Data Cleaning with Pandas

 Cell 1: Import Required Libraries

```
import pandas as pd
import numpy as np
```

 Cell 2: Load the CSV File

```
df = pd.read_csv("/content/Messy_Employment_India_Dataset.csv")
```

 Cell 3: View the Dataset

```
df.head(110)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 2000,\n  \"fields\":
[\n    {\n      \"column\": \"Status\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 6,\n
\"samples\": [\n          \"EMPLOYED\",\n          \"UNEMPLOYED\",\n
\"Employed\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Age
Group\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n
\"25_34\",\n          \"18-24\",\n          \"25-34\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n        \"samples\": [\n
\"Bachelors\",\n          \"Diploma\",\n          \"Master\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Industry\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          \" Fintech\",\n          \"Healthcare\",\n
\"Technology\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Location\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          \"rural\",\n          \"Mumbai\",\n          \"Urban\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"AI Risk\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 6,\n        \"samples\": [\n
\"moderate\",\n          \"Low\",\n          \"High\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Years of Experience\",\n
```

```
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
9.103940962675752,\n        \"min\": 0.0,\n        \"max\": 30.0,\n
\"num_unique_values\": 31,\n        \"samples\": [\n          19.0,\n
8.0,\n          27.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Monthly Salary (INR)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 41628.00805409873,\n        \"min\":
5100.0,\n        \"max\": 149900.0,\n        \"num_unique_values\":
978,\n        \"samples\": [\n          126900.0,\n
117100.0,\n          31500.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Date Recorded\",\n      \"properties\": {\n
\"dtype\": \"object\",\n        \"num_unique_values\": 2000,\n
\"samples\": [\n          \"2/4/2028\",\n          \"12/20/2023\",\n
\"8/26/2026\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


 Cell 4: Understand Dataset Structure

```
df.shape      # number of rows and columns
```

```
(2000, 9)
```

```
df.info()     # data types and missing values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Status                1732 non-null   object
 1   Age Group             1768 non-null   object
 2   Education             1804 non-null   object
 3   Industry              1799 non-null   object
 4   Location              1787 non-null   object
 5   AI Risk               1716 non-null   object
 6   Years of Experience   980 non-null    float64
 7   Monthly Salary (INR)  1613 non-null   float64
 8   Date Recorded         2000 non-null   object
dtypes: float64(2), object(7)
memory usage: 140.8+ KB
```

```
df.describe() # statistical summary
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"Years of Experience\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
341.5416910202622,\n        \"min\": 0.0,\n        \"max\": 980.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
```

```
15.244897959183673,\n          16.0,\n          980.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Monthly Salary (INR)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
51354.52706818114,\n        \"min\": 1613.0,\n        \"max\":
149900.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
76886.3608183509,\n          77200.0,\n          1613.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


 Cell 5: Check Missing Values

```
df.isnull()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 2000,\n  \"fields\":
[\n    {\n      \"column\": \"Status\",\n      \"properties\": {\n
\"dtype\": \"boolean\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          true,\n          false\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age Group\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          true,\n
false\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Education\",\n      \"properties\": {\n        \"dtype\":
\"boolean\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          true,\n          false\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Industry\",\n      \"properties\":
{\n        \"dtype\": \"boolean\",\n        \"num_unique_values\": 2,\
n        \"samples\": [\n          true,\n          false\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Location\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          true,\n
false\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"AI
Risk\",\n      \"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          false,\n
true\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Years of Experience\",\n      \"properties\": {\n        \"dtype\":
\"boolean\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          false,\n          true\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Monthly Salary (INR)\",\n
\"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          true,\n
false\n        ],\n        \"semantic_type\": \"\",\n
```

```
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Date Recorded\",\n      \"properties\": {\n        \"dtype\":
\"boolean\",\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          false\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
df.isnull().sum()
```

```
Status                    268
Age Group                 232
Education                 196
Industry                  201
Location                  213
AI Risk                   284
Years of Experience      1020
Monthly Salary (INR)      387
Date Recorded               0
dtype: int64
```

Cell 6: Handle Missing Values

Fill missing numerical values with mean

```
df['Years of Experience'] = df['Years of Experience'].fillna(df['Years
of Experience'].mean())
```

```
df
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 2000,\n  \"fields\":
[\n    {\n      \"column\": \"Status\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 6,\n
\"samples\": [\n          \"EMPLOYED\",\n          \"UNEMPLOYED\",\n
\"Employed\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Age
Group\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n
\"25_34\",\n          \"18-24\",\n          \"25-34\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n        \"samples\": [\n
\"Bachelors\",\n          \"Diploma\",\n          \"Master\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Industry\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          \" Fintech\",\n          \"Healthcare\",\n
\"Technology\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Location\",\n      \"properties\": {\n        \"dtype\":
```

```
\"category\",\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          \"rural\",\n          \"Mumbai\",\n          \"Urban\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"AI Risk\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 6,\n        \"samples\": [\n
\"moderate\",\n          \"Low\",\n          \"High\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Years of Experience\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
6.371099410521485,\n        \"min\": 0.0,\n        \"max\": 30.0,\n
\"num_unique_values\": 32,\n        \"samples\": [\n          29.0,\n
2.0,\n          27.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Monthly Salary (INR)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 41628.00805409873,\n        \"min\":
5100.0,\n        \"max\": 149900.0,\n        \"num_unique_values\":
978,\n        \"samples\": [\n          126900.0,\n
117100.0,\n          31500.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Date Recorded\",\n      \"properties\": {\n
\"dtype\": \"object\",\n        \"num_unique_values\": 2000,\n
\"samples\": [\n          \"2/4/2028\",\n          \"12/20/2023\",\n
\"8/26/2026\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


Fill missing categorical values

```
df['Location'] = df['Location'].fillna("Unknown")

df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 2000,\n  \"fields\":
[\n    {\n      \"column\": \"Status\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 6,\n
\"samples\": [\n          \"EMPLOYED\",\n          \"UNEMPLOYED\",\n
\"Employed\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Age
Group\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n
\"25_34\",\n          \"18-24\",\n          \"25-34\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n        \"samples\": [\n
\"Bachelors\",\n          \"Diploma\",\n          \"Master\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"Industry\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          \" Fintech\",\n          \"Healthcare\",\n
\"Technology\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Location\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 9,\n        \"samples\":
[\n          \"urban\",\n          \"rural\",\n          \"Mumbai\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"AI Risk\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 6,\n        \"samples\": [\n
\"moderate\",\n          \"Low\",\n          \"High\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Years of Experience\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
6.371099410521485,\n        \"min\": 0.0,\n        \"max\": 30.0,\n
\"num_unique_values\": 32,\n        \"samples\": [\n          29.0,\n
2.0,\n          27.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Monthly Salary (INR)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 41628.00805409873,\n        \"min\":
5100.0,\n        \"max\": 149900.0,\n        \"num_unique_values\":
978,\n        \"samples\": [\n          126900.0,\n
117100.0,\n          31500.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Date Recorded\",\n      \"properties\": {\n
\"dtype\": \"object\",\n        \"num_unique_values\": 2000,\n
\"samples\": [\n          \"2/4/2028\",\n          \"12/20/2023\",\n
\"8/26/2026\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


Cell 7: Remove Duplicate Rows

```
df = df.drop_duplicates()
```

Cell 8: Check and Fix Data Types

```
df.dtypes
```

```
Status                   object
Age Group                object
Education                object
Industry                 object
Location                 object
AI Risk                  object
Years of Experience     float64
Monthly Salary (INR)    float64
```

```
Date Recorded            object
dtype: object
```

```
df['Age Group'] = df['Age Group'].astype(int)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/tmp/ipython-input-2511504638.py in <cell line: 0>()
----> 1 df['Age Group'] = df['Age Group'].astype(int)

/usr/local/lib/python3.12/dist-packages/pandas/core/generic.py in astype(self, dtype, copy, errors)
   6641         else:
   6642             # else, only a single dtype is given
-> 6643             new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
   6644             res = self._constructor_from_mgr(new_data, axes=new_data.axes)
   6645             return res.__finalize__(self, method="astype")

/usr/local/lib/python3.12/dist-packages/pandas/core/internals/managers.py in astype(self, dtype, copy, errors)
    428             copy = False
    429
--> 430         return self.apply(
    431             "astype",
    432             dtype=dtype,

/usr/local/lib/python3.12/dist-packages/pandas/core/internals/managers.py in apply(self, f, align_keys, **kwargs)
    361                 applied = b.apply(f, **kwargs)
    362             else:
--> 363                 applied = getattr(b, f)(**kwargs)
    364             result_blocks = extend_blocks(applied, result_blocks)
    365

/usr/local/lib/python3.12/dist-packages/pandas/core/internals/blocks.py in astype(self, dtype, copy, errors, using_cow, squeeze)
    756             values = values[0, :]  # type: ignore[call-overload]
    757
--> 758         new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
    759
    760         new_values = maybe_coerce_values(new_values)

/usr/local/lib/python3.12/dist-packages/pandas/core/dtypes/astype.py
```

```
in astype_array_safe(values, dtype, copy, errors)
    235
    236     try:
--> 237         new_values = astype_array(values, dtype, copy=copy)
    238     except (ValueError, TypeError):
    239         # e.g. _astype_nansafe can fail on object-dtype of
strings

/usr/local/lib/python3.12/dist-packages/pandas/core/dtypes/astype.py
in astype_array(values, dtype, copy)
    180
    181     else:
--> 182         values = _astype_nansafe(values, dtype, copy=copy)
    183
    184     # in pandas we don't store numpy str dtypes, so convert to
object

/usr/local/lib/python3.12/dist-packages/pandas/core/dtypes/astype.py
in _astype_nansafe(arr, dtype, copy, skipna)
    131     if copy or arr.dtype == object or dtype == object:
    132         # Explicit copy, or required since NumPy can't view
from / to object.
--> 133         return arr.astype(dtype, copy=True)
    134
    135     return arr.astype(dtype, copy=copy)

ValueError: invalid literal for int() with base 10: '18-24'
```

 Cell 9: Rename Columns (Optional)

```
df = df.rename(columns={
    'Years of Experience': 'YOF',
    'Location': 'Loc'
})

df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 2000,\n  \"fields\":
[\n    {\n      \"column\": \"Status\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 6,\n
\"samples\": [\n          \"EMPLOYED\",\n          \"UNEMPLOYED\",\n
\"Employed\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Age
Group\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n
\"25_34\",\n          \"18-24\",\n          \"25-34\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n        \"samples\": [\n
```

```
\"Bachelors\",\n          \"Diploma\",\n          \"Master\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Industry\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          \" Fintech\",\n          \"Healthcare\",\n
\"Technology\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Loc\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n        \"samples\": [\n
\"urban\",\n          \"rural\",\n          \"Mumbai\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"AI Risk\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
6,\n        \"samples\": [\n          \"moderate\",\n
\"Low\",\n          \"High\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"YOF\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 6.371099410521485,\n        \"min\":
0.0,\n        \"max\": 30.0,\n        \"num_unique_values\": 32,\n
\"samples\": [\n          29.0,\n          2.0,\n          27.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Monthly Salary (INR)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
41628.00805409873,\n        \"min\": 5100.0,\n        \"max\":
149900.0,\n        \"num_unique_values\": 978,\n        \"samples\":
[\n          126900.0,\n          117100.0,\n          31500.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Date Recorded\",\n
\"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 2000,\n        \"samples\": [\n
\"2/4/2028\",\n          \"12/20/2023\",\n          \"8/26/2026\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


 Cell 10: Handle Outliers (Simple Filtering)

```
df_sort = df[df['Monthly Salary (INR)'] < 10000]
df_sort.head()
```


```
{"repr_error":"0","type":"dataframe","variable_name":"df_sort"}
```


 Cell 11: Clean Text Data

```
#Removes extra spaces from the beginning and end of each city name.
df['Education'] = df['Education'].str.strip()
#Converts all city names to lowercase letters.
df['Loc'] = df['Loc'].str.lower()
```

 Cell 12: Data Validation

```
df = df[df['YOF'] > 0]
#df = df[(df['Age'] > 0) & (df['Age'] <= 120)]
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 1965,\n  \"fields\":
[\n    {\n      \"column\": \"Status\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 6,\n
\"samples\": [\n          \"EMPLOYED\",\n          \"UNEMPLOYED\",\n
\"Employed\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Age
Group\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n
\"25_34\",\n          \"18-24\",\n          \"25-34\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n        \"samples\": [\n
\"Bachelors\",\n          \"Diploma\",\n          \"Master\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Industry\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          \" Fintech\",\n          \"Healthcare\",\n
\"Technology\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Loc\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 8,\n        \"samples\": [\n
\"rural\",\n          \"mumbai\",\n          \"urban\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"AI Risk\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
6,\n        \"samples\": [\n          \"moderate\",\n
\"Low\",\n          \"Medium\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"YOF\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 6.090880707692265,\n        \"min\":
1.0,\n        \"max\": 30.0,\n        \"num_unique_values\": 31,\n
\"samples\": [\n          19.0,\n          8.0,\n          27.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Monthly Salary (INR)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
41699.44279962968,\n        \"min\": 5100.0,\n        \"max\":
149900.0,\n        \"num_unique_values\": 965,\n        \"samples\":
[\n          34600.0,\n          116700.0,\n          31500.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Date Recorded\",\n
\"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 1965,\n        \"samples\": [\n
```

\"5/16/2028\",\n   \"10/22/2025\",\n   \"2/28/2023\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe","variable_name":"df"}

 Cell 13: Final Check

```
df.head()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 1965,\n \"fields\":
[\n   {\n   \"column\": \"Status\",\n   \"properties\": {\n
\"dtype\": \"category\",\n   \"num_unique_values\": 6,\n
\"samples\": [\n   \"EMPLOYED\",\n   \"UNEMPLOYED\",\n
\"Employed\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Age
Group\",\n   \"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n   \"samples\": [\n
\"25_34\",\n   \"18-24\",\n   \"25-34\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Education\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 9,\n   \"samples\": [\n
\"Bachelors\",\n   \"Diploma\",\n   \"Master\"\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Industry\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 8,\n   \"samples\":
[\n   \" Fintech\",\n   \"Healthcare\",\n
\"Technology\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Loc\",\n   \"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 8,\n   \"samples\": [\n
\"rural\",\n   \"mumbai\",\n   \"urban\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"AI Risk\",\n   \"properties\":
{\n   \"dtype\": \"category\",\n   \"num_unique_values\":
6,\n   \"samples\": [\n   \"moderate\",\n
\"Low\",\n   \"Medium\"\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"YOF\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 6.090880707692265,\n   \"min\":
1.0,\n   \"max\": 30.0,\n   \"num_unique_values\": 31,\n
\"samples\": [\n   19.0,\n   8.0,\n   27.0\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Monthly Salary (INR)\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
41699.44279962968,\n   \"min\": 5100.0,\n   \"max\":
149900.0,\n   \"num_unique_values\": 965,\n   \"samples\":
[\n   34600.0,\n   116700.0,\n   31500.0\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n

```
}\n    },\n    {\n      \"column\": \"Date Recorded\",\n
\"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 1965,\n        \"samples\": [\n
\"5/16/2028\",\n          \"10/22/2025\",\n          \"2/28/2023\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1965 entries, 0 to 1999
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Status                1702 non-null   object
 1   Age Group             1738 non-null   object
 2   Education             1775 non-null   object
 3   Industry              1767 non-null   object
 4   Loc                   1965 non-null   object
 5   AI Risk               1684 non-null   object
 6   YOF                   1965 non-null   float64
 7   Monthly Salary (INR)  1582 non-null   float64
 8   Date Recorded         1965 non-null   object
dtypes: float64(2), object(7)
memory usage: 218.1+ KB
```

 Cell 14: Save Cleaned Data

```
df.to_csv("cleaned_data.csv", index=False)
```