Pre processing Dataset

Country	Age	Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40	NaN	Yes
France	35	58000	Yes
Spain	NaN	52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes.

04-08-25

Expt No: 3 (a) - Experiment to understand the data preprocessing in Data science

Description: Understand the importance of Data pre-processing in data science.

Soln: Aim: To understand the importance of data preprocessing in data science by handling missing values, encoding categorical data and transforming the dataset into a machine learning ready format.

Algorithm:

1. Import Libraries

Import numpy and pandas for data manipulation.

2. Load Dataset

Read the dataset into a DataFrame

3. Check Missing Data

Use df.info() to identify missing values.

4. Handle Missing Values

Fill missing categorical data (country) with the mode, missing numerical data (age) with the median, missing numerical data (salary) with mean (rounded)

5. Encode Categorical Variables

- Convert Country into dummy variables using pd.get_dummies( ).

- Replace Purchased column values (Yes/No) with binary values (1/0).

6. Create Final Dataset.

Program:

```
import numpy as np
import pandas as pd
df = pd.read_csv(" /content/pre-process_datasample.cs
df['Country'].fillna(df['Country'].mode()
[0], inplace = True)
df['Age'].fillna(df['Age'].median(), inplace = Tru
df['Salary'].fillna(round(df['Salary'].mean()),
inplace = True)
country_dummies = pd.get_dummies(df['Country'])
updated_dataset = pd.concat([country_dummies,
df[['Age', 'Salary', 'Purchased']]], axis = 1)
updated_dataset['Purchased'].replace(['No', 'Yes'],
[0,1], inplace = True)
print(updated_dataset)
```

Output:

Country.	Age	Salary	Purchased
France	44	72000.0	No
Spain	27	48000.0	Yes
Germany	30	54000.0	No
Spain	38	61000.0	No
Germany	40	61000.0	Yes
France	35	58000.0	Yes
Spain	44	52000.0	No
France	48	79000.0	Yes
France	50	83000.0	No
France	37	67000.0	Yes

Result:

Thus the csv file is preprocessed and cleaned, saved again into the same file.

Dataset Given :

~~Hotel.csv~~

~~Customer ID~~ ~~Age Group~~ ~~Rating (1-5)~~ Hotel ~~Food~~ ~~Bill~~ ~~No of Pax~~

Customer ID	Age Group	Rating (1-5)	Hotel	Food Preference	Bill	No of Pax	Estimated Salary	Age Group
1	20-25	4	Ibis	veg	1300	2	40000	20-25
2	30-35	5	Lemon Tree	Non-veg	2000	3	59000	30-35
3	25-30	6	Red Fox	Veg	1322	2	30000	25-2
4	20-25	-1	Lemon Tree	Veg	1234	2	120000	25-30
5	35+	3	Ibis	Vegetarian	984	2	45000	20-25
6	35+	3	Ibis	Non-Veg	1909	2	122220	35+
7	35+	4	Red Fox	Vegetarian	1000	-1	21122	35+
8	20-25	7	Lemon Tree	Veg	2999	-10	345673	20-25
9	25-30	2	Ibis	Non-Veg	3456	3	-99999	25-30
10	30-35	5	Ibis	Non-Veg	3456	3	-99999	25-30
			Red Fox	non-Veg	6755	4	87777	30-35

11-08-25

Expt No: 3(b) - Handling Missing and Inappropriate Data in a Dataset.

Description: Demonstrate an experiment to handle missing data and inappropriate data in a Data set using Python Pandas Library for Data Preprocessing.

Soln: Aim: To demonstrate ~~and~~ how to detect and fix missing, duplicate and inappropriate values in a dataset.

Algorithm:

1. Load the dataset into a pandas Data Frame
2. Remove duplicate rows and redundant columns
3. Detect invalid values and mark them as missing. (eg - negative, out of range)
4. Standardize categorical values
5. Impute missing values: mean for continuous, median for discrete.
6. Verify cleaned dataset with unique values and basic statistics and data types.

Program:

```
import pandas as pd
import numpy as np
df = pd.read_csv ("Hotel - Dataset.csv")
df.drop_duplicates (inplace = True)
df.reset_index (drop = True, inplace = True)
df.loc [df.Customer ID < 0, 'Customer ID'] = np.nan
df.loc [df.Bill <0, 'Bill'] = np.nan
df.loc [df.Estimated salary <0, 'Estimated Salary'] = np.n
df.loc [(df.No of Pax < 1) | (df.No of Pax > 20), 'No Of Pa
= np.nan
df['Hotel'].replace ('Ibys', 'Ibis', inplace = True)
df['Food Preference'].replace (['Vegetarian', 'veg'], 'Veg
inplace = True)
df['Food Preference'].replace ('non-Veg', 'Non-Veg',
inplace = True)
df.fillna ({'Estimated Salary': round (df['Estimated Salary
.mean()), 'No of Pax': round (df['No Of Pax'].median())
'Rating (1-5)': round (df['Rating (1-5)'].median()),
'Bill': round (df['Bill'].mean())}, inplace = True)
print (df)
```

Result:

Thus the dataset was cleaned and pre processed by handling duplicates, invalid values and missing data using Pandas.

11-08-25

Expt No: 3(c) - Data Preprocessing - Create own dataset in CSV format and do data preprocessing.

Aim:

To ~~perform~~ preprocess a dataset and clean the data by handling missing values, encoding categorical variables.

Algorithm:

1. Create Dataset - Build a csv file with categorical values and numerical values.
2. Load dataset - Read the dataset using Pandas into a Data Frame.
3. Handle Missing Values
4. Encode categorical variables
5. Save the csv file

Program:

```
import pandas as pd
data = {"Country": ["India", "USA", "UK", "India",
"Germany", None, "UK", "USA", "India",
None], "Age": [25, 30, 22, None, 40, 35, None,
28, 32, 45], "Salary": [50000, 60000, None,
52000, 75000, 65000, 62000, None, 70000,
80000], "Purchased": ["Yes", "No", "Yes", "No",
None, "Yes", "No", "Yes", "Yes", "No"]}
df = pd.DataFrame(data)
df['Country'].fillna(df['Country'].mode()[0],
inplace = True)
df['Age'].fillna(df['Age'].median(), inplace = True)
df['Salary'].fillna(round(df['Salary'].mean()), inplace
= True)
df['Purchased'].fillna(df['Purchased'].mode()[0], inplace
= True)
country_dummies = pd.get_dummies(df['Country'])
updated_dataset = pd.concat([country_dummies,
df[['Age', 'Salary', 'Purchased']]], axis = 1)
updated_dataset['Purchased'].replace(['No', 'Yes'], [0, 1]
inplace = True)
print(updated_dataset)
```

Output :

Purchased	Germany	India	UK	USA	Age	Salary
~~No~~ 1	0	1	0	0	25.0	50000.0
~~Yes~~ 0	0	0	0	1	30.0	60000.0
~~Yes~~ 1	0	0	0	0	22.0	64625.0
0	0	1	0	0	31.0	52000.0
1	1	0	0	0	40.0	75000.0
1	0	1	0	0	35.0	68000.0
0	0	0	0	0	31.0	62000.0
1	0	0	0	1	25.0	64625.0
1	0	1	0	0	32.0	70000.0
0	0	1	0	0	45.0	80000.0

Result : Thus, the dataset is ~~cleaned~~, ~~co~~ imported. and cleaned and preprocessed using Pandas.