Tratado de valores faltantes

Minería de Datos
Zavala Roman Irvin Eduardo 1270771

Dataset “San Francisco Building Permits”

Un permiso de construcción es un documento oficial para dar paso a la construcción o remodelación de una propiedad.

```
dataset = pd.read_csv("Building_Permits.csv", on_bad_lines='skip')
print("\tINFORMACION DEL DATASET\n_____________________________________________)"
dataset.info()
```

INFORMACION DEL DATASET

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 198900 entries, 0 to 198899
Data columns (total 43 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 7   Street Number Suffix              2216 non-null    object
 9   Street Suffix                   196132 non-null  object
 10  Unit                             29479 non-null   float64
 11  Unit Suffix                       1961 non-null    object
 12  Description                     198610 non-null  object
 16  Issued Date                     183960 non-null  object
                                 …
 39  Neighborhoods - Analysis Boundaries     197175 non-null  object
 40  Zipcode                         197184 non-null  float64
 41  Location                        197200 non-null  object
dtypes: float64(12), int64(3), object(28)
memory usage: 65.3+ MB
```

Si hay muchos datos faltantes

Dataset “San Francisco Building Permits”

```
print("\tCANTIDAD DE DATOS FALTANTES EN DATASET (TOTAL ROWS X
COLS)\n_____________________________________________________________________________")

print("\t% of missing data: ", dataset.isna().sum().sum() / (dataset.size) *100)
```

```
    CANTIDAD DE DATOS FALTANTES EN DATASET (TOTAL ROWS X COLS)

_____________________________________________________________________________
    % of missing data:  26.260023150584O3
```

Dataset “San Francisco Building Permits”: Eliminar NaN’s

```
#Metodo 1: Eliminar registros con valores faltantes
dataset_ignored_NAN = dataset.copy()
#Si se hace dropna() se elimina todo el dataset ya que en filas y columnas faltan
datos
dataset_ignored_NAN.dropna(inplace=True)#Se puede usar axis = ‘columns’ y se
eliminan las columnas con datos faltantes
dataset_ignored_NAN.info()
#Metodo menos recomendado porque como faltan muchos datos al querer ignorarlos se
ignora todo el dataset :o
```

```
Int64Index: 0 entries
Data columns (total 43 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
```

Dataset “San Francisco Building Permits”: Asignar constante

```
#Metodo 2: Reemplazar NANs con una constante (No recomendado porque no hay supervicion de un experto)
dataset_replace_constant = dataset.copy()
dataset_replace_constant.fillna(0, inplace=True) #Se reemplaza con 0
dataset_replace_constant.info() #Se puede ver que ahora no faltan datos, pero a que costo :'(
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 198900 entries, 0 to 198899
Data columns (total 43 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   Permit Number                         198900 non-null  object
 1   Permit Type                           198900 non-null  int64
 2   Permit Type Definition                198900 non-null  object
 3   Permit Creation Date                  198900 non-null  object
 39  Neighborhoods - Analysis Boundaries   198900 non-null  object
 40  Zipcode                               198900 non-null  float64
 41  Location                              198900 non-null  object
 42  Record ID                             198900 non-null  int64
dtypes: float64(12), int64(3), object(28)
memory usage: 65.3+ MB
```

Dataset “San Francisco Building Permits”: Medida de tendencia central por atributo

```
#Metodo 3: Reemplazar NANs con una medida de tendencia central
def reemplazarPorMTC(modo, df):
    df = df.select_dtypes(['number'])
    for i in df.columns:
        try:
            if(df[i].isnull().values.any()):
                if(modo == 0): #Media
                    df[i].fillna(df[i].mean(), inplace=True)
                    print(i, df[i].mean())
                if(modo == 1): #Mediana
                    df[i].fillna(df[i].median(), inplace=True)
                    print(i, df[i].median())
        except:
            continue
    return df
```

Dataset “San Francisco Building Permits”: Medida de tendencia central por atributo

```
                    Media
Unit 78.51718172258218
Number of Existing Stories 5.705773271157344
Number of Proposed Stories 5.745042683552092
Estimated Cost 168955.44329681533
Revised Cost 132856.18649174884
Existing Units 15.666164275729155
Proposed Units 16.510950138185947
Plansets 1.2746501971025614
Existing Construction Type 4.072877955945324
Proposed Construction Type 4.0895285672090305
Supervisor District 5.538403412058849
Zipcode 94115.5005578546
                   Mediana
Unit 0.0
Number of Existing Stories 3.0
Number of Proposed Stories 3.0
Estimated Cost 11000.0
Revised Cost 7000.0
Existing Units 1.0
Proposed Units 2.0
Plansets 2.0
Existing Construction Type 5.0
Proposed Construction Type 5.0
Supervisor District 6.0
Zipcode 94114.0
```

Dataset “Singapore Airbnb”

```
#DATASET AIRBNB
dataset = pd.read_csv("listings.csv")
print("\tINFORMACION DEL DATASET\n_____________________________________________" )
dataset.info()
```

```
    INFORMACION DEL DATASET
_____________________________________
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7907 entries, 0 to 7906
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              7907 non-null   int64
 1   name                            7905 non-null   object
 2   host_id                         7907 non-null   int64
 3   host_name                       7907 non-null   object
 4   neighbourhood_group             7907 non-null   object
 5   neighbourhood                   7907 non-null   object
 6   latitude                        7907 non-null   float64
 7   longitude                       7907 non-null   float64
 8   room_type                       7907 non-null   object
 9   price                           7907 non-null   int64
 10  minimum_nights                  7907 non-null   int64
 11  number_of_reviews               7907 non-null   int64
 12  last_review                     5149 non-null   object
 13  reviews_per_month               5149 non-null   float64
 14  calculated_host_listings_count  7907 non-null   int64
 15  availability_365                7907 non-null   int64
dtypes: float64(3), int64(7), object(6)
memory usage: 988.5+ KB
```

Dataset “Singapore Airbnb”

```
print("\tCANTIDAD DE DATOS FALTANTES EN DATASET (TOTAL ROWS X
COLS)\n_____________________________________________________________________________" )

print("\t% of missing data: ", dataset.isna().sum().sum() / (dataset.size) * 100)
```

```
    CANTIDAD DE DATOS FALTANTES EN DATASET (TOTAL ROWS X COLS)
_____________________________________________________________________________
    % of missing data:  4.361641583407107
```

Dataset “Singapore Airbnb”: Eliminar NaN’s

```
#Metodo 1: Eliminar registros con valores faltantes
dataset_ignored_NAN = dataset.copy()
dataset_ignored_NAN.dropna(axis =  'index',inplace= True)
dataset_ignored_NAN.info()
```

De 7907 a 5148 datos

```
#   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
0   id                              5148 non-null   int64
1   name                            5148 non-null   object
2   host_id                         5148 non-null   int64
3   host_name                       5148 non-null   object
4   neighbourhood_group             5148 non-null   object
5   neighbourhood                   5148 non-null   object
6   latitude                        5148 non-null   float64
7   longitude                       5148 non-null   float64
8   room_type                       5148 non-null   object
9   price                           5148 non-null   int64
10  minimum_nights                  5148 non-null   int64
11  number_of_reviews               5148 non-null   int64
12  last_review                     5148 non-null   object
13  reviews_per_month               5148 non-null   float64
14  calculated_host_listings_count  5148 non-null   int64
15  availability_365                5148 non-null   int64
```

Dataset “Singapore Airbnb”: Asignar constante

```
#Metodo 2: Reemplazar NANs con una constante (No recomendado porque no hay supervicion de un experto)
dataset_replace_constant = dataset.copy()
dataset_replace_constant.fillna( 0, inplace=True) #Se reemplaza con 0
dataset_replace_constant.info()  #Se puede ver que ahora no faltan datos, pero a que costo :'(
```

```
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              7907 non-null   int64  
 1   name                            7907 non-null   object 
 2   host_id                         7907 non-null   int64  
 3   host_name                       7907 non-null   object 
 4   neighbourhood_group             7907 non-null   object 
 5   neighbourhood                   7907 non-null   object 
 6   latitude                        7907 non-null   float64
 7   longitude                       7907 non-null   float64
 8   room_type                       7907 non-null   object 
 9   price                           7907 non-null   int64  
 10  minimum_nights                  7907 non-null   int64  
 11  number_of_reviews               7907 non-null   int64  
 12  last_review                     7907 non-null   object 
 13  reviews_per_month               7907 non-null   float64
 14  calculated_host_listings_count  7907 non-null   int64  
 15  availability_365                7907 non-null   int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 988.5+ KB
```

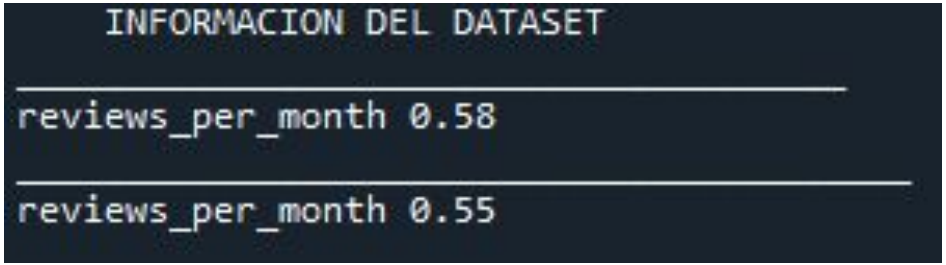
Dataset “Singapore Airbnb”: Medida de tendencia central por atributo/clase

```
dataset = pd.read_csv("listings.csv")
for i in dataset.columns:
    try:
        if(dataset.isna()[i].sum().sum() == 0):
            continue
        dataset[i].fillna(dataset.groupby("neighbourhood_group")[i].transform("median"), inplace=True)
    except:
        continue
    print(i, dataset[i].median())
print("_____________________________________________")
dataset = pd.read_csv("listings.csv")
for i in dataset.columns:
    try:
        if(dataset.isna()[i].sum().sum() == 0):
            continue
        dataset[i].fillna(dataset[i].median(), inplace=True)
    except:
        continue
    print(i, dataset[i].median())
```

Dataset “Singapore Airbnb”: Medida de tendencia central por atributo/clase

Por clase

Por atributo



Dataset “Human activity”

Consta de una gran cantidad de dispositivos tienen actividades físicas registradas por estos.

```
dataset = pd.read_csv("Watch_accelerometer.csv",index_col=False)
print("\tINFORMACION DEL
DATASET\n_____________________________________")
print(dataset.info())
print(dataset.isna().sum())
```

```
Index                0
Arrival_Time         0
Creation_Time        0
x                    0
y                    0
z                    0
User                 0
Model                0
Device               0
gt              520357
```

```
    INFORMACION DEL DATASET
_____________________________________
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3540962 entries, 0 to 3540961
Data columns (total 10 columns):
 #   Column         Dtype
---  ------         -----
 0   Index          int64
 1   Arrival_Time   int64
 2   Creation_Time  int64
 3   x              float64
 4   y              float64
 5   z              float64
 6   User           object
 7   Model          object
 8   Device         object
 9   gt             object
```

Dataset “Human activity”: Eliminar NaN’s

```
#Metodo 1: Eliminar registros con valores faltantes
dataset_ignored_NAN = dataset.copy()
dataset_ignored_NAN.dropna(inplace=  True)
print(dataset_ignored_NAN.isna(). sum())
print(dataset_ignored_NAN.shape)
```

```
Index           0
Arrival_Time    0
Creation_Time   0
x               0
y               0
z               0
User            0
Model           0
Device          0
gt              0
dtype: int64
(3020605, 10)
```

De 3540962 a 3020605 datos

Dataset “Human activity”: Asignar constante

```
#Metodo 2: Reemplazar NANs con una constante
dataset_replace_constant = dataset.copy()
dataset_replace_constant.fillna( 'No identificado', inplace=True)
print(dataset_replace_constant.isna(). sum())
print(dataset_replace_constant.shape)
```

```
Index           0
Arrival_Time    0
Creation_Time   0
x               0
y               0
z               0
User            0
Model           0
Device          0
gt              0
dtype: int64
(3540962, 10)
```

Dataset “Human activity”: Medida de tendencia central por clase

Como el valor que falta es tipo objeto (str), solo se puede usar la moda. Para que no sea la misma moda para todo el atributo, se cambiaron los NaNs por la moda de cada Usuario.

```
#Metodo 3: Reemplazar NANs con una medida de tendencia central
dataset_replace_mode = dataset.copy()
for i in dataset_replace_mode.columns:
    if(dataset_replace_mode[i].isnull().values.any()):
        dataset_replace_mode[i].fillna(dataset_replace_mode.groupby("User")[i].transform(lambda x:
x.value_counts().idxmax()), inplace=True) #Se reemplaza con la moda
print(dataset_replace_mode.isna().sum())
print(dataset_replace_mode.shape)
```

Dataset “Human activity”: Medida de tendencia central por atributo

```
Index           0
Arrival_Time    0
Creation_Time   0
x               0
y               0
z               0
User            0
Model           0
Device          0
gt              0
dtype: int64
(3540962, 10)
```