

Data preprocessing

Code

```
1  # Data preprocessing
2
3  # Importing libraries
4
5  import numpy as np
6  import pandas as pd
7  import matplotlib.pyplot as plt
8
9  # Import data and divide into dv & iv
10 dataset = pd.read_csv(r'D:\1. Professional\Data Science\08-09-2023\Data.csv')
11
12 X = dataset.iloc[:, :-1].values
13 y = dataset.iloc[:, 3].values
14
15 # Filling null values in X dataset
16
17 from sklearn.impute import SimpleImputer
18 # default strategy is mean
19 impute = SimpleImputer(missing_values=np.nan, strategy="median")
20
21 impute = impute.fit(X[:, 1:3])
22 X[:, 1:3] = impute.transform(X[:, 1:3])
23
24 # Convert categorical data into numerical data
25
26 from sklearn.preprocessing import LabelEncoder
27
28 labelencoder_X = LabelEncoder()
29 X[:, 0] = labelencoder_X.fit_transform(X[:, 0])
30
31
32 labelencoder_y = LabelEncoder()
33 y = labelencoder_y.fit_transform(y)
34
35 # Splitting data into training set and testing set
36
37 from sklearn.model_selection import train_test_split
38
39 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)
40
```

Dataset

dataset - DataFrame

	Index	State	Age	Salary	Purchased
0	Mumbai	44	72000	No	
1	Bangalore	27	48000	Yes	
2	Hyderabad	30	54000	No	
3	Bangalore	38	61000	No	
4	Hyderabad	40	nan	Yes	
5	Mumbai	35	58000	Yes	
6	Bangalore	nan	52000	No	
7	Mumbai	48	79000	Yes	
8	Hyderabad	50	83000	No	
9	Mumbai	37	67000	Yes	

Format

Resize

☒ Background color

☒ Column min/max

Save and Close

Close

X & y dataset

X - NumPy object array (read only)

	0	1	2
0	Mumbai	44.0	72000.0
1	Bangalore	27.0	48000.0
2	Hyderabad	30.0	54000.0
3	Bangalore	38.0	61000.0
4	Hyderabad	40.0	nan
5	Mumbai	35.0	58000.0
6	Bangalore	nan	52000.0
7	Mumbai	48.0	79000.0
8	Hyderabad	50.0	83000.0
9	Mumbai	37.0	67000.0

Format Resize ☐ Background color Close

y - NumPy object array (read only)

	0
0	No
1	Yes
2	No
3	No
4	Yes
5	Yes
6	No
7	Yes
8	No
9	Yes

Format Resize ☐ Background color Close

Simple imputed X dataset

X - NumPy object array (read only)

	0	1	2
0	Mumbai	44.0	72000.0
1	Bangalore	27.0	48000.0
2	Hyderabad	30.0	54000.0
3	Bangalore	38.0	61000.0
4	Hyderabad	40.0	61000.0
5	Mumbai	35.0	58000.0
6	Bangalore	38.0	52000.0
7	Mumbai	48.0	79000.0
8	Hyderabad	50.0	83000.0
9	Mumbai	37.0	67000.0

Format Resize ☐ Background color Close

Label encoded X & y

X - NumPy object array (read only)

	0	1	2
0	2	44.0	72000.0
1	0	27.0	48000.0
2	1	30.0	54000.0
3	0	38.0	61000.0
4	1	40.0	61000.0
5	2	35.0	58000.0
6	0	38.0	52000.0
7	2	48.0	79000.0
8	1	50.0	83000.0
9	2	37.0	67000.0

Format Resize ☐ Background color Close

y - NumPy object array

	0
0	0
1	1
2	0
3	0
4	1
5	1
6	0
7	1
8	0
9	1

Format Resize ☒ Background color Save and Close Close

Train test and split data

The image shows five Jupyter Notebook windows displaying NumPy arrays for training and testing data. The windows are titled 'X - NumPy object array (read only)', 'X_train - NumPy object array (read only)', 'X_test - NumPy object array (read only)', 'y_test - NumPy object array', and 'y_train - NumPy object array'. Each window displays a table of data with columns and rows, and a control bar at the bottom with buttons for Format, Resize, Background color, Save and Close, and Close.

X - NumPy object array (read only)

	0	1	2
0	2	44.0	72000.0
1	0	27.0	48000.0
2	1	30.0	54000.0
3	0	38.0	61000.0
4	1	40.0	61000.0
5	2	35.0	58000.0
6	0	38.0	52000.0
7	2	48.0	79000.0
8	1	50.0	83000.0
9	2	37.0	67000.0

X_train - NumPy object array (read only)

	0	1	2
0	2	44.0	72000.0
1	1	40.0	61000.0
2	2	35.0	58000.0
3	2	48.0	79000.0
4	2	37.0	67000.0
5	1	50.0	83000.0
6	0	27.0	48000.0

X_test - NumPy object array (read only)

	0	1	2
0	0	38.0	52000.0
1	1	30.0	54000.0
2	0	38.0	61000.0

y_test - NumPy object array

	0
0	0
1	0
2	0

y_train - NumPy object array

	0
0	0
1	1
2	1
3	1
4	1
5	0
6	1

y - NumPy object array

	0
0	0
1	1
2	0
3	0
4	1
5	1
6	0
7	1
8	0
9	1