Introduction:-

This report presents a comprehensive analysis of the given employee dataset. The analysis is structured into several key steps, including data exploration, cleaning, statistical analysis, visualization, encoding, and feature scaling. Each step is essential for preparing the data for further insights and machine learning applications.

- Data Exploration – Understanding the dataset by listing unique values and performing statistical analysis.
- Data Cleaning – Handling missing, inappropriate, and duplicate values, as well as detecting outliers.
- Data Analysis – Filtering specific data points and visualizing key relationships.
- Data Encoding – Converting categorical variables into numerical representations.
- Feature Scaling – Normalizing numerical values for better model performance.

Code:-

Q1. Explore the data, list down the unique values in each feature and find its length.

Perform the statistical analysis and renaming of the columns.

Q2. Find the missing and inappropriate values, treat them appropriately.

Remove all duplicate rows.

Find the outliers.

Replace the value 0 in age as NaN

Treat the null values in all columns using any measures(removing/ replace the values with mean/median/mode)

Q3. Filter the data with age >40 and salary<5000

Plot the chart with age and salary

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

Q4. Convert categorical variables into numerical representations using techniques such as one-hot encoding, label encoding, making them suitable for analysis by machine learning algorithms.

Q5. After the process of encoding, perform the scaling of the features using standardscaler and minmaxscaler.

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

# Load the dataset
file_path = "C:/Users/sayan/Desktop/New folder/Employee.csv"
df = pd.read_csv(file_path)
```

```
# 1. Data Exploration
print("Dataset Info:\n", df.info())
print("\nUnique Values in Each Column:\n", {col: df[col].nunique() for col in df.columns})

# 2. Data Cleaning
df["Age"].replace(0, np.nan, inplace=True)  # Replace 0 in Age with NaN

# Standardizing Company Names
company_replacements = {
    "Tata Consultancy Services": "TCS",
    "Congnizant": "CTS",
    "Infosys Pvt Lmt": "Infosys"
}
df["Company"] = df["Company"].replace(company_replacements)

# Remove duplicate rows
df.drop_duplicates(inplace=True)

# Fill missing numerical values with median
df["Age"].fillna(df["Age"].median(), inplace=True)
df["Salary"].fillna(df["Salary"].median(), inplace=True)

# Fill missing categorical values with mode
df["Company"].fillna(df["Company"].mode()[0], inplace=True)
df["Place"].fillna(df["Place"].mode()[0], inplace=True)

print("\nMissing Values After Cleaning:\n", df.isnull().sum())

# 3. Outlier Detection using Boxplots
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.boxplot(df["Age"].dropna(), vert=False)
plt.title("Boxplot of Age")

plt.subplot(1, 2, 2)
plt.boxplot(df["Salary"].dropna(), vert=False)
plt.title("Boxplot of Salary")

plt.show()

# 4. Data Filtering (Age > 40 and Salary < 5000)
filtered_df = df[(df["Age"] > 40) & (df["Salary"] < 5000)]
print("\nFiltered Data:\n", filtered_df.head())

# 5. Data Visualization
plt.figure(figsize=(8, 5))
plt.scatter(df["Age"], df["Salary"], alpha=0.5)
plt.xlabel("Age")
plt.ylabel("Salary")
plt.title("Age vs Salary Distribution")
plt.grid(True)
```

```
plt.show()

# Count of people from each place
place_counts = df["Place"].value_counts()
plt.figure(figsize=(10, 5))
plt.barh(place_counts.index, place_counts.values)
plt.xlabel("Count")
plt.ylabel("Place")
plt.title("Number of People from Each Place")
plt.grid(True, linestyle="--", alpha=0.7)
plt.show()

# 6. Manual Encoding (Convert categorical variables into numerical values)
def manual_label_encoding(column):
    unique_vals = column.unique()
    mapping = {val: idx for idx, val in enumerate(unique_vals)}
    return column.map(mapping)

df["Company"] = manual_label_encoding(df["Company"])
df["Place"] = manual_label_encoding(df["Place"])

print("\nData After Encoding:\n", df.head())

# 7. Manual Feature Scaling
# Standardization: (x - mean) / std
df["Age_Standardized"] = (df["Age"] - df["Age"].mean()) / df["Age"].std()
df["Salary_Standardized"] = (df["Salary"] - df["Salary"].mean()) / df["Salary"].std()

# Min-Max Scaling: (x - min) / (max - min)
df["Age_MinMax"] = (df["Age"] - df["Age"].min()) / (df["Age"].max() - df["Age"].min())
df["Salary_MinMax"] = (df["Salary"] - df["Salary"].min()) / (df["Salary"].max() - df["Salary"].min())

print("\nData After Scaling:\n", df.head())
```

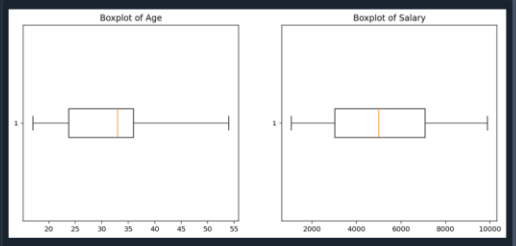
Spyder

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\sayan\.spyder-py3\main_program.py

main_program.py

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

# Load the dataset
file_path = "C:/Users/sayan/Desktop/New folder/Employee.csv"
df = pd.read_csv(file_path)

# 1. Data Exploration
print("Dataset Info:\n", df.info())
print("\nUnique Values in Each Column:\n", {col: df[col].nunique() for col in df.columns})

# 2. Data Cleaning
df["Age"].replace(0, np.nan, inplace=True)  # Replace 0 in Age with NaN

# Standardizing Company Names
company_replacements = {
    "Tata Consultancy Services": "TCS",
    "Congnizant": "CTS",
    "Infosys Pvt Lmt": "Infosys"
}
df["Company"] = df["Company"].replace(company_replacements)

# Remove duplicate rows
df.drop_duplicates(inplace=True)

# Fill missing numerical values with median
df["Age"].fillna(df["Age"].median(), inplace=True)
df["Salary"].fillna(df["Salary"].median(), inplace=True)

# Fill missing categorical values with mode
df["Company"].fillna(df["Company"].mode()[0], inplace=True)
df["Place"].fillna(df["Place"].mode()[0], inplace=True)

print("\nMissing Values After Cleaning:\n", df.isnull().sum())

# 3. Outlier Detection using Boxplots
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
```



Help Debugger Plots Files

Console 3/A

```
21  Infosys  50.0  3184.0      Delhi   India       0
32  Infosys  45.0  4034.0   Calcutta   India       0
39  Infosys  41.0  3000.0     Mumbai   India       0
50  Infosys  41.0  3000.0    Chennai   India       0
57  Infosys  51.0  3184.0  Hyderabad   India       0

Data After Encoding:
    Company   Age  Salary  Place Country  Gender
0         0  20.0  5000.0      0   India       0
1         1  30.0  5000.0      1   India       0
2         0  35.0  2300.0      2   India       0
3         1  40.0  3000.0      3   India       0
4         0  23.0  4000.0      1   India       0

In [2]:
```

IPython Console History

Inline Conda: spyder-runtime (Python 3.11.11) LSP: Python Line 6, Col 17 UTF-8 CRLF RW Mem 84%

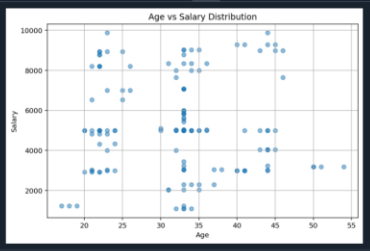
Spyder

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\sayan\.spyder-py3\main_program.py

main_program.py

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

# Load the dataset
file_path = "C:/Users/sayan/Desktop/New folder/Employee.csv"
df = pd.read_csv(file_path)

# 1. Data Exploration
print("Dataset Info:\n", df.info())
print("\nUnique Values in Each Column:\n", {col: df[col].nunique() for col in df.columns})

# 2. Data Cleaning
df["Age"].replace(0, np.nan, inplace=True)  # Replace 0 in Age with NaN

# Standardizing Company Names
company_replacements = {
    "Tata Consultancy Services": "TCS",
    "Congnizant": "CTS",
    "Infosys Pvt Lmt": "Infosys"
}
df["Company"] = df["Company"].replace(company_replacements)

# Remove duplicate rows
df.drop_duplicates(inplace=True)

# Fill missing numerical values with median
df["Age"].fillna(df["Age"].median(), inplace=True)
df["Salary"].fillna(df["Salary"].median(), inplace=True)

# Fill missing categorical values with mode
df["Company"].fillna(df["Company"].mode()[0], inplace=True)
df["Place"].fillna(df["Place"].mode()[0], inplace=True)

print("\nMissing Values After Cleaning:\n", df.isnull().sum())

# 3. Outlier Detection using Boxplots
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
```



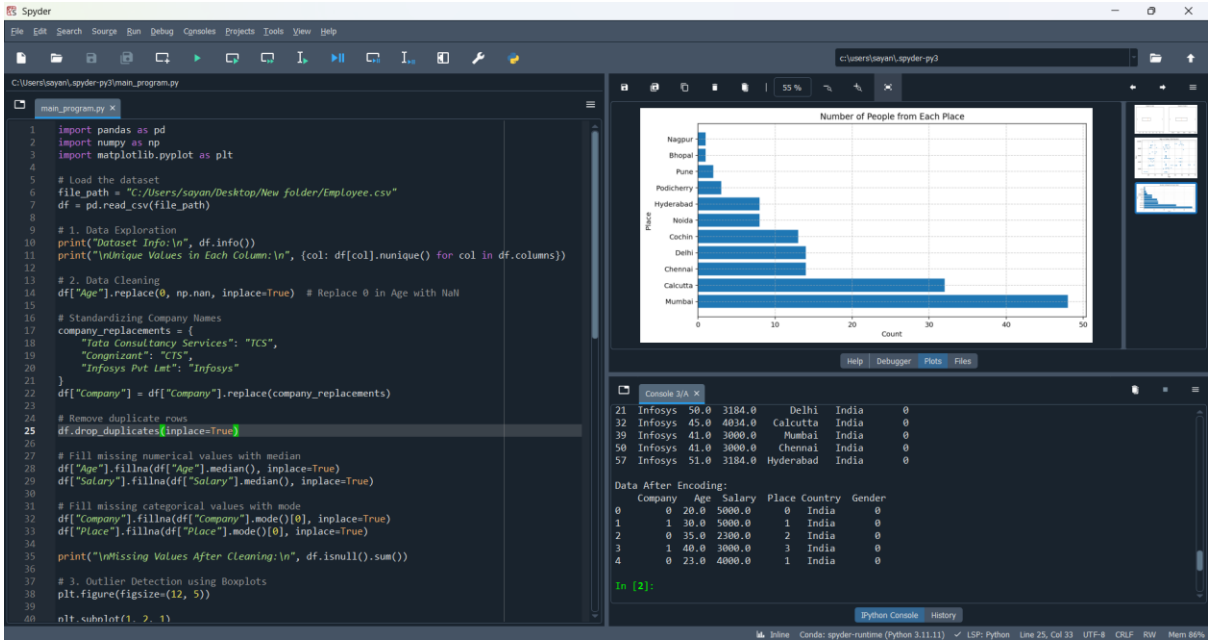
Help Debugger Plots Files

Console 3/A

```
21  Infosys  50.0  3184.0      Delhi   India       0
32  Infosys  45.0  4034.0   Calcutta   India       0
39  Infosys  41.0  3000.0     Mumbai   India       0
50  Infosys  41.0  3000.0    Chennai   India       0
57  Infosys  51.0  3184.0  Hyderabad   India       0

Data After Encoding:
    Company   Age  Salary  Place Country  Gender
0         0  20.0  5000.0      0   India       0
1         1  30.0  5000.0      1   India       0
2         0  35.0  2300.0      2   India       0
3         1  40.0  3000.0      3   India       0
4         0  23.0  4000.0      1   India       0

In [2]:
```

IPython Console History

Inline Conda: spyder-runtime (Python 3.11.11) LSP: Python Line 6, Col 17 UTF-8 CRLF RW Mem 87%


Spyder
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\sayan\.spyder-py3\main_program.py
main_program.py
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Load the dataset
file_path = "C:/Users/sayan/Desktop/New folder/Employee.csv"
df = pd.read_csv(file_path)
# 1. Data Exploration
print("Dataset Info:\n", df.info())
print("\nUnique Values in Each Column:\n", {col: df[col].nunique() for col in df.columns})
# 2. Data Cleaning
df["Age"].replace(0, np.nan, inplace=True) # Replace 0 in Age with NaN
# Standardizing Company Names
company_replacements = {
"Tata Consultancy Services": "TCS",
"Congnizant": "CTS",
"Infosys Pvt Lmt": "Infosys"
}
df["Company"] = df["Company"].replace(company_replacements)
# Remove duplicate rows
df.drop_duplicates(inplace=True)
# Fill missing numerical values with median
df["Age"].fillna(df["Age"].median(), inplace=True)
df["Salary"].fillna(df["Salary"].median(), inplace=True)
Fill missing categorical values with mode
df["Company"].fillna(df["Company"].mode()[0], inplace=True)
df["Place"].fillna(df["Place"].mode()[0], inplace=True)
print("\nMissing Values After Cleaning:\n", df.isnull().sum())
# 3. Outlier Detection using Boxplots
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
Number of People from Each Place
Nagpur
Bhopal
Pune
Podicherry
Hyderabad
Noida
Cochin
Delhi
Chennai
Calcutta
Mumbai
Place
0
10
20
30
40
50
Count
Help Debugger Plots Files
Console 3/A
21 Infosys 50.0 3184.0 Delhi India 0
32 Infosys 45.0 4034.0 Calcutta India 0
39 Infosys 41.0 3000.0 Mumbai India 0
50 Infosys 41.0 3000.0 Chennai India 0
57 Infosys 51.0 3184.0 Hyderabad India 0
Data After Encoding:
Company Age Salary Place Country Gender
0 0 20.0 5000.0 0 India 0
1 1 30.0 5000.0 1 India 0
2 0 35.0 2300.0 2 India 0
3 1 40.0 3000.0 3 India 0
4 0 23.0 4000.0 1 India 0
In [2]:
IPython Console History
Inline Conda: spyder-runtime (Python 3.11.11) LSP: Python Line 25, Col 33 UTF-8 CRLF RW Mem 86%