PROYECTO 1 – LIMPIEZA DE DATOS

Alumna: Dana Domínguez Arce

Materia: Introducción a la Ciencia de Datos

Nombre del profesor: Jaime Alejandro Romero Sierra

Fecha de entrega: 20 de octubre del 2025

Link al repositorio de GitHub:

https://github.com/danadom06-jpg/Proyecto-1---Limpieza-de-datos.git

2. Descripción inicial de la base de datos

Fuente o contexto de la base de datos

La base de datos proviene de un conjunto sintético de información sobre salarios y habilidades laborales en inteligencia artificial y ciencia de datos durante el periodo 2024-2025. Fue creada con el propósito de analizar tendencias globales en empleos tecnológicos, incluyendo puestos, habilidades requeridas, salarios y localización geográfica. En este trabajo, la base se utilizó para practicar la limpieza y el manejo de datos, simulando un escenario real en el que la información presenta errores o valores faltantes. El dataframe originalmente estaba limpio, sin embargo, fue ensuciada por un programa.

Descripción general del contenido

El conjunto de datos incluía columnas con información sobre el puesto de trabajo, país, salario anual, experiencia requerida, habilidades principales y nivel educativo. Sin embargo, presentaba inconsistencias como valores nulos, datos duplicados y errores en el formato de algunas columnas numéricas (por ejemplo, salarios en texto).

Significado de cada columna

Nombre de la columna	Descripción	Valores de ejemplo
id_del_trabajo	Identificador único para cada oferta de empleo	AI00001, AI00002
título profesional	Título/función anunciado	Científico investigador de IA, analista de datos
salario_usd	Salario anual convertido a USD	90376, 124355
moneda_salaria	Moneda en la que se ofrece el salario	USD, EUR, GBP
nivel de experiencia	Nivel de experiencia requerido (EN=Entrante, MI=Medio, SE=Senior, EX=Ejecutivo)	SE, EN, MI, EX

tipo_de_empleo	Tipo de contrato (CT=Contrato, FT=Tiempo completo, PT=Tiempo parcial, FL=Freelance)	Tiempo completo, tiempo parcial, tiempo ...
ubicación de la empresa	País donde tiene su sede la empresa	China, Canadá, Alemania
tamaño de la empresa	Tamaño de la empresa (S=Pequeña, M=Mediana, L=Grande)	Tallas pequeñas, medianas y grandes
residencia del empleado	País de residencia del empleado	China, Irlanda, Singapur
relación remota	% de trabajo remoto permitido (0=Presencial, 50=Híbrido, 100=Remoto)	0, 50, 100
habilidades requeridas	Habilidades técnicas clave requeridas (separadas por comas)	Python, SQL, Tableau, Docker
educación_requerida	Nivel mínimo de educación requerido	Licenciatura, Maestría, Doctorado, Asociado
años_de_experiencia	Se requieren años mínimos de experiencia	0, 4, 9, 15
industria	Sector industrial del puesto de trabajo	Automoción, medios de comunicación, salud
fecha de publicación	Fecha en que se publicó el trabajo (AAAA-MM-DD)	18 de octubre de 2024
fecha límite de solicitud	Última fecha para postular al empleo (AAAA-MM-DD)	7 de noviembre de 2024
longitud de la descripción del trabajo	Longitud de la descripción del puesto en caracteres	1076, 2340
puntuación de beneficios	Puntuación numérica (0-10) que refleja la calidad de los beneficios	5.9, 9.4

nombre de empresa	Nombre de la empresa contratante	Análisis inteligente, TechCorp Inc

información especifica:

Nivel de experiencia: EN (Entrada), MI (Medio), SE (Senior), EX (Ejecutivo).

Tipo de empleo: FT (tiempo completo), PT (tiempo parcial), CT (contrato), FL (autónomo).

Tamaño de la empresa: S (Pequeña), M (Mediana), L (Grande).

Relación remota: 0 (en sitio), 50 (híbrido), 100 (remoto).

CAPTURAS DE PANTALLA:

En ellas se observa:

• Revisión de datos faltantes (isnull(), info(), etc.)

• Detección y manejo de duplicados

• Corrección de valores atípicos o inconsistentes

• Detección y corrección de palabras extrañas o mal escritas

• Traducción de textos al español (cuando aplique)

• Cambio de nombres de columnas para mayor claridad o consistencia

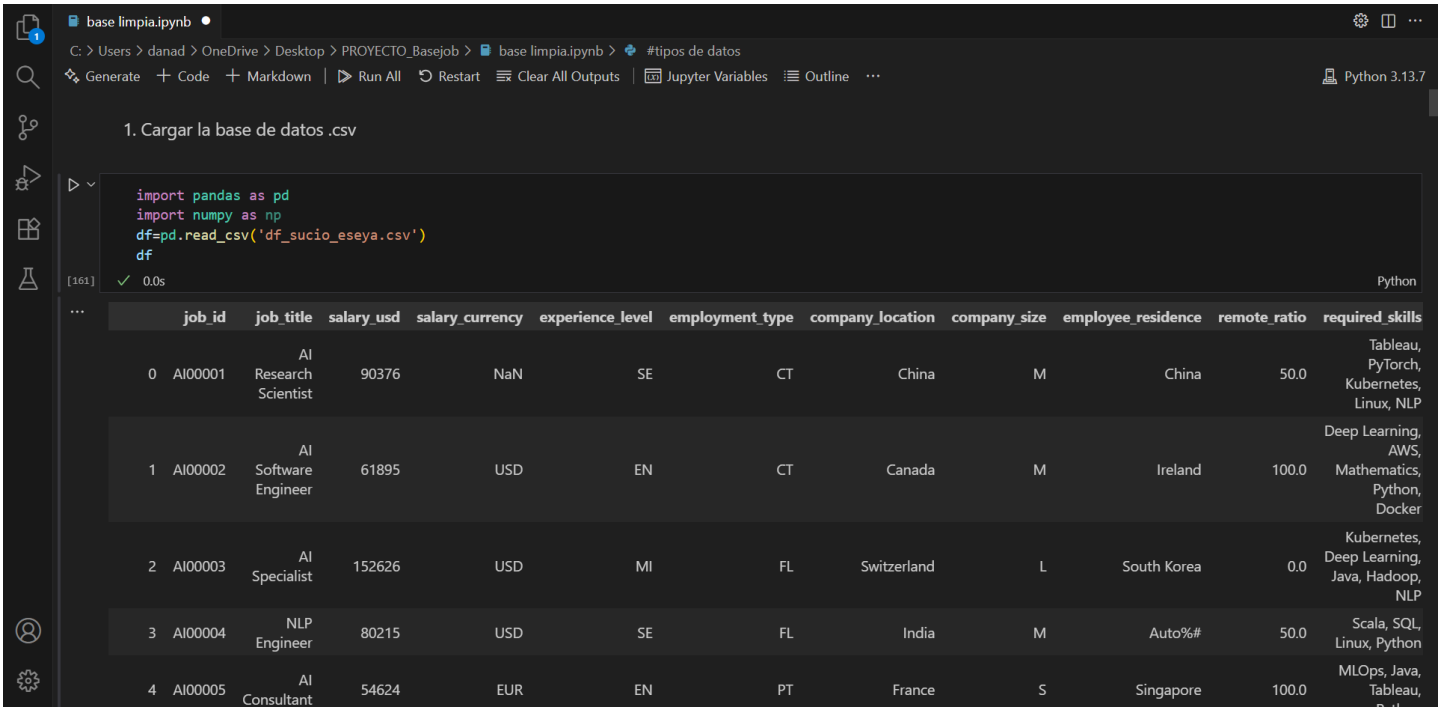
• Conversión de tipos de datos (numéricos, fechas, etc.)

• Validación final mostrando que la base quedó limpia y coherente

Cada paso debe estar claramente documentado con explicación, código y resultado.

3. Proceso de Limpieza (con evidencias)

Incluye capturas de pantalla y fragmentos de código donde muestres los pasos realizados:



base limpia.ipynb

C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > #tipos de datos

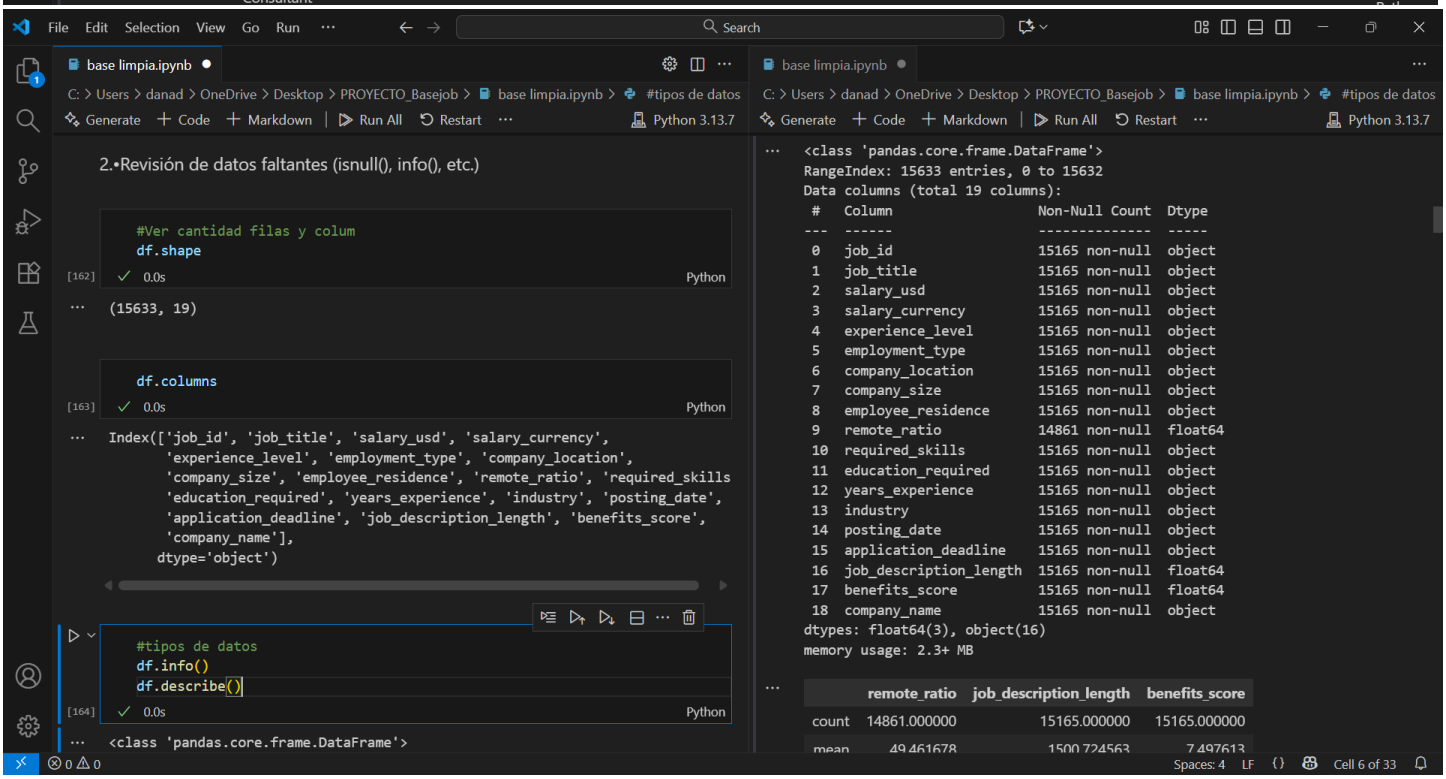
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ... Python 3.13.7

1. Cargar la base de datos .csv

```
import pandas as pd
import numpy as np
df=pd.read_csv('df_sucio_eseya.csv')
df
```

[161] ✓ 0.0s Python

	job_id	job_title	salary_usd	salary_currency	experience_level	employment_type	company_location	company_size	employee_residence	remote_ratio	required_skills
0	AI00001	AI Research Scientist	90376	NaN	SE	CT	China	M	China	50.0	Tableau, PyTorch, Kubernetes, Linux, NLP
1	AI00002	AI Software Engineer	61895	USD	EN	CT	Canada	M	Ireland	100.0	Deep Learning, AWS, Mathematics, Python, Docker
2	AI00003	AI Specialist	152626	USD	MI	FL	Switzerland	L	South Korea	0.0	Kubernetes, Deep Learning, Java, Hadoop, NLP
3	AI00004	NLP Engineer	80215	USD	SE	FL	India	M	Auto%#	50.0	Scala, SQL, Linux, Python
4	AI00005	AI Consultant	54624	EUR	EN	PT	France	S	Singapore	100.0	MLOps, Java, Tableau,



File Edit Selection View Go Run ... Search

base limpia.ipynb

C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > #tipos de datos

Generate + Code + Markdown | Run All Restart ... Python 3.13.7

2.•Revisión de datos faltantes (isnull(), info(), etc.)

```
#Ver cantidad filas y colum
df.shape
```

[162] ✓ 0.0s Python

```
(15633, 19)
```

```
df.columns
```

[163] ✓ 0.0s Python

```
Index(['job_id', 'job_title', 'salary_usd', 'salary_currency',
       'experience_level', 'employment_type', 'company_location',
       'company_size', 'employee_residence', 'remote_ratio', 'required_skills
       'education_required', 'years_experience', 'industry', 'posting_date',
       'application_deadline', 'job_description_length', 'benefits_score',
       'company_name'],
      dtype='object')
```

```
#tipos de datos
df.info()
df.describe()
```

[164] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
```

base limpia.ipynb

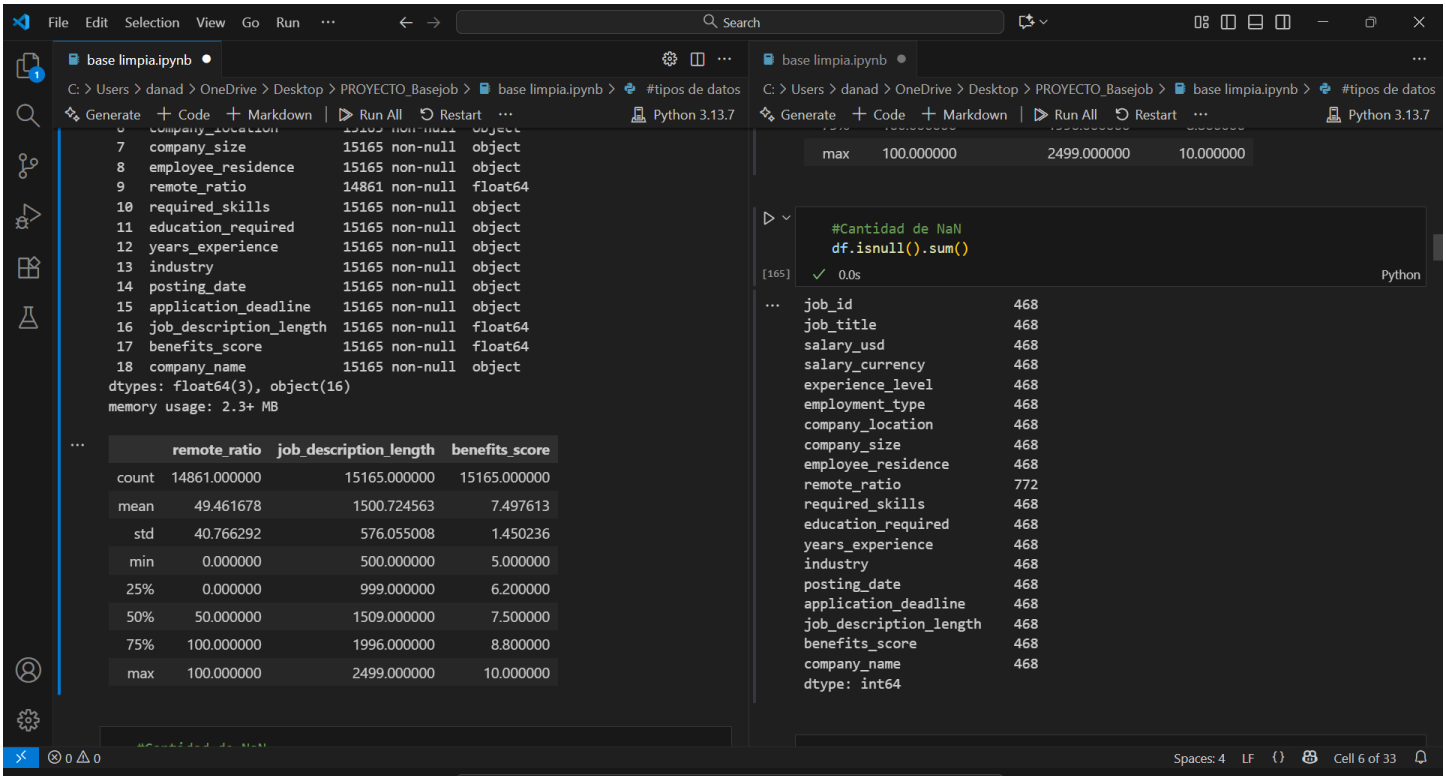
C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > #tipos de datos

Generate + Code + Markdown | Run All Restart ... Python 3.13.7

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15633 entries, 0 to 15632
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   job_id                15165 non-null  object
 1   job_title             15165 non-null  object
 2   salary_usd            15165 non-null  object
 3   salary_currency       15165 non-null  object
 4   experience_level      15165 non-null  object
 5   employment_type       15165 non-null  object
 6   company_location      15165 non-null  object
 7   company_size          15165 non-null  object
 8   employee_residence    15165 non-null  object
 9   remote_ratio          14861 non-null  float64
 10  required_skills       15165 non-null  object
 11  education_required    15165 non-null  object
 12  years_experience      15165 non-null  object
 13  industry              15165 non-null  object
 14  posting_date          15165 non-null  object
 15  application_deadline  15165 non-null  object
 16  job_description_length 15165 non-null  float64
 17  benefits_score        15165 non-null  float64
 18  company_name          15165 non-null  object
dtypes: float64(3), object(16)
memory usage: 2.3+ MB
```

	remote_ratio	job_description_length	benefits_score
count	14861.000000	15165.000000	15165.000000
mean	49.461678	1500.724563	7.497613

⊗ 0 ⚠ 0 Spaces: 4 LF {} Cell 6 of 33



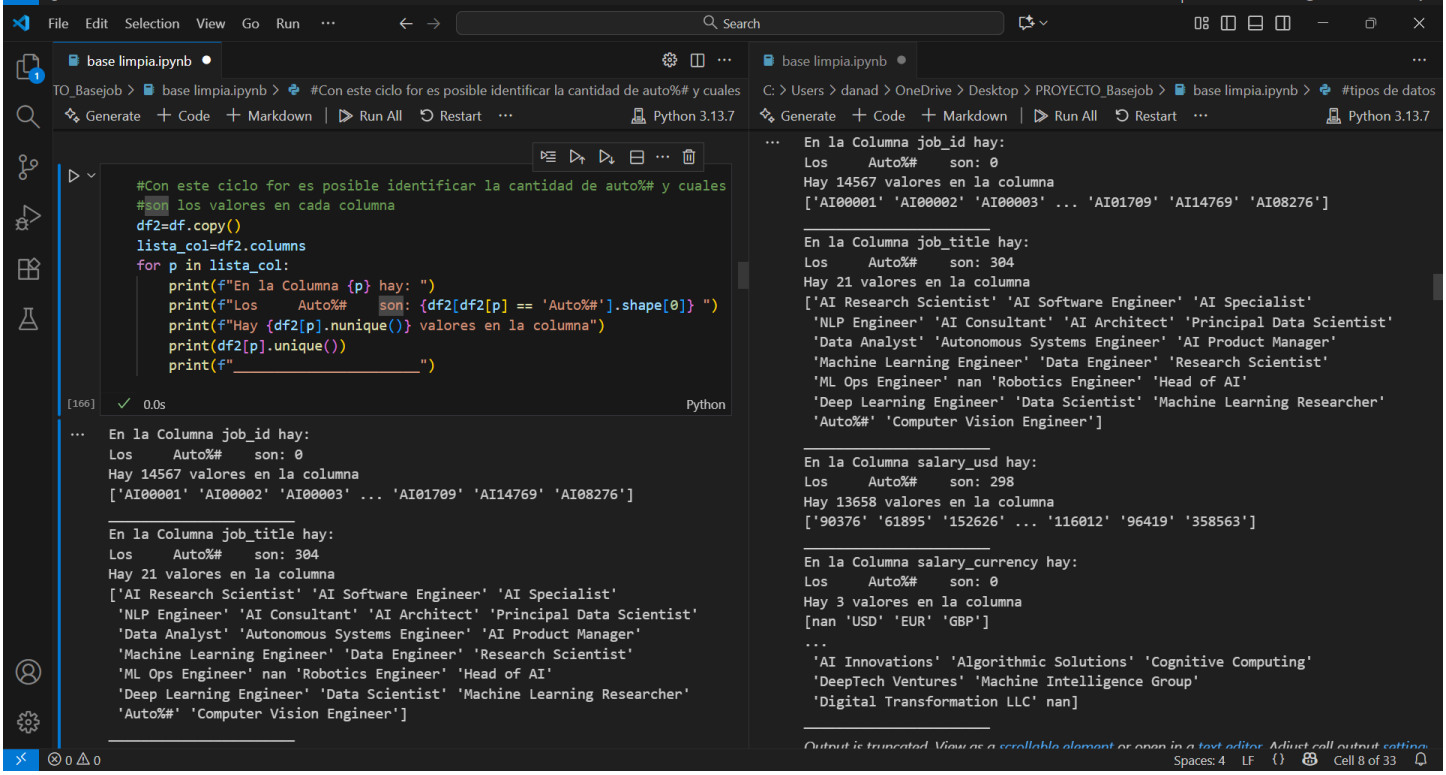
```
 7   company_size            15165 non-null  object
 8   employee_residence      15165 non-null  object
 9   remote_ratio            14861 non-null  float64
 10  required_skills         15165 non-null  object
 11  education_required      15165 non-null  object
 12  years_experience        15165 non-null  object
 13  industry                15165 non-null  object
 14  posting_date            15165 non-null  object
 15  application_deadline    15165 non-null  object
 16  job_description_length  15165 non-null  float64
 17  benefits_score          15165 non-null  float64
 18  company_name            15165 non-null  object
dtypes: float64(3), object(16)
memory usage: 2.3+ MB
```

	remote_ratio	job_description_length	benefits_score
count	14861.000000	15165.000000	15165.000000
mean	49.461678	1500.724563	7.497613
std	40.766292	576.055008	1.450236
min	0.000000	500.000000	5.000000
25%	0.000000	999.000000	6.200000
50%	50.000000	1509.000000	7.500000
75%	100.000000	1996.000000	8.800000
max	100.000000	2499.000000	10.000000

```
#Cantidad de NaN
df.isnull().sum()
```

[165] ✓ 0.0s Python

```
job_id                   468
job_title                468
salary_usd               468
salary_currency          468
experience_level         468
employment_type          468
company_location         468
company_size             468
employee_residence       468
remote_ratio             772
required_skills          468
education_required       468
years_experience         468
industry                 468
posting_date             468
application_deadline     468
job_description_length   468
benefits_score           468
company_name             468
dtype: int64
```



```
#Con este ciclo for es posible identificar la cantidad de auto%# y cuales
#son los valores en cada columna
df2=df.copy()
lista_col=df2.columns
for p in lista_col:
    print(f"En la Columna {p} hay: ")
    print(f"Los     Auto%#    son: {df2[df2[p] == 'Auto%#'].shape[0]} ")
    print(f"Hay {df2[p].nunique()} valores en la columna")
    print(df2[p].unique())
    print(f"_____________________")
```

[166] ✓ 0.0s Python

```
En la Columna job_id hay:
Los     Auto%#    son: 0
Hay 14567 valores en la columna
['AI00001' 'AI00002' 'AI00003' ... 'AI01709' 'AI14769' 'AI08276']
_____________________
En la Columna job_title hay:
Los     Auto%#    son: 304
Hay 21 valores en la columna
['AI Research Scientist' 'AI Software Engineer' 'AI Specialist'
 'NLP Engineer' 'AI Consultant' 'AI Architect' 'Principal Data Scientist'
 'Data Analyst' 'Autonomous Systems Engineer' 'AI Product Manager'
 'Machine Learning Engineer' 'Data Engineer' 'Research Scientist'
 'ML Ops Engineer' nan 'Robotics Engineer' 'Head of AI'
 'Deep Learning Engineer' 'Data Scientist' 'Machine Learning Researcher'
 'Auto%#' 'Computer Vision Engineer']
_____________________
En la Columna job_id hay:
Los     Auto%#    son: 0
Hay 14567 valores en la columna
['AI00001' 'AI00002' 'AI00003' ... 'AI01709' 'AI14769' 'AI08276']
_____________________
En la Columna job_title hay:
Los     Auto%#    son: 304
Hay 21 valores en la columna
['AI Research Scientist' 'AI Software Engineer' 'AI Specialist'
 'NLP Engineer' 'AI Consultant' 'AI Architect' 'Principal Data Scientist'
 'Data Analyst' 'Autonomous Systems Engineer' 'AI Product Manager'
 'Machine Learning Engineer' 'Data Engineer' 'Research Scientist'
 'ML Ops Engineer' nan 'Robotics Engineer' 'Head of AI'
 'Deep Learning Engineer' 'Data Scientist' 'Machine Learning Researcher'
 'Auto%#' 'Computer Vision Engineer']
_____________________
En la Columna salary_usd hay:
Los     Auto%#    son: 298
Hay 13658 valores en la columna
['90376' '61895' '152626' ... '116012' '96419' '358563']
_____________________
En la Columna salary_currency hay:
Los     Auto%#    son: 0
Hay 3 valores en la columna
[nan 'USD' 'EUR' 'GBP']
...
 'AI Innovations' 'Algorithmic Solutions' 'Cognitive Computing'
 'DeepTech Ventures' 'Machine Intelligence Group'
 'Digital Transformation LLC' nan]
_____________________
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output setting

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output setting:

```python
#Cuantos valores tiene cada columna:
for c in lista_col:
    unicos = df[c].nunique()
    print(f'Columna {c} tiene {unicos} valores únicos')
```

[167] ✓ 0.0s Python

```
Columna job_id tiene 14567 valores únicos
Columna job_title tiene 21 valores únicos
Columna salary_usd tiene 13658 valores únicos
Columna salary_currency tiene 3 valores únicos
Columna experience_level tiene 5 valores únicos
Columna employment_type tiene 5 valores únicos
Columna company_location tiene 20 valores únicos
Columna company_size tiene 3 valores únicos
Columna employee_residence tiene 21 valores únicos
Columna remote_ratio tiene 3 valores únicos
Columna required_skills tiene 13289 valores únicos
Columna education_required tiene 5 valores únicos
Columna years_experience tiene 21 valores únicos
Columna industry tiene 15 valores únicos
Columna posting_date tiene 486 valores únicos
Columna application_deadline tiene 544 valores únicos
Columna job_description_length tiene 1999 valores únicos
Columna benefits_score tiene 51 valores únicos
Columna company_name tiene 16 valores únicos
```

3. Detección y manejo de duplicados

Con este comando se crea un booleano que muestra renglones duplicados

```python
df2.duplicated()
```

[168] ✓ 0.0s Python

```
0        False
1        False
2        False
3        False
4        False
         ...
15628    False
15629    False
15630    False
15631    False
15632    False
Length: 15633, dtype: bool
```

```python
df2.duplicated().sum()
```

[169] ✓ 0.0s Python

```
np.int64(165)
```

En las columnas de la lista "colum_float" se remplazan por 0, un valor numerico.

```python
df3 = df3.copy()
colum_float=['salary_usd', 'years_experience','remote_ratio', 'job_description_length','benefits_score' ]
df3.replace(to_replace='Auto%#', value=0, regex=True, inplace=True)

for col in colum_float:
    df3[col] = pd.to_numeric(df3[col], errors='coerce')
    #Convierte a Nan SI NO ES NUMERO
    promedio = df3[col].mean()
    promedio_redondeado = round(promedio, 2)
    # Lo anterior ayuda a redondear el promedio para mantener consistencia y evitar decimales demasiado largos
    df3[col] = df3[col].fillna(promedio_redondeado)
    print(f'Para columna: {col} // Valores nulos reemplazados por: {promedio_redondeado}')
df3['salary_usd'] = df3['salary_usd'].apply(lambda x: f"${x:,.2f}")
```

[244] ✓ 0.2s Python

```
Para columna: salary_usd // Valores nulos reemplazados por: 112960.55
Para columna: years_experience // Valores nulos reemplazados por: 6.12
Para columna: remote_ratio // Valores nulos reemplazados por: 49.4
Para columna: job_description_length // Valores nulos reemplazados por: 1501.47
Para columna: benefits_score // Valores nulos reemplazados por: 7.5
```

Con la ultima linea de codigo cambiamos el formato de la columna "salary_usd" a una monetaria

```python
df3.head()
```

[245] ✓ 0.0s Python

File Edit Selection View Go Run ··· Search

base limpia.ipynb

C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > df3 = df3.copy()

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· Python 3.13.7

Con la ultima linea de codigo cambiamos el formato de la columna "salary_usd" a una monetaria

```
df3.head()
```

[245] ✓ 0.0s Python

	job_id	job_title	salary_usd	salary_currency	experience_level	employment_type	company_location	company_size	employee_residence	remote_ratio	required_skills	e
0	AI00001	AI Research Scientist	$90,376.00	NaN	SE	CT	China	M	China	50.0	Tableau, PyTorch, Kubernetes, Linux, NLP	
1	AI00002	AI Software Engineer	$61,895.00	USD	EN	CT	Canada	M	Ireland	100.0	Deep Learning, AWS, Mathematics, Python, Docker	
2	AI00003	AI Specialist	$152,626.00	USD	MI	FL	Switzerland	L	South Korea	0.0	Kubernetes, Deep Learning, Java, Hadoop, NLP	
3	AI00004	NLP Engineer	$80,215.00	USD	SE	FL	India	M	0	50.0	Scala, SQL, Linux, Python	
4	AI00005	AI Consultant	$54,624.00	EUR	EN	PT	France	S	Singapore	100.0	MLOps, Java, Tableau, Python	

⊗ 0 ⚠ 0 Spaces: 4 {} Cell 19 of 34

File Edit Selection View Go Run ··· Search

base limpia.ipynb

C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > df3.head()

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· Python 3.13.7

```
#creando un ciclo for para rellenar los nan, seleccionando las columnas donde se puedan remplazar como datos tipo objeto
colum_object=['job_id','job_title', 'salary_currency',
       'experience_level', 'employment_type', 'company_location',
       'company_size', 'employee_residence', 'required_skills',
       'education_required', 'industry', 'posting_date',
       'application_deadline','company_name']
for k in colum_object:
    df3[k]=df3[k].fillna('sin dato')
df3.head()
```

[246] ✓ 0.1s Python

	job_id	job_title	salary_usd	salary_currency	experience_level	employment_type	company_location	company_size	employee_residence	remote_ratio	required_skills	e
0	AI00001	AI Research Scientist	$90,376.00	sin dato	SE	CT	China	M	China	50.0	Tableau, PyTorch, Kubernetes, Linux, NLP	
1	AI00002	AI Software Engineer	$61,895.00	USD	EN	CT	Canada	M	Ireland	100.0	Deep Learning, AWS, Mathematics, Python, Docker	
2	AI00003	AI Specialist	$152,626.00	USD	MI	FL	Switzerland	L	South Korea	0.0	Kubernetes, Deep Learning, Java, Hadoop, NLP	
3	AI00004	NLP Engineer	$80,215.00	USD	SE	FL	India	M	0	50.0	Scala, SQL, Linux, Python	
		AI									MLOps, Java,	

⊗ 0 ⚠ 0 Spaces: 4 {} Cell 21 of 34

File Edit Selection View Go Run ··· Search

base limpia.ipynb ●

C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > #Renombrar las columnas

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· Python 3.13.7

6. Traducción de textos al español

Tras una investigacion existen varias formas de traducir los datos, una de ellas es atraves de las librerias, como googletrans: instalada con el comando: pip install googletrans

Sin embargo aveces para datos limitados pueden ser mejores los diccionarios manuales

```python
#Renombrar las columnas
df4= df3.rename(columns={'job_id': 'id_trabajor', 'job_title':'titulo_trabajador', 'salary_currency':'Moneda_salario',
       'experience_level':'Nivel_experencia', 'employment_type':'tipo_empleo', 'company_location':'Ubicacion_empresa',
       'company_size':'tamaño_empresa', 'employee_residence':'residencia_empleado', 'required_skills':'habilidades_empleados',
       'education_required':'educacion_requerida', 'industry':'industria', 'posting_date':'fecha_publicacion',
       'application_deadline':'fecha_limite','company_name':'nombre_empresa'  })
df4
```

[247] ✓ 0.0s Python

	id_trabajor	titulo_trabajador	salary_usd	Moneda_salario	Nivel_experencia	tipo_empleo	Ubicacion_empresa	tamaño_empresa	residencia_empleado	remote_ratio	hab
0	AI00001	AI Research Scientist	$90,376.00	sin dato	SE	CT	China	M	China	50.0	K
1	AI00002	AI Software Engineer	$61,895.00	USD	EN	CT	Canada	M	Ireland	100.0	I
2	AI00003	AI Specialist	$152,626.00	USD	MI	FL	Switzerland	L	South Korea	0.0	Le
3	AI00004	NLP Engineer	$80,215.00	USD	SE	FL	India	M	0	50.0	Sca

⊗ 0 ⚠ 0 Spaces: 7 {} Cell 25 of 34

File Edit Selection View Go Run ··· Search

base limpia.ipynb ●

C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > # Diccionario de traducción

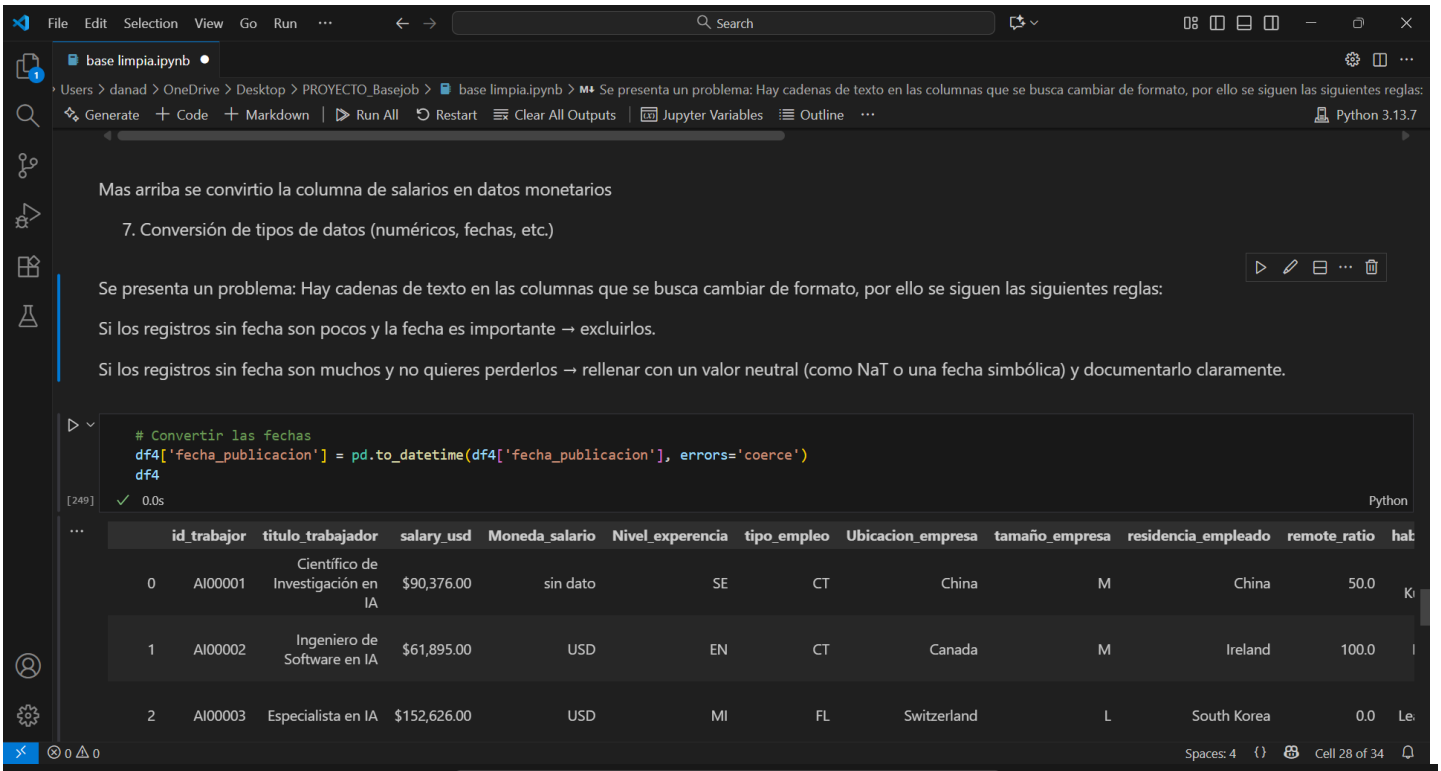
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· Python 3.13.7

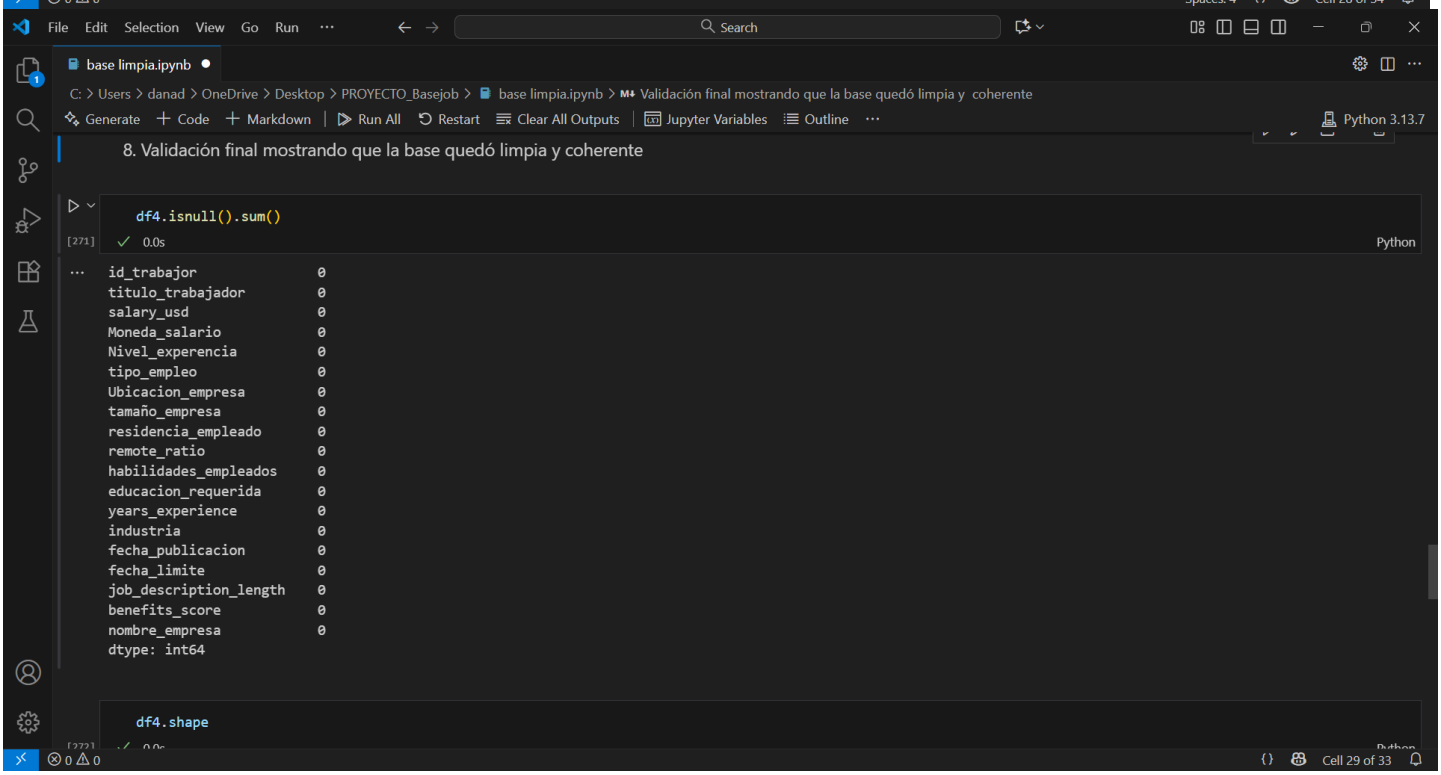
```python
# Diccionario de traducción
traduccion_titulos = {
    'AI Research Scientist': 'Científico de Investigación en IA',
    'AI Software Engineer': 'Ingeniero de Software en IA',
    'AI Specialist': 'Especialista en IA',
    'NLP Engineer': 'Ingeniero de PLN',
    'AI Consultant': 'Consultor de IA',
    'AI Architect': 'Arquitecto de IA',
    'Principal Data Scientist': 'Científico de Datos Principal',
    'Data Analyst': 'Analista de Datos',
    'Autonomous Systems Engineer': 'Ingeniero de Sistemas Autónomos',
    'AI Product Manager': 'Gerente de Producto de IA',
    'Machine Learning Engineer': 'Ingeniero de Aprendizaje Automático',
    'Data Engineer': 'Ingeniero de Datos',
    'Research Scientist': 'Científico Investigador',
    'ML Ops Engineer': 'Ingeniero de MLOps',
    'Robotics Engineer': 'Ingeniero en Robótica',
    'Head of AI': 'Jefe de IA',
    'Deep Learning Engineer': 'Ingeniero de Aprendizaje Profundo',
    'Data Scientist': 'Científico de Datos',
    'Machine Learning Researcher': 'Investigador en Aprendizaje Automático',
    'Computer Vision Engineer': 'Ingeniero de Visión por Computadora'
}
# Reemplazar los nombres de los días en la columna 'dianombre'
df4['titulo_trabajador'] = df4['titulo_trabajador'].replace(traduccion_titulos)
df4
```

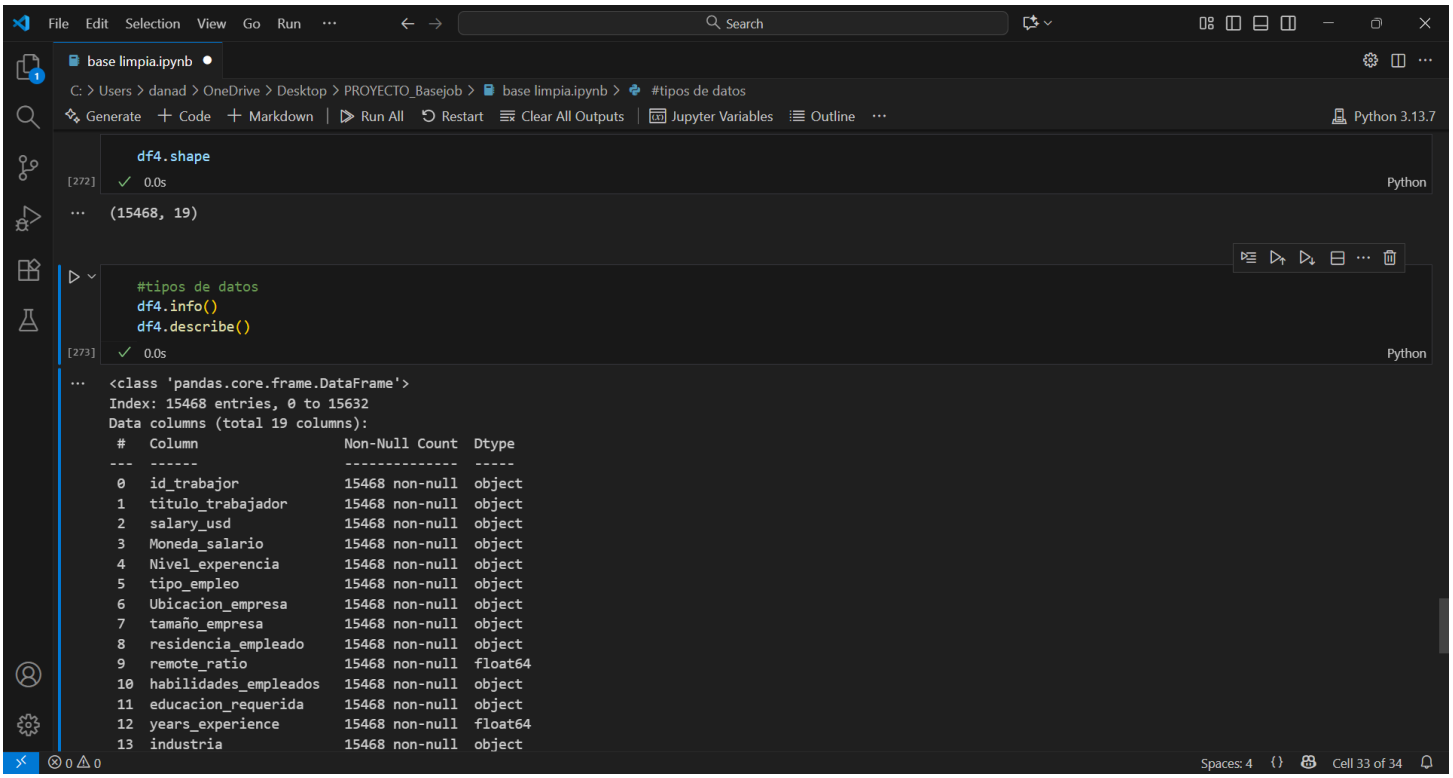
[248] ✓ 0.0s Python

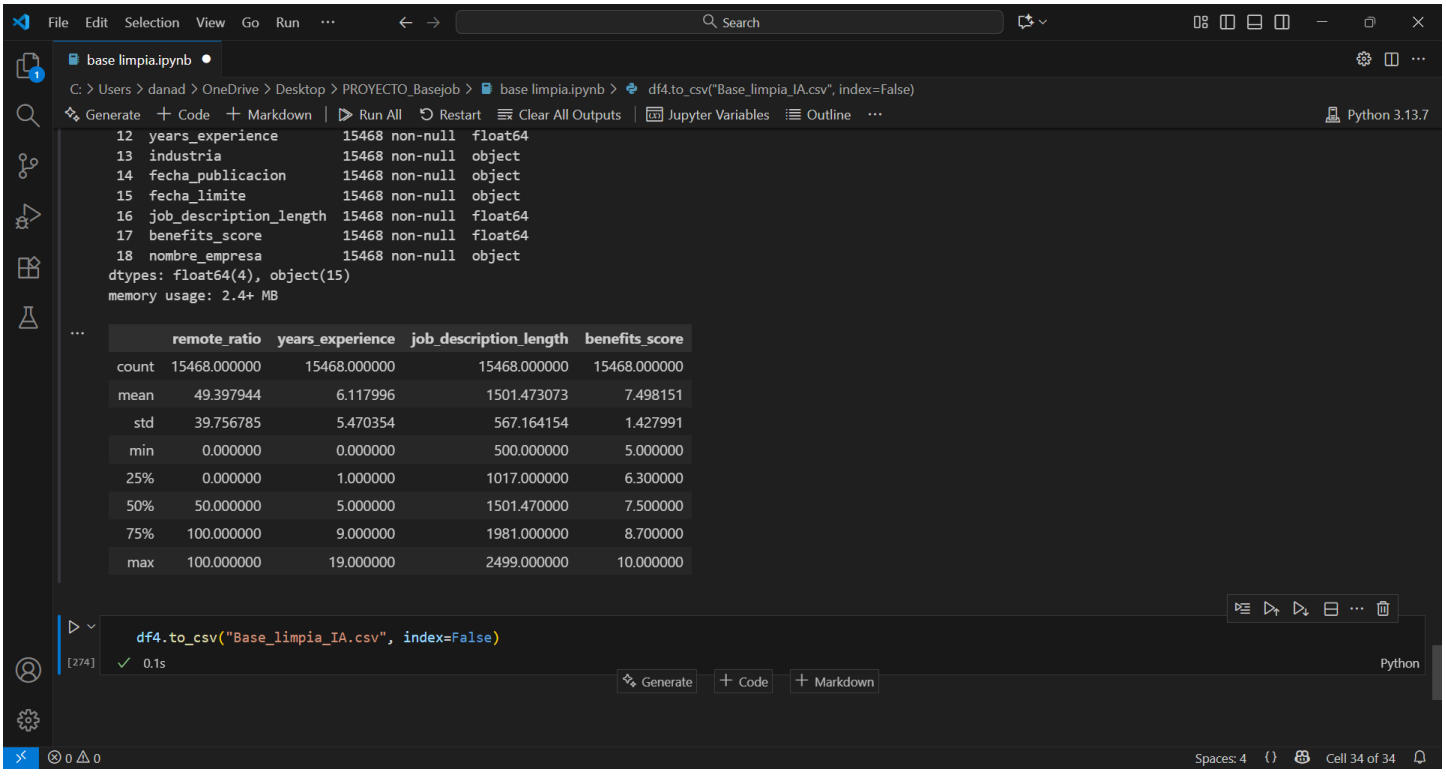
	id_trabajor	titulo_trabajador	salary_usd	Moneda_salario	Nivel_experencia	tipo_empleo	Ubicacion_empresa	tamaño_empresa	residencia_empleado	remote_ratio	hab

⊗ 0 ⚠ 0 Spaces: 4 {} Cell 26 of 34


File Edit Selection View Go Run
Search
base limpia.ipynb
Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > Se presenta un problema: Hay cadenas de texto en las columnas que se busca cambiar de formato, por ello se siguen las siguientes reglas:
Generate Code Markdown Run All Restart Clear All Outputs Jupyter Variables Outline
Python 3.13.7
Mas arriba se convirtio la columna de salarios en datos monetarios
7. Conversión de tipos de datos (numéricos, fechas, etc.)
Se presenta un problema: Hay cadenas de texto en las columnas que se busca cambiar de formato, por ello se siguen las siguientes reglas:
Si los registros sin fecha son pocos y la fecha es importante → excluirlos.
Si los registros sin fecha son muchos y no quieres perderlos → rellenar con un valor neutral (como NaT o una fecha simbólica) y documentarlo claramente.
Convertir las fechas
df4['fecha_publicacion'] = pd.to_datetime(df4['fecha_publicacion'], errors='coerce')
df4
[249] 0.0s
Python
id_trabajor titulo_trabajador salary_usd Moneda_salario Nivel_experencia tipo_empleo Ubicacion_empresa tamaño_empresa residencia_empleado remote_ratio
0 AI00001 Científico de Investigación en IA $90,376.00 sin dato SE CT China M China 50.0
1 AI00002 Ingeniero de Software en IA $61,895.00 USD EN CT Canada M Ireland 100.0
2 AI00003 Especialista en IA $152,626.00 USD MI FL Switzerland L South Korea 0.0
Spaces: 4
Cell 28 of 34


File Edit Selection View Go Run
Search
base limpia.ipynb
C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > Validación final mostrando que la base quedó limpia y coherente
Generate Code Markdown Run All Restart Clear All Outputs Jupyter Variables Outline
Python 3.13.7
8. Validación final mostrando que la base quedó limpia y coherente
df4.isnull().sum()
[271] 0.0s
Python
id_trabajor 0
titulo_trabajador 0
salary_usd 0
Moneda_salario 0
Nivel_experencia 0
tipo_empleo 0
Ubicacion_empresa 0
tamaño_empresa 0
residencia_empleado 0
remote_ratio 0
habilidades_empleados 0
educacion_requerida 0
years_experience 0
industria 0
fecha_publicacion 0
fecha_limite 0
job_description_length 0
benefits_score 0
nombre_empresa 0
dtype: int64
df4.shape
Cell 29 of 33


File Edit Selection View Go Run
Search
base limpia.ipynb
C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > #tipos de datos
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline
Python 3.13.7
df4.shape
[272] 0.0s
Python
(15468, 19)
#tipos de datos
df4.info()
df4.describe()
[273] 0.0s
Python
<class 'pandas.core.frame.DataFrame'>
Index: 15468 entries, 0 to 15632
Data columns (total 19 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 id_trabajor 15468 non-null object
1 titulo_trabajador 15468 non-null object
2 salary_usd 15468 non-null object
3 Moneda_salario 15468 non-null object
4 Nivel_experencia 15468 non-null object
5 tipo_empleo 15468 non-null object
6 Ubicacion_empresa 15468 non-null object
7 tamaño_empresa 15468 non-null object
8 residencia_empleado 15468 non-null object
9 remote_ratio 15468 non-null float64
10 habilidades_empleados 15468 non-null object
11 educacion_requerida 15468 non-null object
12 years_experience 15468 non-null float64
13 industria 15468 non-null object
Spaces: 4 Cell 33 of 34


File Edit Selection View Go Run
Search
base limpia.ipynb
C: > Users > danad > OneDrive > Desktop > PROYECTO_Basejob > base limpia.ipynb > df4.to_csv("Base_limpia_IA.csv", index=False)
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline
Python 3.13.7
12 years_experience 15468 non-null float64
13 industria 15468 non-null object
14 fecha_publicacion 15468 non-null object
15 fecha_limite 15468 non-null object
16 job_description_length 15468 non-null float64
17 benefits_score 15468 non-null float64
18 nombre_empresa 15468 non-null object
dtypes: float64(4), object(15)
memory usage: 2.4+ MB
remote_ratio years_experience job_description_length benefits_score
count 15468.000000 15468.000000 15468.000000 15468.000000
mean 49.397944 6.117996 1501.473073 7.498151
std 39.756785 5.470354 567.164154 1.427991
min 0.000000 0.000000 500.000000 5.000000
25% 0.000000 1.000000 1017.000000 6.300000
50% 50.000000 5.000000 1501.470000 7.500000
75% 100.000000 9.000000 1981.000000 8.700000
max 100.000000 19.000000 2499.000000 10.000000
df4.to_csv("Base_limpia_IA.csv", index=False)
[274] 0.1s
Python
Generate + Code + Markdown
Spaces: 4 Cell 34 of 34

Conclusiones

Durante el proceso de limpieza detecte varios problemas comunes en bases de datos reales. Solo que en este caso no fue ensuciada de manera natural, la corrección de estos errores permitió obtener un conjunto de datos más coherente, uniforme y listo para poder analizar siguiendo el objetivo principal del proyecto. Esto demuestra la importancia de la limpieza de datos como paso previo a cualquier análisis estadístico o de machine learning, ya que los errores iniciales pueden alterar significativamente los resultados.

Problemas principales de la base

Existencia de valores nulos en columnas numéricas y categóricas.

Registros duplicados, que distorsionaban los conteos y promedios.

Datos inconsistentes o mal formateados, como la combinación de tipo de datos entre columnas, haciendo imposible realizar ciertas operaciones, hasta que se realizara un cabio en el tipo de dato de la columna.

Qué técnicas aplicaste para solucionarlos.

Eliminación de duplicados con el comando:

df.drop_duplicates(inplace=True)

Sustitución de valores faltantes por el promedio de la columna utilizando:

df.fillna(df.mean(), inplace=True)

Limpieza de formato en los datos numéricos con expresiones regulares o funciones str.replace() para eliminar símbolos innecesarios.

Estandarización de nombres de columnas con:

df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")

Convertir datos enteros a monetarios:

df3['salary_usd'] = df3['salary_usd'].apply(lambda x: f"${x:,.2f}")

entre otros...

Aprendizaje obtenido del proceso

Aprendí que la limpieza de datos es una de las etapas más importantes y laboriosas del análisis de datos. Requiere atención al detalle, conocimiento de las funciones de pandas y comprensión del contexto de la información. También comprendí que pequeñas inconsistencias pueden generar grandes errores en los resultados, por lo que validar y verificar los datos después de cada paso es esencial para garantizar la calidad del análisis.