Makine Öğrenimi Projelerinde Veriyi Almak ve Hazırlamak

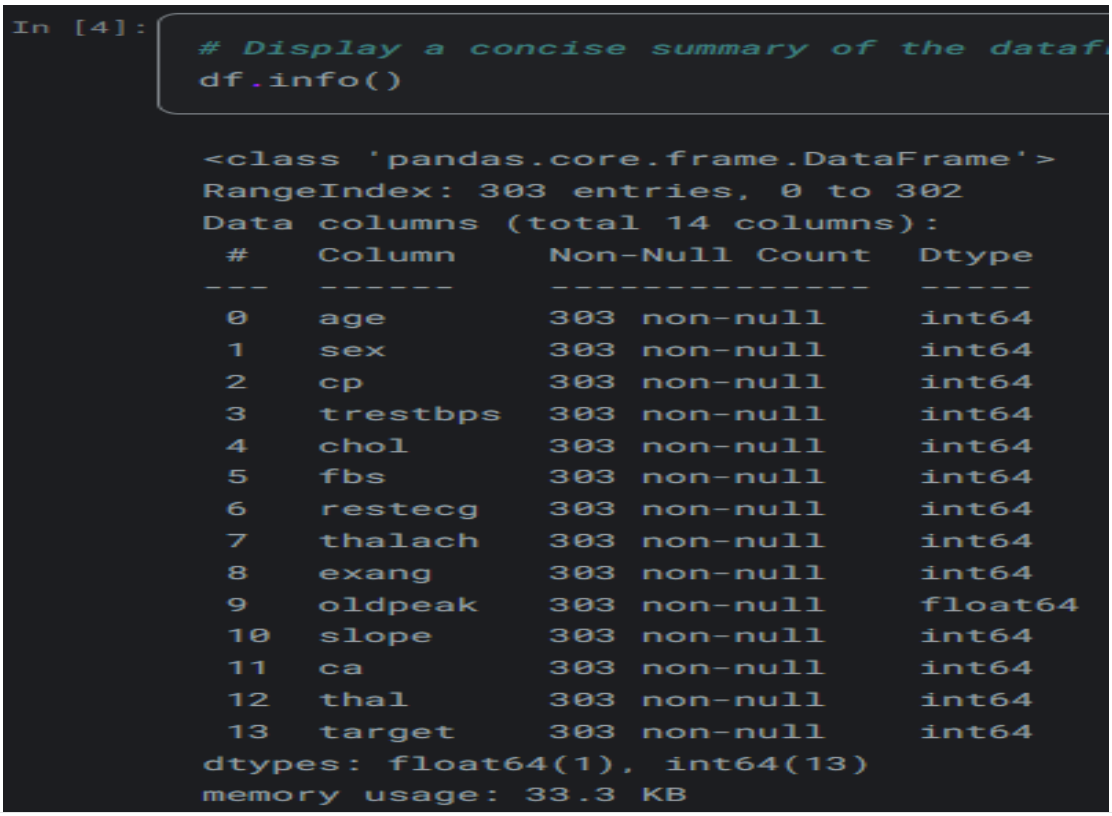
Makine öğrenimi projelerinin başarısı, büyük ölçüde kullanılan verinin kalitesine ve doğru şekilde hazırlanmasına bağlıdır. Bu makalede, bir makine öğrenimi projesi için veri setinin nasıl seçileceği, yükleneceği ve işleneceği adım adım ele alınacaktır.

1. Hangi Veriye ve Ne Kadar Veriye İhtiyacınız Var?

Makine öğrenimi projesi için uygun veri setini belirlemek, problemin çözümü için kritik bir adımdır. Verinin niteliği, problem türüne göre değişir. Örneğin, bir regresyon problemi için sayısal veriler gerekirken, sınıflandırma problemleri için kategorik veriler de gerekebilir. Ayrıca, yeterli miktarda veriye sahip olmak, modelin genel performansı için önemlidir. Genellikle, daha fazla veri daha iyi performans sağlar, ancak veri kalitesi de göz önünde bulundurulmalıdır.

2. Veriniz Ne Kadar Saklama Alanına Mal Olacaktır?

Veri setinin boyutu, saklama alanı ihtiyaçlarını belirlemek için değerlendirilmelidir. Veri setinin boyutu, kullanılan veri tipine (örneğin, sayısal, kategorik) ve veri sayısına bağlı olarak değişir.



```
In [4]: # Display a concise summary of the dataf
        df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #    Column     Non-Null Count  Dtype
---   ------     --------------  -----
 0    age        303 non-null    int64
 1    sex        303 non-null    int64
 2    cp         303 non-null    int64
 3    trestbps   303 non-null    int64
 4    chol       303 non-null    int64
 5    fbs        303 non-null    int64
 6    restecg    303 non-null    int64
 7    thalach    303 non-null    int64
 8    exang      303 non-null    int64
 9    oldpeak    303 non-null    float64
 10   slope      303 non-null    int64
 11   ca         303 non-null    int64
 12   thal       303 non-null    int64
 13   target     303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

3. Çalışma Alanınızı Oluşturun

Makine öğrenimi projelerinde çalışma ortamı, kullanılan programlama dili ve kütüphaneler ile oluşturulur. Python, yaygın olarak tercih edilen bir programlama dilidir ve makine öğrenimi için birçok güçlü kütüphane sunar. Proje ortamı için sanal bir ortam kurulabilir ve gerekli kütüphaneler bu ortama yüklenebilir.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.colors import ListedColormap
from sklearn.model_selection import train_test_split
from scipy.stats import boxcox
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV, StratifiedKFold
from sklearn.metrics import classification_report, accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

4. Veriyi Yükleyin

Çalışma alanı oluşturulduktan sonra, gerekli kütüphaneler yüklenmeli ve veri seti çalışma alanına alınmalıdır. Python'da pandas, numpy, seaborn ve matplotlib gibi kütüphaneler veri işleme ve görselleştirme için kullanılır.

```
df = pd.read_csv('/kaggle/input/heartcsv/heart.csv')
```

5. Veriyi Kolay Kullanabileceğiniz Bir Biçime Dönüştürün

Veri seti yüklendikten sonra, veriyi hızlıca gözden geçirmek önemlidir. Bu, veri setinin yapısını ve içeriğini anlamak için gereklidir.

```
import pandas as pd

# Veri setini yükleme
df = pd.read_csv('/kaggle/input/heartcsv/heart.csv')

# 1. Başlıca Özelliklerin İncelenmesi
print("Veri setinin sütunları ve örnek veri tipleri:")
print(df.dtypes)

# 2. Örnek Satırların Görüntülenmesi
print("\nVeri setinin ilk beş satırı:")
print(df.head())

# 3. Veri Tipi ve Eksik Değerlerin Kontrolü
print("\nVeri setinin veri tipleri ve eksik değerlerin varlığı:")
print(df.info())
```

Bu komut, veri setinin ilk beş satırını görüntüler ve veri setinin genel yapısı hakkında bilgi verir.

6. Verilerin Boyutunu ve Türünü Kontrol Edin

Veri setinin içeriğini ve veri tiplerini anlamak için aşağıdaki komutlar kullanılabilir:

```
# Veri setinin boyutu (satır, sütun)
print("Veri setinin boyutu:", df.shape)

# Sütunların adları ve veri tipleri ile her sütundaki eksik değer sayısı
print("\nSütun adları, veri tipleri ve eksik değer sayıları:")
print(df.info())
```

Bu komut, veri setindeki sütunların adlarını, veri tiplerini ve her sütundaki eksik değer sayısını gösterir. Bu bilgilere dayanarak veri setinin eksiksiz olduğunu ve her sütunun uygun veri tipinde olduğunu teyit edebiliriz.

7. Test Setinizi Oluşturun

Veri setinin bir kısmını eğitim için, bir kısmını ise modelin performansını değerlendirmek için test seti olarak ayırmak gereklidir. Genellikle veri setinin %20'si test seti olarak kullanılır. Aşağıda rastgele bir test seti oluşturma yöntemi verilmiştir.

```
from sklearn.model_selection import train_test_split

# Rastgele tohumu ayarla
random_seed = 42

# Bağımsız değişkenler (X) ve hedef değişken (y) ayrımı
X = df.drop('target', axis=1)
y = df['target']

# Veri setini eğitim ve test setlerine ayırma
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=random_seed)

# Eğitim setinin boyutu
print("Eğitim setinin boyutu:", X_train.shape)

# Test setinin boyutu
print("Test setinin boyutu:", X_test.shape)
```

Bu yöntem, veri setini rastgele bir şekilde ayırır. Ancak, rastgele ayırma her çalıştırmada farklı sonuçlar doğuracağı için, aynı test setini kullanmak isteyenler için uygun olmayabilir.

Sonuç

Makine öğrenimi projelerinde veri hazırlama aşaması, projenin başarısı için kritik bir öneme sahiptir. Bu makalede, uygun veri setinin nasıl seçileceği, yükleneceği ve işleneceği ele alınmıştır. Veri setinin doğru şekilde hazırlanması, modelin performansını önemli ölçüde etkileyebilir ve projenin genel başarısını artırabilir.