Data Preprocessing Tools

Importing the libraries

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

Importing the dataset

```python
dataset = pd.read_csv('Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

```python
print(X)
```

```python
print(y)
```

Taking care of missing data

```python
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer.fit(X[:, 1:3])
X[:, 1:3] = imputer.transform(X[:, 1:3])
```

```python
print(X)
```

Encoding categorical data

Encoding the Independent Variable

```python
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
X = np.array(ct.fit_transform(X))
```

```python
print(X)
```

Encoding the Dependent Variable

```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

```python
print(y)
```

Splitting the dataset into the Training set and Test set

```python
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 = 1)
```

```python
print(X_train)
```

```python
print(X_test)
```

```python
print(y_train)
```

```
print(y_test)
```

▾ Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
print(X_train)
```

```
print(X_test)
```