Data Preparation

انواع دیتاست ها :

به مجموعه داده ها که الگوریتم های یادگیری ماشین به واسطه آن ها فرآیند یادگیری خود را انجام می دهند ، دیتاست می گویند.

- **دیتاست های ساختاریافته (Structured Dataset) :**

در دیتاست های ساختاریافته معمولا سطرها به عنوان نمونه ها و ستون ها به عنوان ویژگی ها هستند. برای مثال فایل های txt ، excel ، csv و... به عنوان دیتاست های ساختاریافته شناخته می شوند.

- **دیتاست های بدون ساختار (Unstructured Dataset) :**

برای مثال دیتاست های تصویر ، صدا ، ویدیو و... به عنوان دیتاست های بدون ساختار شناخته می شوند.

بررسی اولیه دیتاست :

می توان با استفاده از دستورات زیر دید کلی نسبت به دیتاست بدست آورد.

```
import pandas as pd

data = pd.read_csv(r"C:\Users\Asus\Documents\AI\Data Preparation\World Happiness Report.csv")
data.info()
```

یا :

```
import pandas as pd

data = pd.read_csv(r"C:\Users\Asus\Documents\AI\Data Preparation\World Happiness Report.csv")
data.describe()
```

مقادیر نامعلوم (Missing Value) :

در دیتاست ها ممکن است بعضی از ویژگی های مربوط به برخی نمونه ها ، مقادیری نامعلوم یا از دست رفته داشته باشد که به آن ها Null یا None می گویند و یا برخی اوقات مقادیر NaN (Not a Number) را داریم.

به چند روش می توان مقادیر نامعلوم را اصلاح کرد.

- ✓ پر کردن مقادیر نامعلوم با میانگین نمونه های معلوم است.
- ✓ حذف مقادیر نامعلوم که به دو شکل ممکن است. حذف نمونه ای که ویژگی آن مقدار نامعلوم دارد یا حذف ویژگی ، برای تمامی نمونه ها.
- ✓ پر کردن مقادیر نامعلوم به صورت تصادفی.
- ✓ پر کردن مقادیر با نزدیکترین مقدار (KNN Imputer).

یکی از روش ها برای پر کردن مقادیر نامعلوم با استفاده از پایتون به شکل زیر است :

```
print(data.isnull().sum()) # data.isna()
print("Mean with zeros:", data.ConfidenceInNationalGovernment.mean())
mean = data[data.ConfidenceInNationalGovernment != 0].ConfidenceInNationalGovernment.mean()
print("Mean without zeros:", mean)
data.replace({"ConfidenceInNationalGovernment":0}, mean)
```

یکی دیگر از روش ها ، با استفاده از کتابخانه scikit-learn به شکل زیر است :

```
from sklearn.impute import SimpleImputer
import numpy as np

data.replace({"ConfidenceInNationalGovernment":0}, np.nan)
print(data.isna().sum())
SI = SimpleImputer(missing_values = np.nan, strategy = "mean")
SI.fit(data[["ConfidenceInNationalGovernment"]])
data["ConfidenceInNationalGovernment"] = SI.transform(data[["ConfidenceInNationalGovernment"]])
print("----------------------------------------")
print(data.isna().sum())
```

داده های غیر عددی (Categorical) :

در دیتاست ها بعضی از متغیرها یا ویژگی ها به صورت غیر عددی هستند که اگر info مربوط به دیتاست را مشاهده کنیم ، نوع آن ها به صورت object است.

به فرآیند تغییر نوع (Data Type) متغیر ، ویژگی و یا Label غیر عددی به عدد ، Encoding می گویند.

❖ Integer Encoding :

در این روش مقادیر نسبت به هم ترتیب (برتری) دارند.

برای مثال تبدیل First, Second, ... به 1, 2, ... که ترتیب دارند.

❖ One Hot Encoding :

در این روش مقادیر نسبت به هم ترتیب (برتری) ندارند.

برای مثال تبدیل نوع حیوان به عدد.

	عقاب	ببر	اسب
عقاب	1	0	0
ببر	0	1	0
اسب	0	0	1

نکته ! برای تحویل دیتاست به الگوریتم باید مقادیر عددی باشند تا الگوریتم به خوبی بتواند آن ها را درک کند و آموزش ببیند.

با استفاده از پایتون می توانیم Encoding را به صورت زیر انجام دهیم :

```
from sklearn.preprocessing import LabelEncoder

print(data.RegionalIndicator.unique())

enc = LabelEncoder()
data["RegionalIndicator"] = enc.fit_transform(data[["RegionalIndicator"]])
data
```

روش دیگر برای Encoding به صورت زیر است :

```
import numpy as np
from sklearn.preprocessing import OneHotEncoder

enc = OneHotEncoder(sparse = False)
arr = data.RegionalIndicator.unique().reshape(11, 1)
result = enc.fit_transform(arr)
print(result)
```

مقیاس داده ها (Scale) :

در بعضی از دیتاست ها ، مقیاس تمام متغیرها و نمونه های آن ها یکسان نیست. برای مثال نمونه های متغیری بین بازه ۱ تا ۱۰ است در حالی که نمونه های متغیر دیگری بین بازه ۱ تا ۱۰۰۰ است. در فرآیند یادگیری ماشین این تفاوت مقیاس و یا بازه باعث می شود تا الگوریتم یادگیری ماشین برای متغیری اهمیت بیشتری قائل شود در حالی که نباید اینطور در نظر بگیرد.
قبل از فرآیند یادگیری ماشین باید مقیاس نمونه های دیتاست یکسان شوند تا فرآیند یادگیری بهتر صورت گیرد.

❖ نرمال سازی (Normalization) :

در نرمال سازی ، تمامی داده ها بین بازه صفر تا ۱ قرار می گیرند.
برای قرارگیری داده ها بین بازه صفر تا ۱ از رابطه زیر استفاده می شود :

$$x' = \frac{x - min}{\max - min}$$

به این روش ، Min-Max Normalization می گویند.

با استفاده از پایتون می توان نرمال سازی را به صورت زیر انجام داد :

```
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
min_max_LogGDP = scaler.fit_transform(data[["LogGDP"]])
plt.subplot(2, 1, 1)
plt.hist(min_max_LogGDP, bins = 35)
plt.subplot(2, 1, 2)
plt.hist(data.LogGDP, bins = 35)
```

❖ **استاندارد سازی (Standardization) :**

در استاندارد سازی ، تمامی داده ها باید از توزیعی استاندارد با میانگین صفر و انحراف معیار ۱ پیروی کنند. به عبارت دیگر تمامی داده ها تقریبا بین بازه ۳- تا ۳ قرار می گیرند.

برای استاندارد سازی داده ها از رابطه زیر استفاده می شود :

$$Z - score = \frac{x - mean}{STD}$$

با استفاده از پایتون می توان استاندارد سازی را به صورت زیر انجام داد :

```
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
standard_LogGDP = scaler.fit_transform(data[["LogGDP"]])
plt.subplot(2, 1, 1)
plt.hist(standard_LogGDP, bins = 35)
plt.subplot(2, 1, 2)
plt.hist(data.LogGDP, bins = 35)
```

❖ **Binarizer** :

در باینریز کردن داده ها ، باید یک مقدار را به عنوان مرز یا Threshold تعیین کرد. اگر داده ای بزرگتر از آن بود ، مقدار آن را ۱ و اگر داده ای کوچکتر یا مساوی آن بود ، مقدار آن را صفر می کند.

با استفاده از پایتون می توان باینریز کردن داده ها را به صورت زیر انجام داد :

```
import matplotlib.pyplot as plt
from sklearn.preprocessing import Binarizer

binarize = Binarizer(threshold = 4)
binarize_LifeLadder = binarize.fit_transform(data[["LifeLadder"]])
plt.subplot(2, 1, 1)
plt.hist(binarize_LifeLadder, bins = 35)
plt.subplot(2, 1, 2)
plt.hist(data.LifeLadder, bins = 35)
```