```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt
```

```
# Load dataset
df = pd.read_csv(r"C:\Users\hp\Downloads\Data (1).csv")
```

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

```
# Split dataset into features (x) and target (y)
x = df.iloc[:, :-1].values   # All rows, all columns except last one →Features
```

	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

```
y = df.iloc[:, 3].values     # All rows, 4th column (index 3) → Target
```

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

```
# Handle missing values using mean strategy
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy="mean")
x[:, 1:3] = imputer.fit_transform(x[:, 1:3])  # Apply imputation on columns 1
and 2

# Encode categorical data
from sklearn.preprocessing import LabelEncoder

labelencoder_x = LabelEncoder()
x[:, 0] = labelencoder_x.fit_transform(x[:, 0])   # Encode first column of
features

labelencoder_y = LabelEncoder()
```

```
y = labelencoder_y.fit_transform(y)  # Encode target variable

# Split dataset into training and test sets
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=0)
```

	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	63777.77777777…
5	2	35.0	58000.0
6	0	38.77777777777…	52000.0
7	2	48.0	79000.0
8	1	50.0	83000.0
9	2	37.0	67000.0

	0	1	2
0	1	30.0	54000.0
1	1	50.0	83000.0

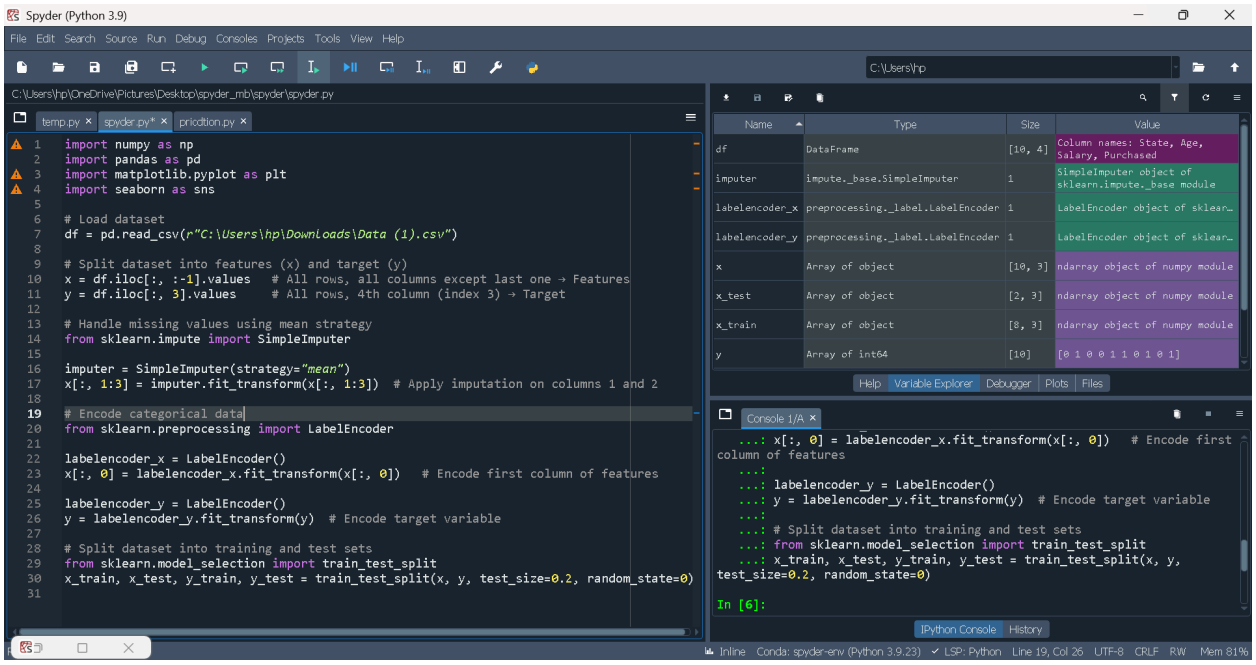
```
x_train
```

	0	1	2
0	1	40.0	63777.77777777…
1	2	37.0	67000.0
2	0	27.0	48000.0
3	0	38.77777777777…	52000.0
4	2	48.0	79000.0
5	0	38.0	61000.0
6	2	44.0	72000.0
7	2	35.0	58000.0

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

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

```
import seaborn as sns

# Load dataset
df = pd.read_csv(r"C:\Users\hp\Downloads\Data (1).csv")

# Split dataset into features (x) and target (y)
x = df.iloc[:, :-1].values   # All rows, all columns except last one → Features
y = df.iloc[:, 3].values     # All rows, 4th column (index 3) → Target

# Handle missing values using mean strategy
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy="mean")
x[:, 1:3] = imputer.fit_transform(x[:, 1:3])  # Apply imputation on columns 1 and 2

# Encode categorical data
from sklearn.preprocessing import LabelEncoder

labelencoder_x = LabelEncoder()
x[:, 0] = labelencoder_x.fit_transform(x[:, 0])   # Encode first column of features

labelencoder_y = LabelEncoder()
y = labelencoder_y.fit_transform(y)  # Encode target variable

# Split dataset into training and test sets
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\hp\OneDrive\Pictures\Desktop\spyder_mb\spyder\spyder.py
temp.py
spyder.py*
pricdtion.py
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5
6 # Load dataset
7 df = pd.read_csv(r"C:\Users\hp\Downloads\Data (1).csv")
8
9 # Split dataset into features (x) and target (y)
10 x = df.iloc[:, :-1].values # All rows, all columns except last one → Features
11 y = df.iloc[:, 3].values # All rows, 4th column (index 3) → Target
12
13 # Handle missing values using mean strategy
14 from sklearn.impute import SimpleImputer
15
16 imputer = SimpleImputer(strategy="mean")
17 x[:, 1:3] = imputer.fit_transform(x[:, 1:3]) # Apply imputation on columns 1 and 2
18
19 # Encode categorical data
20 from sklearn.preprocessing import LabelEncoder
21
22 labelencoder_x = LabelEncoder()
23 x[:, 0] = labelencoder_x.fit_transform(x[:, 0]) # Encode first column of features
24
25 labelencoder_y = LabelEncoder()
26 y = labelencoder_y.fit_transform(y) # Encode target variable
27
28 # Split dataset into training and test sets
29 from sklearn.model_selection import train_test_split
30 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
31
Name Type Size Value
df DataFrame [10, 4] Column names: State, Age, Salary, Purchased
imputer impute._base.SimpleImputer 1 SimpleImputer object of sklearn.impute._base module
labelencoder_x preprocessing._label.LabelEncoder 1 LabelEncoder object of sklear…
labelencoder_y preprocessing._label.LabelEncoder 1 LabelEncoder object of sklear…
x Array of object [10, 3] ndarray object of numpy module
x_test Array of object [2, 3] ndarray object of numpy module
x_train Array of object [8, 3] ndarray object of numpy module
y Array of int64 [10] [0 1 0 0 1 1 0 1 0 1]
Help Variable Explorer Debugger Plots Files
Console 1/A
...: x[:, 0] = labelencoder_x.fit_transform(x[:, 0]) # Encode first column of features
...:
...: labelencoder_y = LabelEncoder()
...: y = labelencoder_y.fit_transform(y) # Encode target variable
...:
...: # Split dataset into training and test sets
...: from sklearn.model_selection import train_test_split
...: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
In [6]:
IPython Console History
Inline Conda: spyder-env (Python 3.9.23) LSP: Python Line 19, Col 26 UTF-8 CRLF RW Mem 81%