Segunda entrega del proyecto

POR:

Juan Sebastián Ortiz Tangarife

MATERIA:

Introducción a la Inteligencia Artificial

PROFESOR:

Raúl Ramos Pollan

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Universidad de Antioquia

Facultad de Ingeniería

Medellín 2023

El dataset utilizado contiene 957919 filas las cuales cumplen con el requerimiento del proyecto de ser mayor a 5000. Sin embargo, en el momento de revisar los datos faltantes y las columnas categóricas no cumplían con el requerimiento ya que habían menos del 5% de datos faltantes y ninguna columna era categórica.

```
f1      15247
f2      15190
f3      15491
f4      15560
f5      15405
        ...
f114    15438
f115    15559
f116    15589
f117    15407
f118    15212
Length: 118, dtype: int64
```

Datos faltantes

```
25] ccols = [i for i in dfAux.columns if not i in dfAux._get_numeric_data()]
    print (ccols)

    []
```

Columnas categóricas

PREPROCESAMIENTO DEL DATASET

Se realizó un preprocesamiento del dataset para poder cumplir con todos los requerimientos. Lo primero que se hizo fue crear un pequeño algoritmo en el que se ingresaban las columnas a las que se querían eliminar los datos, luego, de forma aleatoria, se convertían en nan el 5% del total de filas de dichas columnas, obteniendo así el 5% de datos faltantes en las 3 columnas requeridas.

```
f1      61176
f2      61159
f3      61368
f4      15560
f5      15405
        ...
f114    15438
f115    15559
f116    15589
f117    15407
f118    15212
Length: 118, dtype: int64
```

Se puede apreciar un aumento considerable de datos faltantes en las columnas f1, f2 y f3.

Luego, para categorizar el 10% del total de columnas, que en mi caso serían 12 columnas (120*0,1=12) se procedió a realizar una función que categoriza con 1, 2 o 3 de la siguiente manera:

Si el número es menor al 33% de la longitud del intervalo en el que se mueve la variable, se categoriza como un 1; si está entre el 33,3% y el 66,6% se categoriza como un 2; y si es mayor al 66,6% se categoriza como un 3.
Por ejemplo, si tenemos una variable que toma valores entre 0 y 100, cuando tome el valor de 24, va a categorizarse como un 1.

```
def categorizarValor(valor,a,b):
  if valor<a:
    return "1"
  elif a<=valor<=b:
    return "2"
  else:
    return "3"
```

```
dfaux2=dfAux.copy()
columnasCategorizar=["f6","f10","f20","f30","f41","f51","f65","f72","f81","f90","f101","f110"]

for columna in columnasCategorizar:
  minimo=dfaux2[columna].min()
  maximo=dfaux2[columna].max()
  dfaux2[columna]=dfaux2[columna].apply(categorizarValor, args=(minimo+((maximo-minimo)*0.33),maximo-((maximo-minimo)*0.33)))
```

Una vez aplicada dicha función a las 12 columnas, se verificó que efectivamente esas 12 columnas si son ahora categóricas

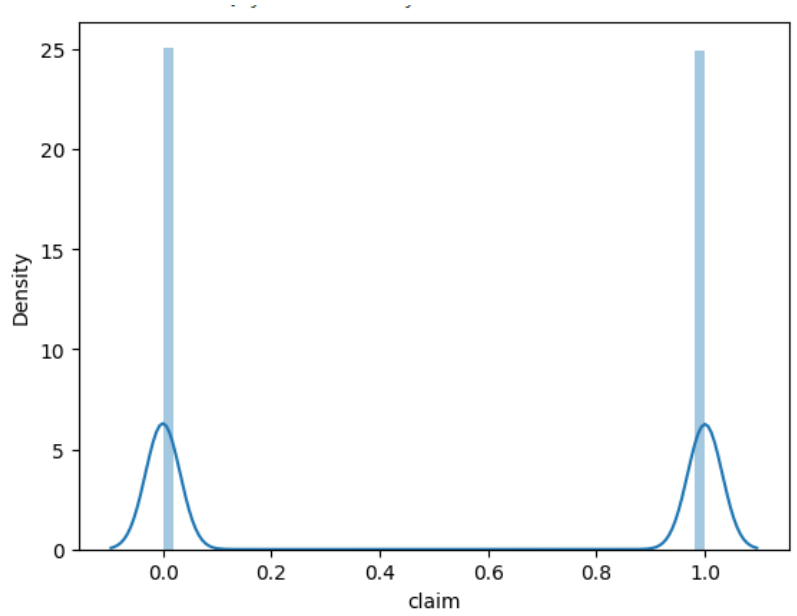
```
ccols = [i for i in dfaux2.columns if not i in dfaux2._get_numeric_data()]
print (ccols)
```

```
['f6', 'f10', 'f20', 'f30', 'f41', 'f51', 'f65', 'f72', 'f81', 'f90', 'f101', 'f110']
```

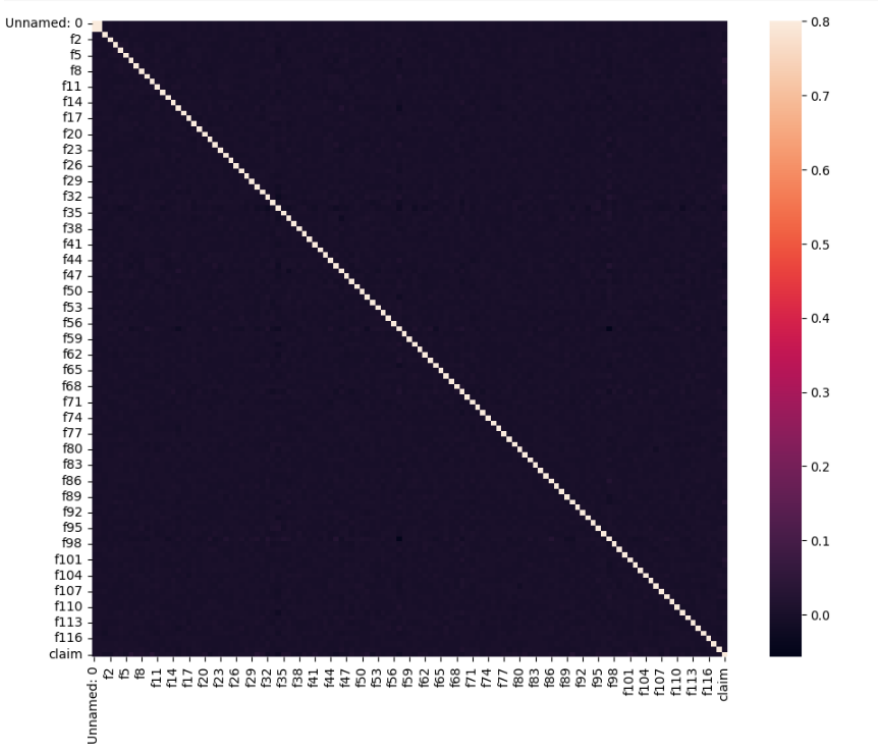
Por último se exportó el csv para poder empezar a trabajar con ese nuevo Data Frame ya preprocesado.

PEQUEÑO ANÁLISIS EXPLORATORIO DE LOS DATOS

Se inspeccionó la variable objetivo llamada Claim para ver qué valores toma y se observa que solo puede tomar dos valores: 0 y 1.



También se realizó un mapa de calor para ver la correlación entre variables o columnas y mirar si existe algún problema de correlación.



Claramente se puede observar que la correlación entre columnas es menor al 10% lo cual es bueno ya que no vamos a tener información redundante que nos pueda llegar a afectar el futuro modelo que creemos.