Data Loading in Machine Learning Projects

In order to start a machine learning project, the first and most important thing required is the data that we need to load for starting any of the machine learning projects. With respect to data, the most common format of data used for the projects is CSV (comma-separated values).

Basically, CSV is a simple file format which is used to store tabular data (number and text) such as a spreadsheet in plain text. In Python, we can load CSV data into with different ways but before loading CSV data we must have to take care about some considerations.

Consideration While Loading CSV data

CSV data format is the most common format for machine learning data, but we need to take care about following major considerations while loading the same into our machine learning projects .

Methods to Load CSV Data Files

In CSV data files, double quotation (“ ”) mark is the default quote character. It is important to consider the role of quotes while uploading the CSV file into ML projects because we can also use other quote character than double quotation mark. But in case of using a different quote character than standard one, we must have to specify it explicitly.

Load CSV with Python Standard Library:

The first and most used approach to load CSV data file is the use of Python standard library which provides us a variety of built-in modules namely *csv module* and the *reader()function.*

Let’s analyze with the help of an example:

Comments

Comments in a CSV file are indicated by a hash (“#”) at the start of a line.If you have comments in your file, depending on the method used to load your data, you may need to indicate whether or not to expect comments and the character to expect to signify a comment line.

Delimiter

The standard delimiter that separates values in fields is the comma (“,”) character.Your file could use a different delimiter like tab (“\t”) in which case you must specify it explicitly.

Quotes

Sometimes field values can have spaces. In these CSV files the values are often quoted.The default quote character is the double quotation marks “\””. Other characters can be used, and you must specify the quote character used in your file.

Loading CSV with NumPy

In this example, we are using the iris *flower data set* which can be downloaded into our local directory. After loading the data file, we can convert it into NumPy array and use it for machine learning projects.

Following is the Python script for loading CSV data file −

First, we need to import the csv module provided by Python standard library as follows

```
import csv
```

Next, we need to import NumPy module for converting the loaded data into NumPy array.

```
import numpy as np
```

Now, provide the full path of the file, stored on our local directory, having the CSV data file –

```
path = r"c:\iris.csv"
```

Next, use the csv.reader()function to read data from CSV file –

```
with open(path,'r') as f:
reader = csv.reader(f,delimiter = ',')
headers = next(reader)
data = list(reader)
data = np.array(data).astype(float)
```

Next script line will give the first three line of data file –

```
path = r"c:\iris.csv"
```

Output

```
['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
(150, 4)
[[5.1 3.5 1.4 0.2]
[4.9 3. 1.4 0.2]
[4.7 3.2 1.3 0.2]]
```

Load CSV with Pandas

Another approach to load CSV data file is by *Pandas* and pandas.read_csv()function. This is the very flexible function that returns a pandas.DataFrame which can be used immediately for plotting. The following is an example of loading CSV data file with the help of it –

The following is the Python script for loading CSV data file using *Pandas* on Iris *Data set* –
Script:

```
from pandas import read_csv
path = r"C:\iris.csv"
data = read_csv(path)
print(data.shape)
print(data[:3])
```

Output

```
(150, 4)
sepal_length sepal_width petal_length petal_width
0 5.1        3.5         1.4          0.2
1 4.9        3.0         1.4          0.2
2 4.7        3.2         1.3          0.2
```

Script-2:

The following is the Python script for loading CSV data file, along with providing the headers names too, using Pandas on Pima Indians Diabetes dataset –

```
from pandas import read_csv
path = r"C:\pima-indians-diabetes.csv"
headernames = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(path, names=headernames)
print(data.shape)
print(data[:3])
```

Output

```
(768, 9)
   preg  plas   pres  skin  test  mass   pedi   age  class
0     6   148     72    35     0  33.6  0.627    50      1
1     1    85     66    29     0  26.6  0.351    31      0
2     8   183     64     0     0  23.3  0.672    32      1
```

The difference between above used three approaches for loading CSV data file can easily be understood with the help of given examples.