Data Preprocessing Tools

Importing the libraries

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

```
arr = np.array([1, 2, 3, 4, 5])
print(arr)
len(arr)
```

```
[1 2 3 4 5]
5
```

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
print(arr)
len(arr)
```

```
[[1 2 3]
 [4 5 6]]
2
```

```
arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
print(arr)
len(arr)
```

```
[[[1 2 3]
  [4 5 6]]

 [[1 2 3]
  [4 5 6]]]
2
```

```
x = np.arange(0,9,1)
print(x)
```

```
[0 1 2 3 4 5 6 7 8]
```

```
y = np.linspace(0,2,9)
print(y)
```

```
[0.   0.25 0.5  0.75 1.   1.25 1.5  1.75 2.  ]
```

```
z = np.add(x,y)
print(z)
type(z)
```

```
[ 0.    1.25  2.5   3.75  5.    6.25  7.5   8.75 10.  ]
numpy.ndarray
```

Importing the dataset

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

```
print(dataset)
```

```
   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
5   France  35.0  58000.0       Yes
6    Spain   NaN  52000.0        No
7   France  48.0  79000.0       Yes
8  Germany  50.0  83000.0        No
9   France  37.0  67000.0       Yes
```

```
print(X)
```

```
[['France' 44.0 72000.0]
 ['Spain' 27.0 48000.0]
 ['Germany' 30.0 54000.0]
 ['Spain' 38.0 61000.0]
 ['Germany' 40.0 nan]
 ['France' 35.0 58000.0]
 ['Spain' nan 52000.0]
 ['France' 48.0 79000.0]
 ['Germany' 50.0 83000.0]
 ['France' 37.0 67000.0]]
```

```
print(y)
```

```
['No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes']
```

Taking care of missing data

```
from sklearn.impute import SimpleImputer
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])
X[:, 1:3] = imputer.fit_transform(X[:, 1:3])
```

```
print(X)
```

```
     [['France' 44.0 72000.0]
      ['Spain' 27.0 48000.0]
      ['Germany' 30.0 54000.0]
      ['Spain' 38.0 61000.0]
      ['Germany' 40.0 63777.77777777778]
      ['France' 35.0 58000.0]
      ['Spain' 38.77777777777778 52000.0]
      ['France' 48.0 79000.0]
      ['Germany' 50.0 83000.0]
      ['France' 37.0 67000.0]]
```

Encoding categorical data

Encoding the Independent Variable

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

```
print(X)
```

```
     [[1.0 0.0 0.0 44.0 72000.0]
      [0.0 0.0 1.0 27.0 48000.0]
      [0.0 1.0 0.0 30.0 54000.0]
      [0.0 0.0 1.0 38.0 61000.0]
      [0.0 1.0 0.0 40.0 63777.77777777778]
      [1.0 0.0 0.0 35.0 58000.0]
      [0.0 0.0 1.0 38.77777777777778 52000.0]
      [1.0 0.0 0.0 48.0 79000.0]
      [0.0 1.0 0.0 50.0 83000.0]
      [1.0 0.0 0.0 37.0 67000.0]]
```

Encoding the Dependent Variable

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

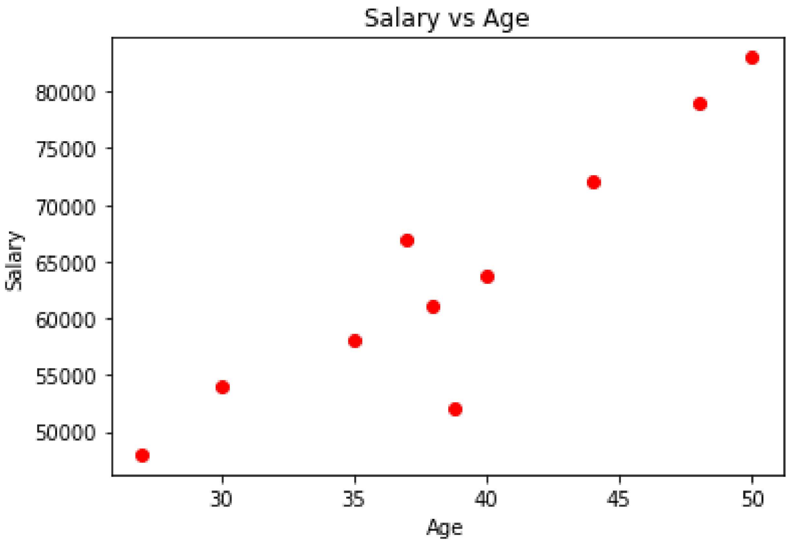
```
y = le.fit_transform(y)
```

```
print(y)
```

```
[0 1 0 0 1 1 0 1 0 1]
```

Plotting Salary vs Age

```
X_axis = X[:, 3:4]
y_axis = X[:, 4:5]
plt.scatter(X_axis, y_axis, color = 'red')
plt.title('Salary vs Age')
plt.xlabel('Age')
plt.ylabel('Salary')
plt.show()
```



Splitting the dataset into the Training set and Test set

```
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)
```

```
print(X_train)
```

```
[[0.0 0.0 1.0 38.77777777777778 52000.0]
 [0.0 1.0 0.0 40.0 63777.77777777778]
 [1.0 0.0 0.0 44.0 72000.0]
 [0.0 0.0 1.0 38.0 61000.0]
 [0.0 0.0 1.0 27.0 48000.0]
 [1.0 0.0 0.0 48.0 79000.0]
 [0.0 1.0 0.0 50.0 83000.0]
 [1.0 0.0 0.0 35.0 58000.0]]
```

```
print(X_test)
```

```
[[0.0 1.0 0.0 30.0 54000.0]
 [1.0 0.0 0.0 37.0 67000.0]]
```

```
print(y_train)
```

```
[0 1 0 0 1 1 0 1]
```

```
print(y_test)
```

```
[0 1]
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train[:, 3:] = sc.fit_transform(X_train[:, 3:])
X_test[:, 3:] = sc.transform(X_test[:, 3:])
```

```
print(X_train)
```

```
[[0.0 0.0 1.0 -0.19159184384578545 -1.0781259408412425]
 [0.0 1.0 0.0 -0.014117293757057777 -0.07013167641635372]
 [1.0 0.0 0.0 0.566708506533324 0.633562432710455]
 [0.0 0.0 1.0 -0.30453019390224867 -0.30786617274297867]
 [0.0 0.0 1.0 -1.9018011447007988 -1.420463615551582]
 [1.0 0.0 0.0 1.1475343068237058 1.232653363453549]
 [0.0 1.0 0.0 1.4379472069688968 1.5749910381638885]
 [1.0 0.0 0.0 -0.7401495441200351 -0.5646194287757332]]
```

```
print(X_test)
```

```
[[0.0 1.0 0.0 -1.4661817944830124 -0.9069571034860727]
 [1.0 0.0 0.0 -0.44973664397484414 0.2056403393225306]]
```

✓ 0s completed at 7:42 PM ● ✕