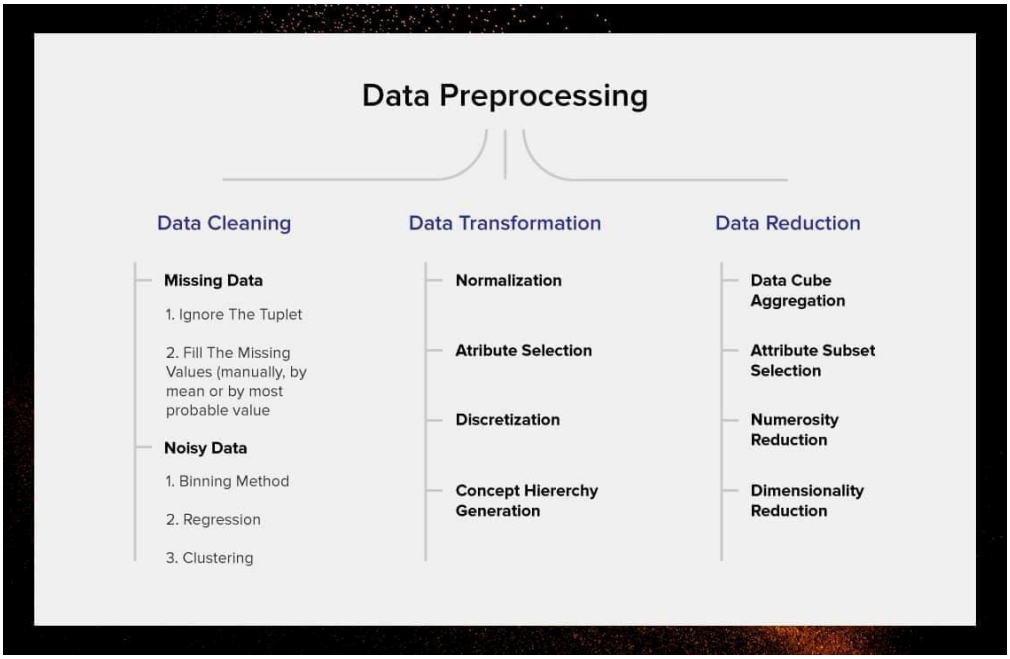
EXPERIMENT NO.3

AIM: To preprocess dataset using different preprocessing techniques.

THEORY:
CASE STUDY: Analyzing and understanding the factors influencing Financing.

PROBLEM STATEMENT:
An online retail company has broadened its offerings to include a Digital Transaction feature, enabling customers to store credit and debit transaction details for purchases made on the platform. This Digital Wallet supports various transaction types, including deposits, withdrawals, and purchases, across different account types such as Savings, Checking, and Credit.

Introduction:
Data preprocessing is an important step in the data mining process. It refers to the cleaning,transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.
Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.



Importance of Data Preprocessing:

Preprocessing data is an important step for data analysis. The following are some benefits of preprocessing data:

● It improves accuracy and reliability. Preprocessing data removes missing or inconsistent data values resulting from human or computer error, which can improve the accuracy and quality of a dataset, making it more reliable.

● It makes data consistent. When collecting data, it's possible to have data duplicates, and discarding them during preprocessing can ensure the data values for analysis are consistent, which helps produce accurate results.

● It increases the data's algorithm readability. Preprocessing enhances the data's quality and makes it easier for machine learning algorithms to read, use, and interpret it.

Features of Data Preprocessing:

Preprocessing has many features that make it an important preparation step for data analysis. The following are the two main features with a brief explanation:

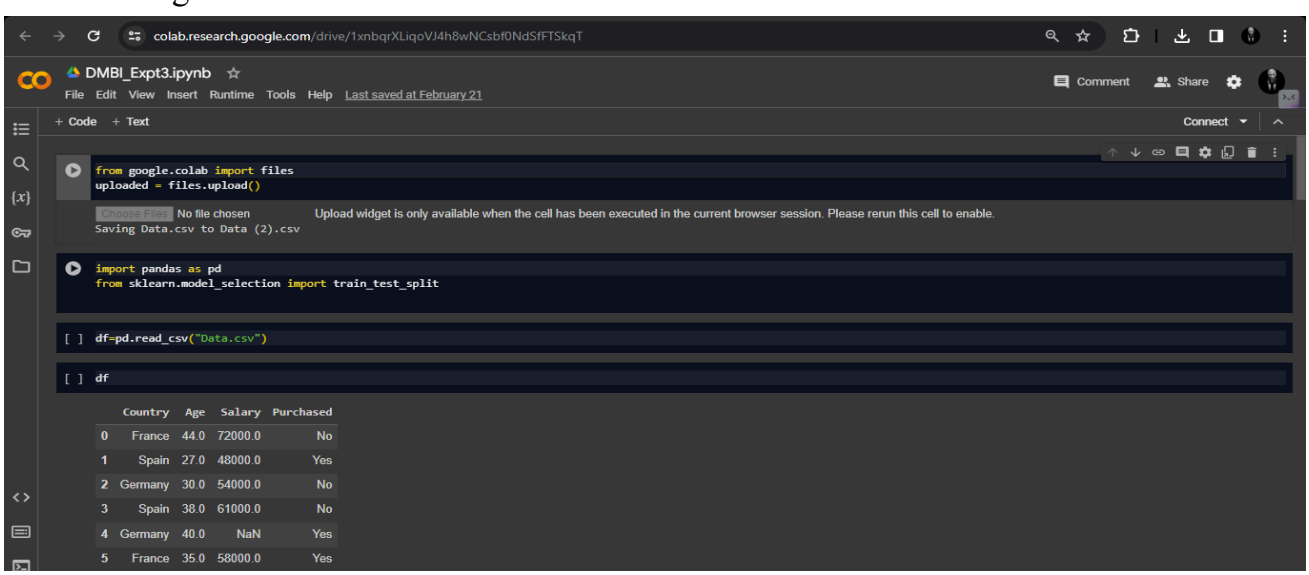
● Data validation: This is the process where businesses analyze and assess the raw data for a project to determine if it's complete and accurate to achieve the best results.

● Data imputation: Data imputation is where you input missing values and rectify data errors during the validation process manually or through programming, like business process automation.

Output:

Data Cleaning - removing missing values (demonstrate removing and replacing Null values)

Data Loading:



Data Description:

```
df.describe()
```

	Age	Salary
count	9.000000	9.000000
mean	38.777778	63777.777778
std	7.693793	12265.579662
min	27.000000	48000.000000
25%	35.000000	54000.000000
50%	38.000000	61000.000000
75%	44.000000	72000.000000
max	50.000000	83000.000000

```
df.head()
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes

Next steps: View recommended plots

Dropping Columns:

```
[6] cols = ['Age']
    df= df.drop(cols,axis=1)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Country    10 non-null     object
 1   Salary     9 non-null      float64
 2   Purchased  10 non-null     object
dtypes: float64(1), object(2)
memory usage: 368.0+ bytes
```

Fill missing values:

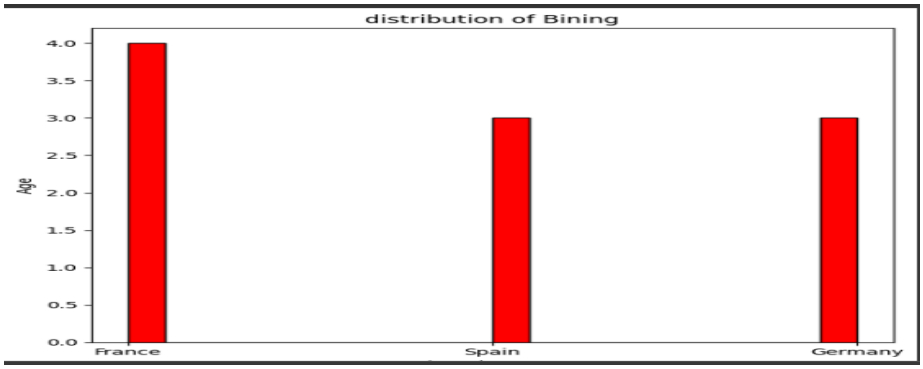
```
[ ] df.isna().sum()

    Country      0
    Age          1
    Salary       1
    Purchased    0
    dtype: int64
```

Drop rows with missing values:

```
[8] df['Country']= df['Country'].interpolate()

    df.isna().sum()

    Country      0
    Salary       1
    Purchased    0
    dtype: int64
```

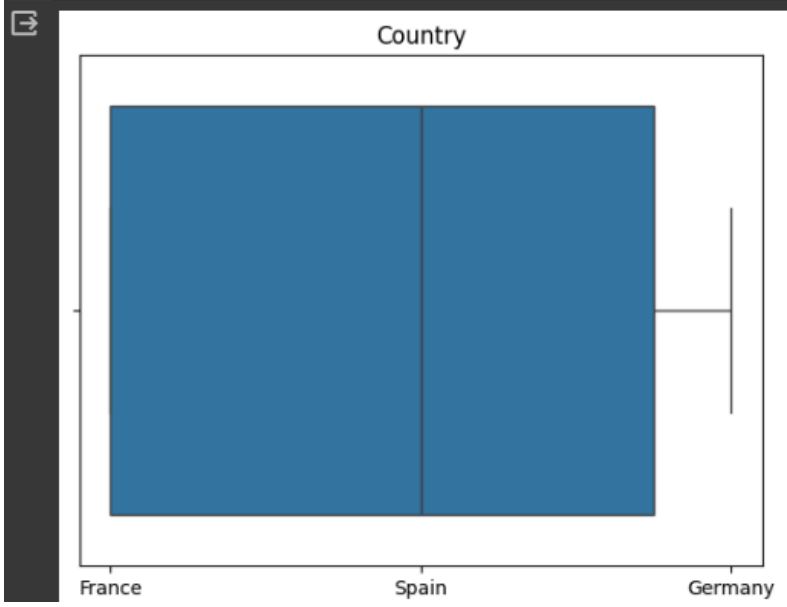
Data Cleaning - removing noisy values (Binning technique), removing outliers-Interquartile Range Method,Boxplot

```
[20] import matplotlib.pyplot as plt
     import seaborn as sns

     plt.figure(figsize=(12,6))
     plt.subplot(1,2,1)
     plt.hist(df['Country'],bins=20,color='red', edgecolor='black')
     plt.title('distribution of Bining')

     plt.xlabel('Binned Country')
     plt.ylabel('Age')

     plt.tight_layout()
     plt.show()
```



Removing outliers using IQR technique:

```
sns.boxplot(x='country',data = df_no_outliners)
plt.title('Box plot after Otliners Removed')
plt.show()
```



Data Transformation - converting numerical attributes to categorical and vice versa/ one hot encoding

```
[ ] from sklearn.preprocessing import LabelEncoder

    l1=LabelEncoder()

    df['Country']=l1.fit_transform(df['Country'])
    df['Purchased']=l1.fit_transform(df['Purchased'])
```

```
print("\nDataFrame after label encoding:")
print(df)
```

```
DataFrame after label encoding:
   Country        Age        Salary  Purchased
0        0  44.000000  72000.000000          0
1        2  27.000000  48000.000000          1
2        1  30.000000  54000.000000          0
3        2  38.000000  61000.000000          0
4        1  40.000000  63777.777778          1
5        0  35.000000  58000.000000          1
6        2  38.777778  52000.000000          0
7        0  48.000000  79000.000000          1
8        1  50.000000  83000.000000          0
9        0  37.000000  67000.000000          1
```

Data Normalization:

```
+ Code  + Text

[21] df_normalized

array([[0.        , 0.13380282, 0.        , 0.43414873, 0.84049976,
        0.40511788],
       [0.        , 0.36619718, 1.        , 0.39696673, 0.84194103,
        0.40511788],
       [0.        , 0.59859155, 0.        , 0.41086106, 0.84240534,
        0.40511788],
       ...,
       [1.        , 0.38732394, 0.        , 0.55313112, 0.65497724,
        0.45888442],
       [1.        , 0.61971831, 0.        , 0.57270059, 0.65501271,
        0.45888442],
       [1.        , 0.85211268, 0.        , 0.59598826, 0.65479569,
        0.45888442]])
```

Data Reduction - reducing the number of rows by attribute-oriented induction or numerosity reduction

```
+ Code  + Text

[24] from sklearn.cluster import KMeans
     kmeans = KMeans(n_clusters=100)
     kmeans.fit(df)
     centroids = kmeans.cluster_centers_

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` exp
  warnings.warn(

[25] # Example of numerosity reduction using sampling
     sampled_df = df.sample(frac=0.5, random_state=42)  # Sample 50% of the data
     sampled_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 3218 entries, 2436 to 0
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         3218 non-null   int64
 1   Date          3218 non-null   int64
 2   Holiday_Flag  3218 non-null   int64
 3   Temperature   3218 non-null   float64
 4   CPI           3218 non-null   float64
 5   Unemployment  3218 non-null   float64
dtypes: float64(3), int64(3)
memory usage: 176.0 KB
```

<u>Conclusion</u>: Hence, we conducted diverse preprocessing techniques on the dataset to enhance its quality and usability for subsequent analysis.