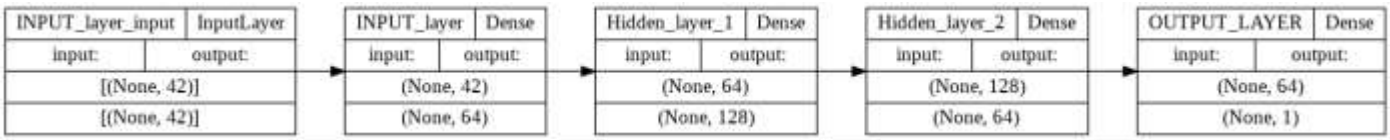
Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

Mientras que en el modelo versión 2, si se usó una estrategia de corrección llamada “EarlyStopping”, este tipo de técnica logra corregir a la red neuronal según el error absoluto medio, el cual deberá bajar en cada iteración o época de entrenamiento, pero si el valor aumenta con respecto al anterior entonces se debe orientar a la red neuronal para que consiga un Error Absoluto Medio(considerado en el parámetro *“monitor”*) menor en cada iteración, para esto se debe especificar el parámetro “*patience*”, el cual indicará hasta cuantas épocas es permitido el valor de incremento y cada cuanto corregir este valor.

Modelo Genérico:

El modelo genérico para las dos versiones de siguientes está determinado por esta arquitectura:

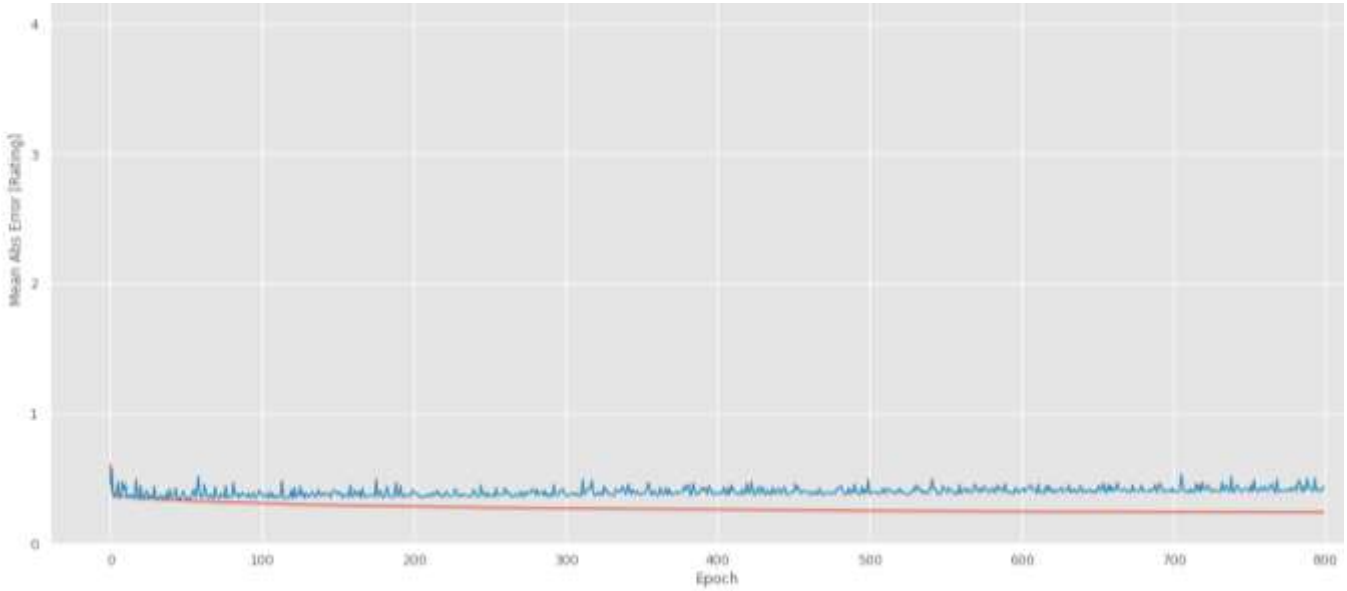
1. La primera capa, está constituida por el Input Layer, donde entran los datos de las variables de cada columna.
2. La segunda, tercera y cuarta columna son las denominadas “Hidden Layers”, las cuales iniciarán con pesos aleatorios en sus conexiones y a medida se entrene el modelo, se modificarán las neuronas conforme a los patrones de los datos.
3. Y finalmente, la última capa es llamada “Output Layer”, donde se entregará el valor final de la variable Rating predecida.
4.



Modelo 1:

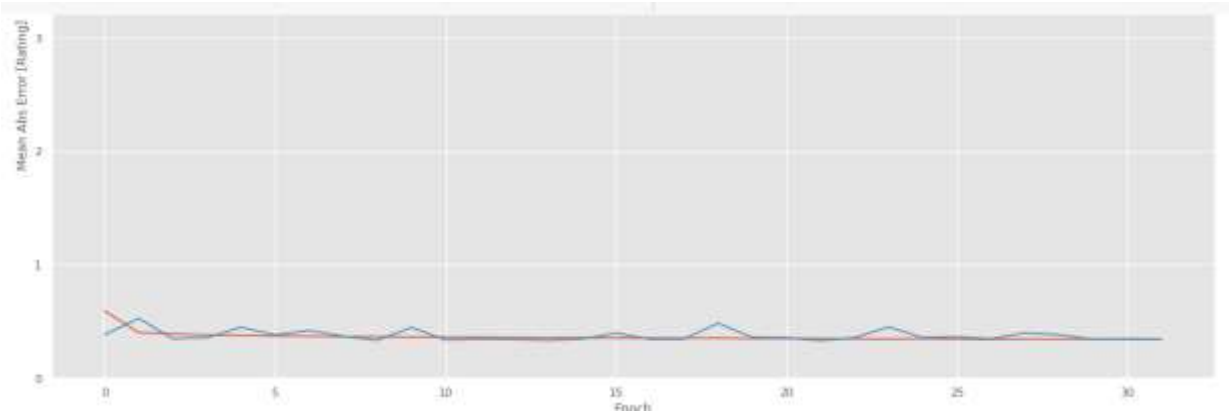
A continuación, se muestra el Mean Absolute Error con respecto a cada época, se puede ver claramente que la diferencia entre el Mean Absolute Error entre los valores predecidos y los valores verdaderos se aleja a medida que las épocas se desarrollan.

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft



Modelo versión 2:

En este modelo, se puede ver a simple vista que los valores convergen y se acercan más y más, debido a la estrategia "EarlyStopping" que corregirá a los valores predecidos conforme al lote de testeo modificando así a las redes neuronales cada vez que las diferencias de Mean Absolute Error son mayores con una corrección de cada 10 épocas.



Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 161-197
https://revistas.ulasalle.edu.pe/innosoft

Métricas del Modelo 1 y Modelo 2:

Luego fueron desarrolladas las métricas con respecto a los dos modelos, como se presenta en la siguiente tabla de comparaciones:

Modelo/Métricas	Mean Absolute Error	Mean Square Error
Modelo versión 1	0.4582	0.4286
Modelo versión 2	0.3581	0.2870

Implementación

En base al análisis realizado se realizó la implementación de la red neuronal usando la herramienta Collaboratory de Google Research, el cual nos permite ejecutar código desde notebooks(páginas donde el código puede ser dividido , con el fin de poder ser ejecutados en pasos) . Se creó un directorio ubicado en el Google Drive personal de cada desarrollador (/MyDrive/INTELIGENCIA_ARTIFICIAL) donde se colocó el archivo original que contiene al dataset, este directorio también sirvió para poder recibir la respuesta de la aplicación, la cual contenía el dataset modificado en su fase de limpieza.

En el momento de la ejecución el servicio de google nos suministra todas las librerías, el almacenamiento, el poder de procesamiento y la RAM, por lo cual simplemente se ejecutan las instrucciones y se obtienen los resultados a través del navegador Web, sin consumir muchos recursos de nuestros sistemas.

Conclusiones

Las conclusiones al respecto de los modelos construidos es que tienen una grán relación entre las variables usadas para la predicción, así el modelo cumple en general con el objetivo de este trabajo, y puede ser una herramienta para la predicción de rating de una App, dada sus variables de input.

De tal manera el modelo hace posible analizar la relación del "Rating" con respecto a otras variables, y ayuda a predecir cómo se comporta la predicción frente a ciertas variables cuando las personas necesiten analizar cómo se comportaría su app en diversas situaciones estimadas o supuestas.

Como trabajo futuro, nos proponemos mejorar y probar el dataset con varios optimizadores, diferentes modelos, y probar con varias "Hidden Layers". Con el fin de mejorar las predicciones.

Referencias

[1] NewsEuropeanParliament, "What is artificial intelligence and how is it used? | News | European Parliament," News European Parliament, 2021.://www.europarl.europa.eu/news/en/headlines/society/20200827STO85804/what-is-artificial-intelligence-and-how-is-it-used (accessed Jun. 29, 2022).

[2] R. P. Diez, Introducción a la inteligencia artificial: Sistemas expertos, redes neuronales artificiales y computación evolutiva. Oviedo: Universidad de Oviedo, Servicio de Publicaciones, 2001.

[3] S. E. T. Sánchez, M. O. Rodríguez, A. E. Jiménez, and H. J. P. Soberanes, "Implementación de Algoritmos de Inteligencia Artificial para el Entrenamiento de Redes Neuronales de Segunda Generación," Jóvenes En La Cienc., vol. 2, no. 1, pp. 6–10, 2016, Accessed: Jun. 29, 2022. [Online]. Available: ://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/715.

[4] M. Gallego Gallego and J. Hernández Cáceres, "Identificación de factores que permitan potencializar el éxito de proyectos de desarrollo de software," Sci. Tech., vol. 20, no. 1, p. 70, Mar. 2015, doi: 10.22517/23447214.9241. (accessed Jun. 29, 2022).

[5] GCFGLOBAL, "¿Cómo usar Android?: Qué es y cómo usar Google Play Store," 2019. https://edu.gcfglobal.org/es/como-usar-android/que-es-y-como-usar-google-play-store/1/ (accessed Jun. 29, 2022).

[6] E. Noei and K. Lyons, "A Survey of Utilizing User-Reviews Posted on Google Play Store," 2019, doi: 10.1145/1122445.1122456.

[7] "(PDF) Capítulo 1: Generalidades de las redes neuronales artificiales." https://www.researchgate.net/publication/327703478_Capitulo_1_Generalidades_de_las_redes_neuronales_artificiales (accessed Jun. 29, 2022).

[8] E. Varela and E. Campbells, "Redes Neuronales Artificiales: Una Revisión del Estado del Arte, Aplicaciones y Tendencias Futuras," Investig. y Desarro. en TIC, vol. 2, no. 1, pp. 18–27, 2011, Accessed: Jun. 22, 2022. [Online]. Available: http://revistas.unisimon.edu.co/index.php/identic/article/view/2455.

[9] F. O.C, *Introducción a los Negocios en un Mundo Cambiante*, 4ª ed. México D.F.. México: McGraw-Hill Interamericana, 2004.

[10]"López Porrero - 2011 - Limpieza de datos reemplazo de valores ausentes y estandarización," Accessed: Aug. 19, 2022. [Online]. Available: https://1library.co/title/limpieza-datos-reemplazo-valores-ausentes-estandarizacion.

[11]"¿Qué son los Datasets? [4 sitios donde encontrarlos]". KeepCoding Tech School. https://keepcoding.io/blog/que-son-datasets/ (accedido el 19 de julio de 2022).

[12]"Google colaboratory". Google Colab. https://colab.research.google.com/?hl=es (accedido el 25 de julio de 2022).

[13]"Mapa autoorganizado". Los diccionarios y las enciclopedias sobre el Académico. https://es-academic.com/dic.nsf/eswiki/683112 (accedido el 5 de agosto de 2022).

[14] J. M. Uriarte, "Google Drive: qué es, cómo funciona y características," *2020*, 2020. https://www.caracteristicas.co/google-drive/#ixzz7c8jGC900 (accessed Aug. 19, 2022).

[15] Santander Universidades, "¿Qué es Python? | Blog Becas Santander," 2021. https://www.becas-santander.com/es/blog/python-que-es.html (accessed Aug. 19, 2022).

[16] "Todo lo que necesitas saber sobre TensorFlow, la plataforma para Inteligencia Artificial de Google – Puentes Digitales," 2021. https://puentesdigitales.com/2018/02/14/todo-lo-que-necesitas-saber-sobre-tensorflow-la-plataforma-para-inteligencia-artificial-de-google/ (accessed Aug. 19, 2022).

[17] Ionos, "¿Qué es Keras? Introducción a la biblioteca de redes neuronales - IONOS," 2020. https://www.ionos.mx/digitalguide/online-marketing/marketing-para-motores-de-busqueda/que-es-keras/ (accessed Aug. 19, 2022).

[18] F. Pérez and H. Fernández, "Las redes neuronales y la evaluación del Riesgo de Crédito," Rev. Ing. Univ. Medellín, pp. 77–91, 2007, Accessed: Aug. 19, 2022. [Online]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1692-33242007000100007

Roles de Autoría

Rudy Roberto Tito Durand: Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación, Redacción - borrador original. **Marcelo Andre Guevara Gutierrez:** Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación, Redacción - borrador original. **Jeampier Anderson Moran Fuño:** Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación, Redacción - borrador original. **Edsel Yael Alvan Ventura:** Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación, Redacción - borrador original.