Step 1 - Finding Errors

Dataset contains 8 columns Unnamed column, ID, Name, Age, Email, Join Date,Salary,Department

1. Raws contain only ID and other values are null.
2. Raws with no value for Name field.
3. Null values.
4. Invalid email.
5. Department contains extra characters in the department name.
6. Age as float value.
7. Date in different formats.
8. Duplicate data.
9. Name contains extra words.

Step 2 - Dataset Cleaning

1. Open Jupyter notebook
2. Create a new notebook

Steps of removing errors

Step 1. Import python library pandas-pandas is used to analyze data

Step 2. Import pandas as pd (importing as pd,we can use pd instead of pandas)

Step 3. Load a dataset(CSV file) into a Pandas DataFrame using read_csv() method

```
df=pd.read_csv("path of dataset")
```

Step 4. Get information about dataset(Non-null columns count,column type(float,object,int)

```
df.info()
```

Step 5. Remove duplicates based on ID

```
df = df.drop_duplicates(subset=["ID"], keep=False)
```

Step 6. Finding count of Null values in Name column

```
df['Name'].isnull().sum()
```

Step 7. Removing rows that contains Null in Name field

```
df.dropna(subset=['Name'], inplace=True)
```

Step 8. Again check count of Null values in Name column-it results in zero if the rows are removed

Step 9. Finding count of Null values in Age column

```
df['Age'].isnull().sum()
```

Step 10. Replace Null values with mean value of that column

```
df['Age'].fillna((df['Age'].mean()), inplace=True)
```

Using fillna() method to fill null values

Step 11. Age is given as float type,changing it to integer type

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

First, this method rounds the floating-point numbers to the nearest integer using Python's round() function, then converts them to integers using astype(int)

Step 12. Check for Null values in Salary Column

```
df['Salary'].isnull().sum()
```

Step 13. Fill null values in salary column with median of the column

```
df['Salary'].fillna((df['Salary'].median()), inplace=True)
```

Step 14. Salary change to one decimal value

```
df['Salary'] = df['Salary'].round(1)
```

Step 15. Fill null values in the Join Date column with values next to the cell,

fillna() method is used for fill null values

The ffill() method replaces the NULL values with the value from the previous row

```
df['Join Date'] = df['Join Date'].fillna(method='ffill')
```

The bfill() method backward fill the missing values in the dataset

```
df['Join Date'] = df['Join Date'].fillna(method='bfill')
```

Step 16. Checking on Join date column, making all values in same format

```
df['Join Date']=df['Join Date'].apply(lambda
x:pd.to_datetime(x).strftime('%d/%m/%Y'))
```

Step 17. Finding count of Null values in Email column

```
df['Email'].isnull().sum()
```

Step 18. Next,checking on email

Using regular expression re module,

Step 19. Declare a pattern using re.compile() method

Step 20. Adding a new field ismail to the dataset for save email is valid or invalid.

Step 21. Based on values in ismail filed(True/False) ,dropping rows with invalid email

```
import re

import pandas as pd

df=pd.read_csv("C:\\Users\\SOUMYA\\Desktop\\latest.csv")

pattern=re.compile(r"(^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]
+$)")  # this is the regex expression to search on

df['ismail'] = df['Email'].apply(lambda x: True if pattern.match(x) else
False)

df.drop(df1[df1['ismail'] == False].index, inplace=True)
```

Step 22. Removing rows that contains Null in Department field

Step 23. Filling null values in Department field

```
df['Department']=df['Department'].fillna(method="ffill")
```

Step 24. Removing extra characters from Department name

```
original_department_names = ['Sales', 'Marketing', 'Support', 'HR',
'Engineering']

# Define function to clean department names

def clean_department_name(name):

    if isinstance(name, str):  # Check if name is a string (not NaN)

        for original_name in original_department_names:

            if name.endswith(original_name):

                return original_name

# If no exact match is found, find the longest matching original
department name
```

```
        longest_match = ' '
        for original_name in original_department_names:
            if original_name in name and len(original_name) > len(longest_match):
                longest_match = original_name

        if longest_match:
            return longest_match
        else:
            return name.strip('ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz')

# Apply the function to 'department' column
df['Department'] = df['Department'].apply(clean_department_name)
```

Step 25. Adding serial number to dataset

```
df1.insert(0, 'Sl_no', range(1, 1 + len(df1)))
```

Step 26. Exporting cleaned dataset

```
df1.to_csv("cleaned_dataset.csv")
```