Título del trabajo: Reporte

Alumno: Isaac Sánchez Flores

Fecha de entrega:

Profesor: Jaime Alejandro Romero

Sierra

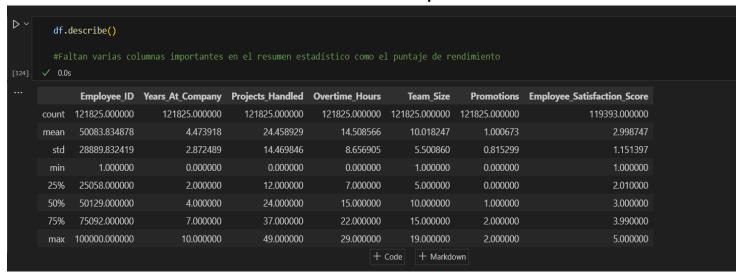
Materia: Introducción a la ciencia

de datos

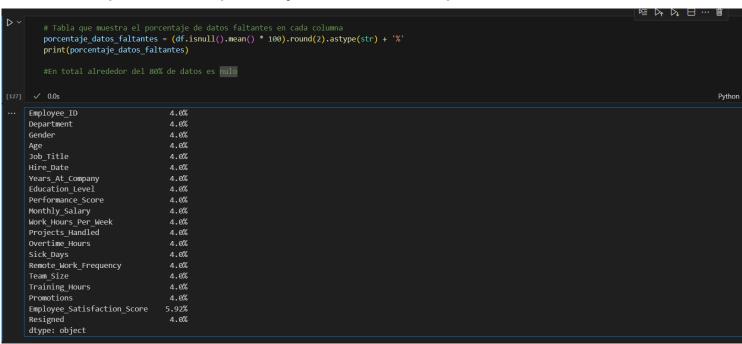


## Análisis inicial

## Resumen estadístico de los datos antes de la limpieza:



# Tabla que muestra el porcentaje de datos faltantes por columna:



# Total de datos repetidos encontrados: (Solo en Employee\_ID ya que en los démas es irrelevante)

```
df.duplicated('Employee_ID').sum() #Comprobamos cuantos datos duplicados hay en Employee_ID

#Tenemos 30032 duplicados

✓ 0.0s

30032
```

## Descripción de los tipos de datos originales y los problemas encontrados:

Los datos originalmente eran de tipo numérico, en su mayoría, pero ahora todos los datos estaban en formato object, esto provoca que datos importantes como el Perfomance Score no aparezcan en el resumen estadístico.

Otro problema que encontré a la hora de hacer el análisis inicial de la base de datos fue que gran parte de las columnas contenían valores inválidos tales como "bbb"

```
lista_columnas = df.columns
                                                        (function) def __eq__(other: object) -> Series[_bool]
   for c in lista_columnas:
       print(f'En la columna {c} los bbb son: {df[df[c] == 'bbb'].shape[0]}')
   #Revisamos cuantos bbb hay en cada columna
 ✓ 0.0s
En la columna Employee_ID los bbb son: 0
En la columna Department los bbb son: 2430
En la columna Gender los bbb son: 0
En la columna Age los bbb son: 2436
En la columna Job Title los bbb son: 0
En la columna Hire_Date los bbb son: 0
En la columna Years_At_Company los bbb son: 0
En la columna Education Level los bbb son: 2445
En la columna Performance Score los bbb son: 2434
En la columna Monthly_Salary los bbb son: 2439
En la columna Work_Hours_Per_Week los bbb son: 2425
En la columna Projects_Handled los bbb son: 0
En la columna Overtime_Hours los bbb son: 0
En la columna Sick_Days los bbb son: 2434
En la columna Remote_Work_Frequency los bbb son: 2441
En la columna Team_Size los bbb son: 0
En la columna Training_Hours los bbb son: 2445
En la columna Promotions los bbb son: 0
En la columna Employee_Satisfaction_Score los bbb son: 0
En la columna Resigned los bbb son: 0
```

# Proceso de limpieza

Para la limpieza de la base de datos además de la librería pandas utilicé la librería de numpy por razones que explicaré a continuación. Las funciones de pandas que utilicé fueron las de eliminación de duplicados y valores NaN, conversión de datos a un formato en específico y llenado de valores NaN ya sea con una leyenda que diga 'Sin Datos' o con el promedio de cada columna. La única función que utilicé de numpy fue la que permite convertir datos a NaN, esto lo hice para sustituir la cadena 'bbb' por valores que me fueran más convenientes con ayuda de la función de pandas mencionada anteriormente.

## Antes de eliminar duplicados:

```
df.duplicated('Employee_ID').sum() #Comprobamos cuantos datos duplicados hay en Employee_ID

#Tenemos 30032 duplicados

v 0.0s

30032
```

## Después de eliminar duplicados:

```
#Primero eliminamos los duplicados en Employee_ID

df = df.drop_duplicates(subset=['Employee_ID'])

df = df.reset_index(drop=True)

df.duplicated(subset=['Employee_ID']).sum()

$\square 0.0s$

Python
```

## Antes de transformar los datos a numérico:

```
df.info()
                    #La mayoría de datos está en formato object, tengo que cambiar a float o int según sea conveniente
[150] V 0.0s
                                                                                                                                                                                                                                                                                                                                                                                                                         Python
            <class 'pandas.core.frame.DataFrame'>
            RangeIndex: 126901 entries, 0 to 126900
            Data columns (total 20 columns):
                                                                                                Non-Null Count Dtype
             # Column

        0
        Employee_ID
        121825 non-null float64

        1
        Department
        121825 non-null object

        2
        Gender
        121825 non-null object

        3
        Age
        121825 non-null object

        4
        Job_Title
        121825 non-null object

        5
        Hire_Date
        121825 non-null object

        6
        Years_At_Company
        121825 non-null float64

        7
        Education_Level
        121825 non-null object

        8
        Performance_Score
        121825 non-null object

        9
        Monthly_Salary
        121825 non-null object

        10
        Work_Hours_Per_Week
        121825 non-null float64

        12
        Overtime_Hours
        121825 non-null float64

        13
        Sick_Days
        121825 non-null object

        14
        Remote_Work_Frequency
        121825 non-null object

              14 Remote_Work_Frequency 121825 non-null object
15 Team_Size 121825 non-null float64
                                                                                           121825 non-null object
              16 Training_Hours
              17 Promotions
                                                                                                 121825 non-null float64
              18 Employee_Satisfaction_Score 119393 non-null float64
              19 Resigned
                                                                                                121825 non-null object
            dtypes: float64(7), object(13)
            memory usage: 19.4+ MB
```

## Después de transformar los datos a numérico:



## Antes de convertir 'bbb' a NaN:

D ~	df									
[148]	✓ 0.0s									
		Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performan
	0	1.0	IT	Male	55	Specialist	NaN	2.0	High School	
	1	2.0	Finance	Male	29	Developer	2024-04-18 08:03:05.556036	0.0	High School	
	2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	
	3	4.0	bbb	Female	48	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	
	4	5.0	Engineering	Female	36	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	
	126896	9646.0	Finance	Male	38	Engineer	2021-06-07 08:03:05.556036	3.0	Bachelor	
							2023-09-05			

# Después de convertir 'bbb' a NaN:

	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Sala
0	1.0	ІТ	Male	55.0	Specialist	NaN	2.0	High School	5.0	6750
1	2.0	Finance	Male	29.0	Developer	2024-04-18 08:03:05.556036	0.0	High School	5.0	7500
2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	3.0	Na
3	4.0	NaN	Female	48.0	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	2.0	4800
4	5.0	Engineering	Female	36.0	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	2.0	4800
96864	72705.0	Engineering	Female	27.0	Specialist	2019-07-23 08:03:05.556036	NaN	Bachelor	NaN	5400
96865	3248.0	Operations	Male	35.0	Analyst	2017-08-29	NaN	Bachelor	2.0	4800

# Antes de remplazar NaN por 'Sin Dato' o el promedio:

#Convertimos los bbb a NaN utilizando la librería de Numpy para luego llenar con el promedio o 'Sin dato', según sea conveniente
df.replace('bbb' , np.nan, inplace=True)
df

d<del>1</del> 159] ✓ (

	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Sala
0	1.0	ІТ	Male	55.0	Specialist	NaN	2.0	High School	5.0	6750
1	2.0	Finance	Male	29.0	Developer	2024-04-18 08:03:05.556036	0.0	High School	5.0	7500
2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	3.0	Ná
3	4.0	NaN	Female	48.0	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	2.0	4800
4	5.0	Engineering	Female	36.0	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	2.0	4800
96864	72705.0	Engineering	Female	27.0	Specialist	2019-07-23 08:03:05.556036	NaN	Bachelor	NaN	540(
96865	3248.0	Operations	Male	35.0	Analyst	2017-08-29 08:03:05.556036	NaN	Bachelor	2.0	4800

## Después de reemplazar 'bbb' por 'Sin Dato' o el promedio:

		Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Salary
	0	1.0	IT	Male	55.000000	Specialist	NaN	2.000000	High School	5.000000	6750.000000
	1	2.0	Finance	Male	29.000000	Developer	2024-04-18 08:03:05.556036	0.000000	High School	5.000000	7500.000000
	2	3.0	Finance	Male	41.024325	Specialist	2015-10-26 08:03:05.556036	8.000000	High School	3.000000	6404.100908
	3	4.0	Sin dato	Female	48.000000	Analyst	2016-10-22 08:03:05.556036	7.000000	Bachelor	2.000000	4800.000000
	4	5.0	Engineering	Female	36.000000	Sin dato	2021-07-23 08:03:05.556036	3.000000	Bachelor	2.000000	4800.000000
	96864	72705.0	Engineering	Female	27.000000	Specialist	2019-07-23 08:03:05.556036	4.478632	Bachelor	2.996184	5400.000000
	96865	3248.0	Operations	Male	35.000000	Analyst	2017-08-29 08:03:05.556036	4.478632	Bachelor	2.000000	4800.000000
	96866	66901.0	Sales	Male	41.024325	Specialist	2015-02-01 08:03:05.556036	9.000000	High School	4.000000	6300.000000
	96867	21770.0	Marketing	Male	33.000000	Technician	2016-10-22 08:03:05.556036	7.000000	Bachelor	3.000000	4550.000000
	96868	33177.0	Operations	Other	39.000000	Specialist	2020-12-24 08:03:05.556036	3.000000	Bachelor	3.000000	5850.000000

# Antes de aplicar dropna en Employee\_ID y Hire\_Date:

```
> ×
               df.info()
        <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 96869 entries, 0 to 96868
         Data columns (total 20 columns):
          # Column
                                                                        Non-Null Count Dtype
                                                    96869 non-null object
96869 non-null object
96869 non-null object
          0 Employee ID
          1 Department
         1 Department 96869 non-null object
2 Gender 96869 non-null object
3 Age 96869 non-null float64
4 Job_Title 96869 non-null object
5 Hire_Date 92983 non-null object
6 Years_At_Company 96869 non-null float64
7 Education_Level 96869 non-null object
8 Performance_Score 96869 non-null float64
9 Monthly_Salary 96869 non-null float64
10 Work_Hours_Per_Week 96869 non-null float64
11 Projects_Handled 96869 non-null float64
12 Overtime_Hours 96869 non-null float64
13 Sick_Days 96869 non-null float64
14 Remote Work_Frequency 96869 non-null float64
          14 Remote_Work_Frequency 96869 non-null float64
15 Team_Size 96869 non-null float64
                 Training_Hours
                                                                        96869 non-null float64
           17 Promotions
                                                                        96869 non-null float64
           18 Employee_Satisfaction_Score 96869 non-null float64
                                                                         96869 non-null object
           19 Resigned
         dtypes: float64(13), object(7)
         memory usage: 14.8+ MB
```

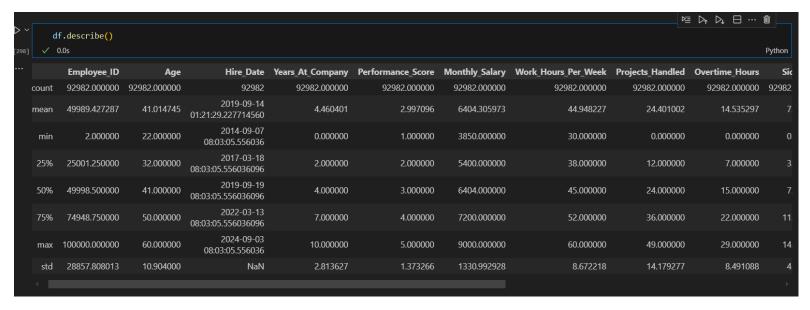
## Después de aplicar dropna en Employee\_ID y Hire\_Date:

# Antes de convertir todos los datos a su respectivo formato:

## Después de convertir todos los datos a su respectivo formato:

## Resultados:

#### Resumen final



### Tabla de datos faltantes final

```
porcentaje_datos_faltantes_final = (df.isnull().mean() * 100).round(2).astype(str) + '%'
   print(porcentaje_datos_faltantes_final)
Employee_ID
                               0.0%
Department
                               0.0%
Gender
                                0.0%
                                0.0%
Age
Job_Title
                                0.0%
Hire_Date
                                0.0%
Years_At_Company
                                0.0%
Education_Level
                                0.0%
Performance_Score
                                0.0%
Monthly_Salary
                                0.0%
Work_Hours_Per_Week
                                0.0%
Projects_Handled
                                0.0%
Overtime_Hours
                                0.0%
Sick Days
                                0.0%
Remote Work Frequency
                                0.0%
Team Size
                                0.0%
Training Hours
                                0.0%
Promotions
                                0.0%
Employee_Satisfaction_Score
                                0.0%
Resigned
                                0.0%
dtype: object
```

# Comprobación de que no hay duplicados ni valores inválidos

```
for c in lista_columnas:
       print(f'En la columna {c} los bbb son: {df[df[c] == 'bbb'].shape[0]}')
En la columna Employee_ID los bbb son: 0
En la columna Department los bbb son: 0
En la columna Gender los bbb son: 0
En la columna Age los bbb son: 0
En la columna Job_Title los bbb son: 0
En la columna Hire_Date los bbb son: 0
En la columna Years_At_Company los bbb son: 0
En la columna Education_Level los bbb son: 0
En la columna Performance Score los bbb son: 0
En la columna Monthly_Salary los bbb son: 0
En la columna Work_Hours_Per_Week los bbb son: 0
En la columna Projects_Handled los bbb son: 0 \,
En la columna Overtime_Hours los bbb son: 0
En la columna Sick_Days los bbb son: 0
En la columna Remote_Work_Frequency los bbb son: 0
En la columna Team_Size los bbb son: 0
En la columna Training_Hours los bbb son: 0
En la columna Promotions los bbb son: 0 \,
En la columna Employee_Satisfaction_Score los bbb son: 0
En la columna Resigned los bbb son: 0
                                                                                                                                            df.duplicated('Employee_ID').sum() #Comprobamos que no hay duplicados
```