

Reporte de Laboratorio Nro. 2

Jhandry Zambrano^{L00380367}

Universidad de las Fuerzas Armadas
jvzambrano3@espe.edu.ec

Tema: Limpieza de datos

Resumen

En el presente trabajo se realiza el proceso de limpieza de datos de un dataset sucio mediante Python. El objetivo principal de este trabajo es aprender a limpiar datos para procesar datos de calidad. En este proceso se empieza con la carga del dataset en consecuencia se identifica los valores nulos y se reemplaza por ceros. Luego se realiza el proceso de eliminación de datos duplicados teniendo en cuenta que por lo general en id no puede existir datos duplicados. También se realizan algunas correcciones en algunos datos específicos. En este proceso se realiza un análisis de la situación, es decir, se detecta el problema y definimos como lo detectamos. En consecuencia, al encontrar el error se debe definir el porque de la solución a aplicar. Además, debemos tener una metodología clara donde muestre el proceso que conlleva los desarrollado. Y finalmente se transforma los datos limpios de csv a Json.

1. Introducción

La limpieza de datos es esencial para el procesamiento de datos. La importancia de datos limpios radica en que para obtener información correcta y consistente es necesario aplicar limpieza de datos. La importancia de la limpieza de datos es que nos proporcionan datos mas precisos y resultados confiables. Es un hecho que hoy en día varias empresas y entidades manejan bases de datos, sin embargo, muchas de ellas poseen datos no deseados e innecesarios para el procesamiento. Estos datos pueden ser: nulos, duplicados, valores atípicos, entre otras.[3].

Para el desarrollo de esta practica se plantea un dataset sucio donde implica la limpieza de datos. Para ello se define una metodología donde abarque un proceso ETL que permita la limpieza del mismo. De la misma forma se realiza la limpieza acorde a la metodología por lo que es necesario identificar el problema, la forma como se lo detecto y porque se considera un problema en mi dataset. Para ello se debe realizar un analisis previo al cambio debido a que cada cambio que exista puede afectar a otras columnas de mi dataset.[4].

Durante la práctica ubo varios obstáculos, en otras palabras, se tuvo bastante cuidado debido a que un mal manejo ETL puede producir un mal procesamiento de datos o incongruencias. La limpieza en la duplicación debe ser congruente como en el caso de códigos únicos como un id. Limpiar datos no solamente implicar eliminar sino lograr encontrar la máxima precisión de los datos intentado perder la mínima información. Esto produce que los datos sean claros para que

en trabajos futuros puedan ser procesados y obtener excelentes resultados. Estos resultados pueden ser estadísticos, de análisis u otro tipo esto permitirá predecir tendencias. Por consiguiente, permitirá la correcta toma de decisiones de cualquier empresa.[2].

2. Método

Para el desarrollo del laboratorio de limpieza de dato primero se implementa las librerías necesarias. Pandas para el manejo de estructuras y demás librerías que vamos a ir agregando en el transcurso de la práctica. También se requiera de numpy para ejercicios matemáticos. A continuación se muestra el código del mismo.

Listing 1: Importación de librerías.

```
'''Importacion de librerias'''
import pandas as pd
import numpy as np
```

Luego se realiza la carga de la base de datos. Para este laboratorio se hace uso de una base que trata acerca de los datos de los crímenes de personas. A continuación se muestra la ubicación de la base de datos y su carga. También ejecutamos un sep para separar objetos y encoding para no tener errores en caso de tener dificultades con símbolos o la letra Ñ.

Listing 2: Carga de datos

```
'''Carga del dataset'''
df=pd.read_csv("data_act_01.csv", sep= ';', encoding='latin-1')
df
```

En este apartado imprimo el código que lo que me permite es verificar cuántos datos nulos existen. Me muestra de forma global es decir por cada una de las columnas. Esto nos permitirá definir la columna a que a ser definida para efectuar cambios posteriores.

Listing 3: Primer error

```
'''Mediante el uso del m todo isnull verificamos las columnas de nuestro dataset que tienen valores nulos'''
df.isnull().sum()
```

El primer error que identificamos es tener valores nulos. Por ende, es necesario realizar un reemplazo en las columnas que poseen datos nulos. Se toma las columnas, ciudad, rango y estado y los valores que tengan nulo serán reemplazados por cero. A continuación se observa el proceso mediante su impresión. Considero que aplico esta solución porque así me permite tener en cuenta que columnas no tienen valores debido a su reemplazo en cero.

Listing 4: Segundo Error

```
''' Reemplazo de valores por cero'''
df[['City', 'Range', 'State']] = df[['City', 'Range', 'State']].fillna(value=0)
'''Impresion de 20 datos para verificar si los cambios son efectuados'''
df.head(20)
```

En este apartado lo que realmente se realiza es el proceso donde se realiza el reemplazo de algunos datos errores de ortografía. Para ellos llamamos a la columna del llamado de donde se toma datos. y aplicamos un replace y corregimos del mal escrito a escribir de forma descabellado. A continuación se aprecia el proceso en la figura siguiente. Para ello es necesario decir que se dedujo este problema debido a que en cada df se va verificando como se van ejecutando los cambios.

Listing 5: Tercero Error

```
df.head()
'''En el segundo error de la columna crímenes se realiza reemplazos de
datos mal escritos'''
df['OriginalCrimeTypeName'].replace({"Agressive": "Aggressive"})
df['OriginalCrimeTypeName'].replace({"Cassing": "Casing"})
df['OriginalCrimeTypeName'].replace({"Drp": "Drop"})
df['OriginalCrimeTypeName'].replace({"Verbals": "Verbal"})
df['City'].replace({"S": "San_Francisco"})
df['State'].replace({"CA": "California"})
```

En este apartado lo que realmente se realiza es el proceso donde se realiza el reemplazo de algunos datos errores de ortografía. Para ellos llamamos a la columna del llamado de donde se toma datos. y aplicamos un replace y corregimos del mal escrito a escribir de forma descabellado. A continuación se aprecia el proceso en la figura siguiente. Para ello es necesario decir que se dedujo este problema debido a que en cada df se va verificando como se van ejecutando los cambios.

Listing 6: Cuarto Error

```
'''Reemplazo de los valores nulos de la columna City por NA'''
values = {'City': 'NA'}
df.fillna(value=values, inplace=True)
'''Visualizar los registros con datos nulos'''
df[df.isna().any(1)]
```

Se realiza el llamado a la columna en este caso es de la persona que cometió el crimen. Es importante tener en cuenta que existen algunos datos repetidos en id y prácticamente este no puede suceder. Por ello es necesario primeramente reconocer estos eventos que permiten modificarlos en caso de tener algún id repetido. Esto se establece mediante un for donde realizamos que el proceso sea iterativo.

Listing 7: Quinto error

```
'''Llamamos a la columna CrimeOid'''
set_duplicates = set(duplicates)

'''Llamamos a la columna CrimeOid para ello debemos tener en cuenta que
este hecho nos permitira tener id repetidos en caso de tenerlo le aumente mas 1'''
for r in set_duplicates:
    dup = df[df['CrimeId'] == r]
    df.loc[dup.iloc[1,:].name, 'CrimeId'] = df.loc[dup.iloc[1,:].name]['CrimeId'] + 1
```


Lo que se va a realizar es eliminar datos innecesarios como podemos darnos cuenta en estos datos tienen datos de fecha y hora. para ello, me veo en la necesidad de eliminar la redundancia de datos. Entonces se debe realizar el proceso que permita tener los datos limpios. Acerca del dato podemos darnos cuenta que este problema se puede observar a simple vista al mirar el dataset df. a continuación se muestra el proceso del mismo.

Listing 8: Sexto error

```
'''Llamamos a la columna CrimeOid'''
set_duplicates = set(duplicates)

'''Llamamos a la columna CrimeOid para ello debemos tener en cuenta que
este hecho nos permitira tener id repetidos en caso de tenerlo le aumente mas 1'''
for r in set_duplicates:
    dup = df[df['CrimeId'] == r]
    df.loc[dup.index[1:], 'CrimeId'] = df.loc[dup.index[1:], 'CrimeId'] + 1
```

Link público de GitHub del código utilizado para el desarrollo en este laboratorio:

 https://github.com/Jhandry1378/LAB2limpieza_datos

3. Análisis y resultado

Después de realizar el proceso de limpieza de datos puedo darme cuenta en si las diferencias que existes entre el archivo sin limpieza y el archivo limpio. Por ende, se considera que al manipular estos datos estos nos producirán datos más consistentes. Este paso del proceso de limpieza de datos es fundamental porque luego se realiza los entrenamientos con los datos o modelos de machine Learning entre otros. Que mejor que poner a trabajar datos concretos, claros y limpios que como consecuencia de igual forma voy a tener resultados increíbles y verifícos. Consecuentemente se procederá a jugar con los datos basándonos en una metodología donde primeramente busco los datos nulos, luego datos repetidos y así sucesivamente. Esto se realiza con el objetivo de realizar una limpieza que no dañe ni afecte el dataset.

4. Discusión

La limpieza de datos en si trata del proceso para evitar inconsistencias de un dataset es decir hacer que un dataset sea el mas limpio posible. La limpieza de datos es un acto que requiere de personas especializadas que puedan realizar estas actividades. Realizar por cuenta propia un dataset es realmente interesante debido a que primeramente debemos conocer el dataset es decir conocer los datos. Consecuentemente se procederá a jugar con los datos basándonos en una metodología donde primeramente busco los datos nulos, luego datos repetidos y así sucesivamente. Esto se realiza con el objetivo de realizar una limpieza que no dañe ni afecte el dataset.[1].

Limpiar este dataset fue un poco engorroso pero necesario teniendo en cuenta que teníamos un dataset de aproximadamente 10051 registros. Este modelo en si, trata de personas que han cometido crímenes. Sus características principales son el id, tipo de crimen, la hora, la fecha, entre otras características. Para el proceso de limpieza de datos es necesario tener en cuenta que primero realizamos en proceso de búsqueda de registros nulos para reemplazarlos por cero. Justo aquí pudimos darnos cuenta que existía la columna rango totalmente sin datos, entonces al no ser necesario se podría eliminar. De igual forma ir poco a poco aplicando procesos ETL que permitan tener datos limpios para su consecuente preprocesamiento.[5].

Al desarrollar los procesos de limpieza de datos pudimos darnos cuenta que no fue una tarea fácil debido a que limpiar datos tiene su complejidad. Por ende, para realizar un proceso ETL, debemos tener claro algunos otros parámetros como conocer realmente si los datos acordes a las

ciudades están en lo correcto. Tuvo algunas dificultades al momento de eliminar una columna debido a que tenía realizada algunos procesos que al antes ser eliminador cambiaba de panorama por ello se cambio el orden del proceso del mismo. Muchas de las empresas hoy en día necesitan que sus datos sean analizados para llegar a una conclusión, por ende, es necesario realizar procesos ETL. Es necesario tener en cuenta que los datos estructurados son modelos un poco mas fáciles. Sin embargo, los no estructurados son mas complejos. Esto se debe porque la información puede estar en imágenes, videos, entre otros. Aunque hubo más cosas o errores que talvez se pudo mejorar. [?].

5. Conclusión

Al culminar este trabajo puedo darme cuenta de la importancia de saber aplicar limpieza de datos a un dataset. Es decir, el hecho de limpiar un dataset en si proporciona información mas precisa para su previo analisis. Esto puede permitir, analisis, fácil implementación de modelos, ahorro en tiempo, recursos entre otros. Por ende, permite la correcta toma de decisiones de cualquier empresa teniendo en cuenta que el futuro se basa en el análisis de datos.

Se logro aplicar conceptos de programación y desarrollo del mismo. Además, es muy práctico e importante tener una metodología debido a que esto ayuda al proceso del mismo. En la metodología aplicamos los pasos para llegar al proceso de datos limpios. Es decir, aparte del desarrollo se establece un analisis identificando el problema, como lo detecte y porque considero que es un problema. Por lo que se tiene claro que la limpieza de datos no es una tarea fácil sin embargo es necesario para procesar algoritmos útiles.

Se logro establecer un proceso secuencial de pasos donde nos familiarizamos con el dataset. Luego procedemos a buscar los valores nulos en las columnas, los datos suplicados, corregir algunos errores estructurales, valores atípicos y datos faltantes. Finalmente se realizo una validación y control de calidad verificando que los datos tengan sentido. Se considera que se tuvo muchos inconvenientes con algunas funciones, sin embargo, se logró solventar debido a que todo lo detallamos de forma clara y precisa.

Referencias

- [1] Laura López. Limpieza de datos con python. cuando trabajamos con datos podemos... — by al mal tiempo, buena data — medium. <https://lauralpezb.medium.com/limpieza-de-datos-con-python-48d436ca9ace>, Junio 2021. (Accessed on 07/09/2022).
- [2] Cory Sarver. Data cleaning: The why and the how. <https://www.springboard.com/blog/data-analytics/data-cleaning/>, Agosto 2022. (Accessed on 06/08/2022).
- [3] Snehal_m. *Data cleaning|what is data cleaning|introduction to data cleaning*. <https://www.analyticsvidhya.com/blog/2021/07/data-cleaning-what-is-data-cleaning/>, Agosto 2021. (Accessed on 06/08/2022).
- [4] Tableau. Data cleaning: Definition, benefits, and how-to — tableau. <https://www.tableau.com/learn/articles/what-is-data-cleaning>, Junio 2019. (Accessed on 06/08/2022).
- [5] Michael Nguyen Tu. Cómo usar las herramientas python de limpieza de datos. <https://adamtheautomator.com/data-cleaning-python/>, Diciembre 2021. (Accessed on 08/09/2022).