Data preprocessing

Objective:

The main objective of this project is to design and implement a robust data preprocessing system that addresses common challenges such as missing values, outliers, inconsistent formatting, and noise. By performing effective data preprocessing, the project aims to enhance the quality, reliability, and usefulness of the data for machine learning

In [43]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [44]:

```
df=pd.read_csv('Employee.csv')
```

In [45]:

```
df
```

Out[45]:

	Company	Age	Salary	Place	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
143	TCS	33.0	9024.0	Calcutta	India	1
144	Infosys	22.0	8787.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

148 rows × 6 columns

In [46]:

```
df.describe
```

Out[46]:

```
<bound method NDFrame.describe of      Company   Age  Salary     Place Cou
ntry  Gender
0        TCS  20.0     NaN   Chennai   India       0
1    Infosys  30.0     NaN    Mumbai   India       0
2        TCS  35.0  2300.0  Calcutta   India       0
3    Infosys  40.0  3000.0     Delhi   India       0
4        TCS  23.0  4000.0    Mumbai   India       0
..       ...   ...     ...       ...     ...     ...
143      TCS  33.0  9024.0  Calcutta   India       1
144  Infosys  22.0  8787.0  Calcutta   India       1
145  Infosys  44.0  4034.0     Delhi   India       1
146      TCS  33.0  5034.0    Mumbai   India       1
147  Infosys  22.0  8202.0    Cochin   India       0

[148 rows x 6 columns]>
```

Data cleaning

In [47]:

```
df.drop_duplicates()
```

Out[47]:

	Company	Age	Salary	Place	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
142	Infosys Pvt Lmt	22.0	8202.0	Mumbai	India	0
143	TCS	33.0	9024.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

144 rows × 6 columns

In [48]:

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

Out[48]:

```
Company     8
Age        18
Salary     24
Place      14
Country     0
Gender      0
dtype: int64
```

In [49]:

```
df.Age.mean()
```

Out[49]:

```
30.484615384615385
```

In [50]:

```
df["Age"].replace(0.0,np.nan,inplace=True)
```

In [51]:

```
df["Age"]=df["Age"].fillna(30)
```

In [52]:

```
df
```

Out[52]:

	Company	Age	Salary	Place	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
143	TCS	33.0	9024.0	Calcutta	India	1
144	Infosys	22.0	8787.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

148 rows × 6 columns

In [53]:

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

Out[53]:

```
Company     8
Age         0
Salary     24
Place      14
Country     0
Gender      0
dtype: int64
```

Outlayer removel for Age

In [54]:

```
min=df.Age.quantile(0.1)
min
```

Out[54]:

```
22.0
```

In [55]:

```
max=df.Age.quantile(0.999)
max
```

Out[55]:

```
53.559000000000026
```

In [56]:

```
age_outlayer=df[(df.Age<min)|(df.Age>max)]
age_outlayer
```

Out[56]:

	Company	Age	Salary	Place	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
13	CTS	18.0	1234.0	Mumbai	India	0
22	TCS	21.0	4824.0	Mumbai	India	0
31	CTS	20.0	2934.0	Mumbai	India	0
49	CTS	19.0	1234.0	Cochin	India	0
52	Infosys	21.0	3030.0	Calcutta	India	0
54	TCS	21.0	6544.0	Mumbai	India	0
67	Congnizant	21.0	2934.0	Mumbai	India	0
70	Infosys Pvt Lmt	21.0	8202.0	Chennai	India	0
85	CTS	17.0	1234.0	Calcutta	India	0
87	TCS	21.0	3000.0	Mumbai	India	0
90	TCS	21.0	NaN	Mumbai	India	0
93	Infosys	54.0	3184.0	Mumbai	India	0
126	TCS	20.0	5009.0	NaN	India	1
130	TCS	21.0	4824.0	Mumbai	India	0

7 outlayers found for age

In [57]:

```
df1=df[(df.Age>min)&(df.Age<max)]
df1
```

Out[57]:

	Company	Age	Salary	Place	Country	Gender
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
5	Infosys	30.0	5000.0	Calcutta	India	0
...	...	...	...	...	...	...
140	Infosys	44.0	4034.0	Hyderabad	India	0
141	TCS	33.0	5034.0	Calcutta	India	0
143	TCS	33.0	9024.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1

117 rows × 6 columns

In [58]:

```
df1.Salary.describe()
```

Out[58]:

```
count      99.000000
mean     5320.101010
std      2531.199673
min      1089.000000
25%      3045.000000
50%      5034.000000
75%      7654.000000
max      9876.000000
Name: Salary, dtype: float64
```

In [59]:

```
df1.Salary.mean()
```

Out[59]:

```
5320.10101010101
```

In [60]:

```
df1["Salary"].fillna(5609, inplace=True)
```

```
C:\Users\hp\Anaconda3\lib\site-packages\pandas\core\generic.py:6287: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._update_inplace(new_data)
```

Outlayer detection in salary

In [61]:

```
q1=df1.Salary.quantile(0.01)
q1
```

Out[61]:

```
1089.0
```

In [62]:

```
q2=df1.Salary.quantile(0.99)
q2
```

Out[62]:

```
9781.280000000002
```

In [63]:

```
iqr=q2-q1
iqr
```

Out[63]:

```
8692.280000000002
```

In [64]:

```
min=q1-1.5*iqr
max=q2+1.5*iqr
min,max
```

Out[64]:

```
(-11949.420000000004, 22819.700000000004)
```

In [65]:

```
df2=df1[(df1["Salary"]>min)&(df1["Salary"]<max)]
df2
```

Out[65]:

	Company	Age	Salary	Place	Country	Gender
1	Infosys	30.0	5609.0	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
5	Infosys	30.0	5000.0	Calcutta	India	0
...	...	...	...	...	...	...
140	Infosys	44.0	4034.0	Hyderabad	India	0
141	TCS	33.0	5034.0	Calcutta	India	0
143	TCS	33.0	9024.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1

117 rows × 6 columns

No outlayers detected for salary

In [66]:

```
df2["Company"].unique()
```

Out[66]:

```
array(['Infosys', 'TCS', 'CTS', nan, 'Tata Consultancy Services'],
      dtype=object)
```

In [67]:

```
df2.replace("Infosys Pvt Lmt",'Infosys',inplace=True)
```

In [68]:

```
df2.replace('TCS', 'Tata Consultancy Services',inplace=True)
```

In [69]:

```
df2.Company.mode()
```

Out[69]:

```
0    Tata Consultancy Services
dtype: object
```

In [70]:

```
df2.Company.fillna((df2["Company"].mode()[0]),inplace=True)
```

In [71]:

```
df2["Place"].mode()
```

Out[71]:

```
0    Mumbai
dtype: object
```

In [72]:

```
df2.Place.fillna((df2["Place"].mode()[0]),inplace=True)
```

In [73]:

```
df2.isnull().sum()
```

Out[73]:

```
Company    0
Age        0
Salary     0
Place      0
Country    0
Gender     0
dtype: int64
```

In [74]:

```
df2
```

Out[74]:

	Company	Age	Salary	Place	Country	Gender
1	Infosys	30.0	5609.0	Mumbai	India	0
2	Tata Consultancy Services	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	Tata Consultancy Services	23.0	4000.0	Mumbai	India	0
5	Infosys	30.0	5000.0	Calcutta	India	0
...	...	...	...	...	...	...
140	Infosys	44.0	4034.0	Hyderabad	India	0
141	Tata Consultancy Services	33.0	5034.0	Calcutta	India	0
143	Tata Consultancy Services	33.0	9024.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	Tata Consultancy Services	33.0	5034.0	Mumbai	India	1

117 rows × 6 columns

Data Analysis

Filter the data with age >40 and salary<5000

In [75]:

```
filtered_df=df2[(df2["Age"]>40) & (df2["Age"]<5000)]
filtered_df
```

Out[75]:

	Company	Age	Salary	Place	Country	Gender
9	CTS	45.0	9000.0	Delhi	India	0
12	CTS	45.0	5609.0	Chennai	India	0
21	Infosys	50.0	3184.0	Delhi	India	0
27	CTS	45.0	9284.0	Delhi	India	1
30	CTS	46.0	7654.0	Chennai	India	0
32	Infosys	45.0	4034.0	Calcutta	India	0
39	Infosys	41.0	3000.0	Mumbai	India	0
45	CTS	46.0	9000.0	Hyderabad	India	1
48	CTS	43.0	5609.0	Mumbai	India	0
50	Infosys	41.0	3000.0	Chennai	India	0
57	Infosys	51.0	3184.0	Hyderabad	India	0
63	CTS	41.0	9284.0	Mumbai	India	1
66	CTS	41.0	5609.0	Calcutta	India	0
68	Infosys	43.0	4034.0	Mumbai	India	0
75	Infosys	44.0	3000.0	Cochin	India	0
81	CTS	43.0	9000.0	Pune	India	1
84	CTS	43.0	5609.0	Mumbai	India	0
86	Infosys	41.0	3000.0	Delhi	India	0
99	CTS	44.0	9284.0	Podicherry	India	1
102	CTS	44.0	5609.0	Mumbai	India	0
104	Infosys	44.0	4034.0	Delhi	India	0
117	CTS	44.0	9876.0	Mumbai	India	1
120	CTS	44.0	5609.0	Hyderabad	India	0
122	Infosys	44.0	3234.0	Mumbai	India	0
129	Infosys	50.0	3184.0	Calcutta	India	0
138	CTS	44.0	3033.0	Cochin	India	0
140	Infosys	44.0	4034.0	Hyderabad	India	0
145	Infosys	44.0	4034.0	Delhi	India	1

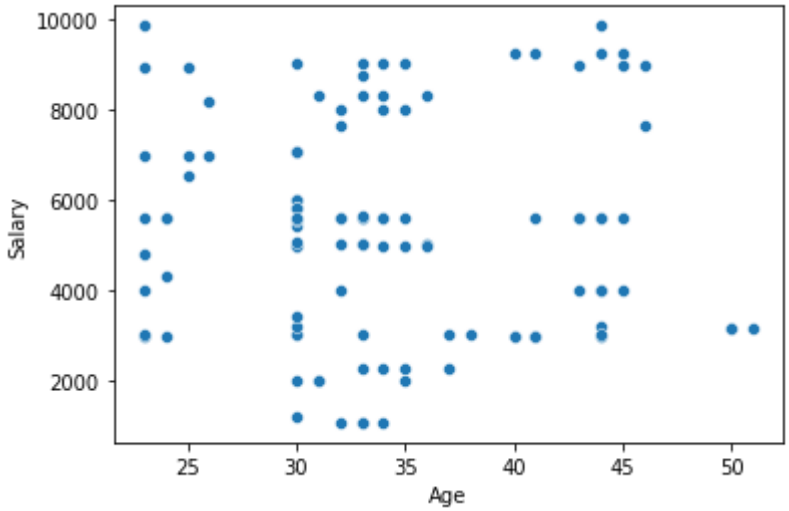
Plot the chart with age and salary

In [76]:

```
sns.scatterplot(data=df2,x='Age',y='Salary')
```

Out[76]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b5050f6ec8>
```



There is no correlation between Age and Salary

Count the number of people from each place and represent it visually

In [77]:

```
df2.Place.value_counts()
```

Out[77]:

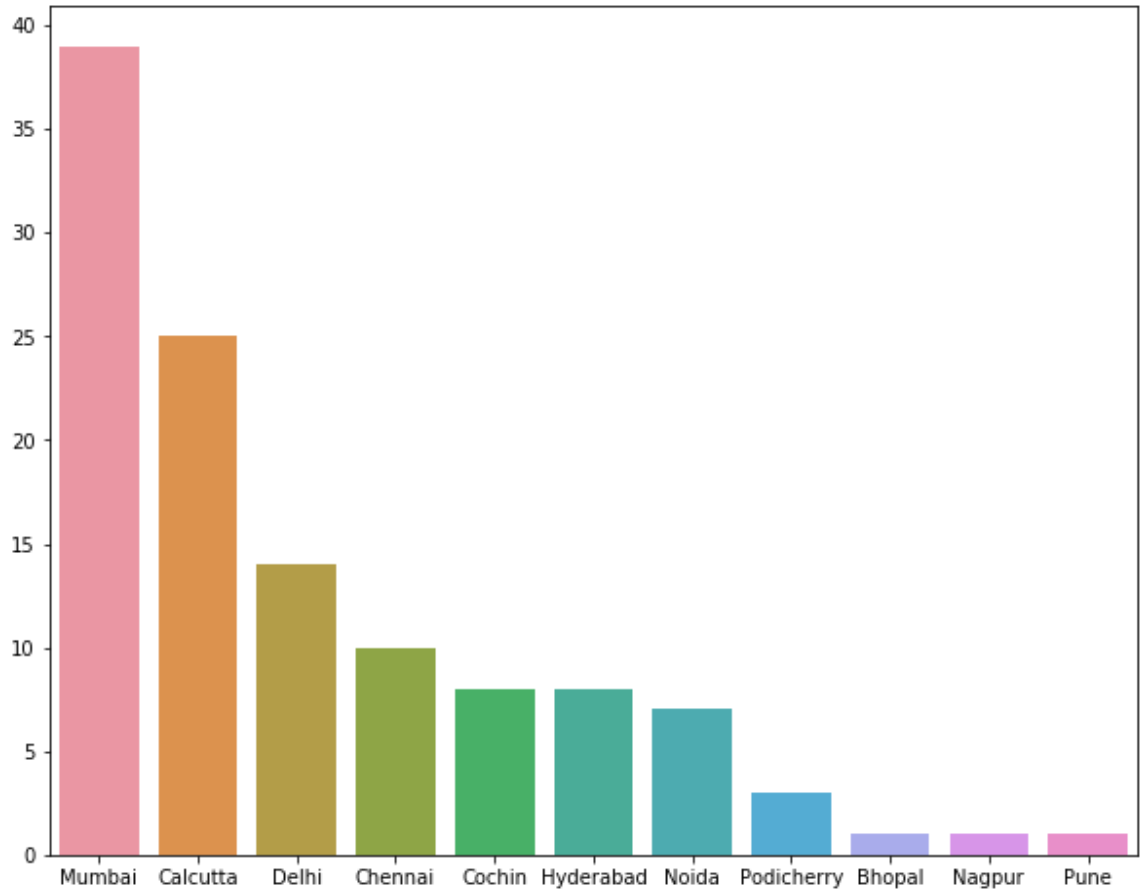
```
Mumbai        39
Calcutta      25
Delhi         14
Chennai       10
Cochin         8
Hyderabad      8
Noida          7
Podicherry     3
Bhopal         1
Nagpur         1
Pune           1
Name: Place, dtype: int64
```

In [78]:

```
plt.figure(figsize=(10,8))
sns.barplot(x=df2.Place.value_counts().index,y=df2.Place.value_counts().values)
```

Out[78]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b505125708>
```



Data Encoding:

Convert categorical variables into numerical representations using techniques such as one-hot encoding, label encoding, making them suitable for analysis by machine learning algorithms.

In [79]:

```
from sklearn.preprocessing import OneHotEncoder,LabelEncoder
```

Label Encoding of Company

In [80]:

```
label_encoder = LabelEncoder()
df3=df2
df3['Company'] = label_encoder.fit_transform(df2['Company'])
df3
```

Out[80]:

	Company	Age	Salary	Place	Country	Gender
1	1	30.0	5609.0	Mumbai	India	0
2	2	35.0	2300.0	Calcutta	India	0
3	1	40.0	3000.0	Delhi	India	0
4	2	23.0	4000.0	Mumbai	India	0
5	1	30.0	5000.0	Calcutta	India	0
...	...	...	...	...	...	...
140	1	44.0	4034.0	Hyderabad	India	0
141	2	33.0	5034.0	Calcutta	India	0
143	2	33.0	9024.0	Calcutta	India	1
145	1	44.0	4034.0	Delhi	India	1
146	2	33.0	5034.0	Mumbai	India	1

117 rows × 6 columns

OneHotEncoding of country,Company,Place

In [81]:

```
ohe_encoded_data= pd.get_dummies(df2, columns=['Country','Company','Place'])
ohe_encoded_data
```

Out[81]:

	Age	Salary	Gender	Country_India	Company_0	Company_1	Company_2	Place_Bhopa
1	30.0	5609.0	0	1	0	1	0	
2	35.0	2300.0	0	1	0	0	1	
3	40.0	3000.0	0	1	0	1	0	
4	23.0	4000.0	0	1	0	0	1	
5	30.0	5000.0	0	1	0	1	0	
...	...	...	...	...	...	...	...	.
140	44.0	4034.0	0	1	0	1	0	
141	33.0	5034.0	0	1	0	0	1	
143	33.0	9024.0	1	1	0	0	1	
145	44.0	4034.0	1	1	0	1	0	
146	33.0	5034.0	1	1	0	0	1	

117 rows × 18 columns

Feature Scaling:

After the process of encoding, perform the scaling of the features using standardscaler and minmaxscaler

StandardScaler

In [82]:

```
from sklearn.preprocessing import StandardScaler,MinMaxScaler
```

In [83]:

```
std_scalar=StandardScaler()
ohe_encoded_data[['Age', 'Salary']]=std_scalar.fit_transform(ohe_encoded_data[['Age',
'Salary']])
ohe_encoded_data
```

Out[83]:

	Age	Salary	Gender	Country_India	Company_0	Company_1	Company_2	Plac
1	-0.603365	0.105417	0	1	0	1	0	
2	0.124405	-1.321541	0	1	0	0	1	
3	0.852175	-1.019676	0	1	0	1	0	
4	-1.622242	-0.588441	0	1	0	0	1	
5	-0.603365	-0.157206	0	1	0	1	0	
...	...	...	...	...	...	...	...	
140	1.434390	-0.573779	0	1	0	1	0	
141	-0.166703	-0.142544	0	1	0	0	1	
143	-0.166703	1.578085	1	1	0	0	1	
145	1.434390	-0.573779	1	1	0	1	0	
146	-0.166703	-0.142544	1	1	0	0	1	

117 rows × 18 columns

MinMaxScaler

In [84]:

```
MM_scalar=MinMaxScaler()
ohe_encoded_data[['Age', 'Salary']]=MM_scalar.fit_transform(ohe_encoded_data[['Age', 'Salary']])
ohe_encoded_data
```

Out[84]:

	Age	Salary	Gender	Country_India	Company_0	Company_1	Company_2	Place_
1	0.250000	0.514396	0	1	0	1	0	
2	0.428571	0.137817	0	1	0	0	1	
3	0.607143	0.217480	0	1	0	1	0	
4	0.000000	0.331285	0	1	0	0	1	
5	0.250000	0.445089	0	1	0	1	0	
...	...	...	...	...	...	...	...	...
140	0.750000	0.335154	0	1	0	1	0	
141	0.357143	0.448959	0	1	0	0	1	
143	0.357143	0.903039	1	1	0	0	1	
145	0.750000	0.335154	1	1	0	1	0	
146	0.357143	0.448959	1	1	0	0	1	

117 rows × 18 columns

In [ ]: