Chapter 1 Load Data From CSV

arpit patel

13 May 2024

1. Introduction

You must know how to load data before you can use it to train a machine learning model. When starting out, it is a good idea to stick with small in-memory datasets using standard file formats like comma separated value (.csv). In this tutorial you will discover how to load your data in Python from scratch.

1.1 Description

1. The standard file format for small datasets is Comma Separated Values or CSV. In its simplest form, CSV files are comprised of rows of data. Each row is divided into columns using a comma (,). In this tutorial, we are going to practice loading two different, standard machine learning datasets in CSV format

2. Pima Indians Diabetes Dataset In this tutorial we will use the Pima Indians Diabetes Dataset. This dataset involves the predic- tion of the onset of diabetes within 5 years. The baseline performance on the problem is approximately 65 percentage. You can learn more about it in Appendix A, Section A.4. Download the dataset and save it into your current working directory with the filename pima-indians-diabetes.csv.

3. Iris Flower Species Dataset In this tutorial we will also use the Iris Flower Species Dataset. This dataset involves the prediction of iris flower species. The baseline performance on the problem is approximately 26 percentage. You can learn more about it in Appendix A,

Section A.7. Download the dataset and save it into your current working directory with the filename iris.csv.

1.2. Tutorial

This tutorial is divided into 3 parts:

1. Load a file.
2. Load a file and convert Strings to Floats. 3. Load a file and convert Strings to Integers.

These steps will provide the foundations you need to handle loading your own data.

1.2.1 Load CSV File

The first step is to load the CSV file. We will use the csv module that is a part of the standard library. The reader() function in the csv module takes a file as an argument.
We will create a function called load csv() to wrap this behavior that will take a filename and return our dataset. We will represent the loaded dataset as a list of lists. The first list is a list of observations or rows, and the second list is the list of column values for a given row. Below is the complete function for loading a CSV file.

```
# Load a CSV file
def load_csv(filename):
  file = open(filename, "r")
  lines = reader(file)
  dataset = list(lines)
  return dataset
```

We can test this function by loading the Pima Indians dataset. Taking a peek at the first 5 rows of the raw data file we can see the following:

```
6,148,72,35,0,33.6,0.627,50,1
1,85,66,29,0,26.6,0.351,31,0
8,183,64,0,0,23.3,0.672,32,1
1,89,66,23,94,28.1,0.167,21,0
0,137,40,35,168,43.1,2.288,33,1
```

1.2.2 Convert String to Floats

Most, if not all machine learning algorithms prefer to work with numbers. Specifically, floating point numbers are preferred. Our code for loading a CSV file returns a dataset as a list of lists, but each value is a string.

1.2.3 Convert String to Integers

The iris flowers dataset is like the Pima Indians dataset, in that the columns contain numeric data. The difference is the final column, traditionally used to hold the outcome or value to be predicted for a given row. The final column in the iris flowers data is the iris flower species as a string.