Datenvorbereitung neu ohne Kategorisierung ohne Bilder

December 26, 2020

1 Code zum Einlesen der Daten

Zu Beginn müssen wieder alle Bibliotheken eingebunden werden.

[1]:
```
#import of all necessary packages
import os
import numpy as np
import pandas as pd
from sklearn.utils import shuffle
from sklearn.preprocessing import MinMaxScaler
```

Für den Fall, dass die Datensätze nicht im selben Verzeichnis wie die Python-Datei liegt muss das Verzeichnis des Datensatzes angegeben werden. Dies kann durch folgenden Befehl durchgeführt werden.

[2]:
```
#function to set the start working directory manually
#not necessary, if the .py file is executed in the directory of the .csv files

#os.chdir('D:\\OneDrive - bwedu\\Uni\\09 ABC 1\\Neuronale␣
↪Netzwerke\\Python\\Wein')
```

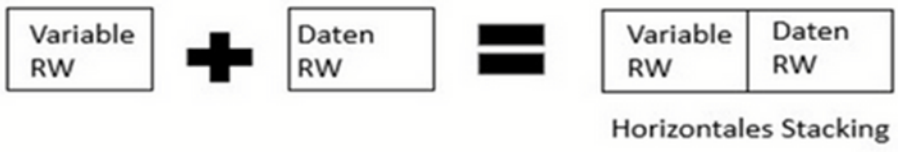
Anschließend folgt der Import der Datensätze. Die importierten Daten müssen anschließend in ein numpy-Array konvertiert werden.

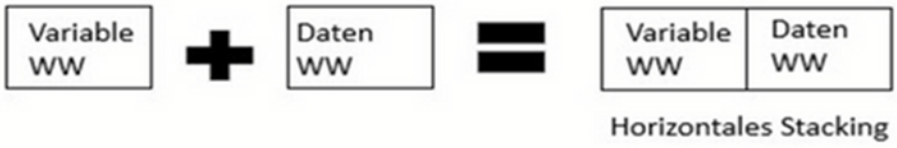
[3]:
```
#load raw data from the .csv files as panda dataframe
RW_orig = pd.read_csv('Rotwein.csv')
WW_orig = pd.read_csv('Weisswein.csv')

#convert panda dataframe to a numpy array
RW = pd.DataFrame.to_numpy(RW_orig)
WW = pd.DataFrame.to_numpy(WW_orig)
```

Erstellung und Zuordnung der Variablen für Rot (0)- und Weißweine (1):

[4]:
```
#create a new variabele for the two categories red (0) and white wine (1)
RW_num =  np.zeros((RW.shape[0],1), dtype = int)
WW_num = np.ones((WW.shape[0],1), dtype = int)
```

Hinzufügen der beiden Variablen in die Datensätze (horizontales Stacken):





[5]:
```
#add the new variable two the datasets
RW = np.hstack((RW_num, RW))
WW = np.hstack((WW_num, WW))
```

Vereinigung der beiden Datensätze zu einem gesammten Datensatz (vertikales Stacken):

[6]:
```
#Merge the two datasets
W = np.vstack((RW, WW))
```

Randomisieren der Daten:

[7]:
```
#shuffle the dataset
W = shuffle(W)
```

Separieren der 12 Weincharakteristika von der Weinqualität (response):

[8]:
```
#seperate the sample columns from the column with the labels (winequality)
W_samples = W[:,0:12]
W_labels = W[:,12]
```

Überführen der Daten in ein für das Netzwerk geeigneten Zahlenbereich (0-1)

[9]:
```
#scale the sample variables in a range from 0 to 1 (ANN can only work with␣
↪numbers from 0 to 1)
scaler = MinMaxScaler(feature_range=(0,1))
scaled_W_samples = scaler.fit_transform(W_samples)
```

Danach wird unser Datensatz in einen Trainings- und Testdatensatz aufgesplittet, da wir nur einen Ausgangsdatensatz zur Verfügung haben.

[10]:
```
#seperate the dataset into a training and a test dataset
W_train_samples = scaled_W_samples[0:5800,]
W_test_samples = scaled_W_samples[5800:6497,]
```

Nun werden die Daten in ein neues Verzeichnis abgespeichert. Dies hat den Vorteil, dass Datenvorbereitung und Training voneinander entkoppelt sind. Sollte z.B. beim Trainieren ein Fehler auftreten oder das Trainingsskript mehr als nur 1-Mal durchgeführt werden, müssen die Daten nicht erneut vorbereitet werden.

Codeabsatz 1 erstellt den Ordner für die Ablage der Dateien (sofern dieser noch nicht vorhanden ist) und Absatz 2 speichert die vorbereiteten Dateien in diesen Ordner ab.

[11]:
```
#save the current working directory
current_wd = os.getcwd()
#check if a directory 'prep_dataset' excist in the current working directory,␣
↪if not then create it
if os.path.isdir('prep_dataset') is False:
    os.makedirs('prep_dataset')
#change the working directory to the directory 'prep_dataset'
os.chdir('prep_dataset')

#export all the numpy arrays as .csv file, by converting it to a panda␣
↪dataframe and then exporting as a .csv file
pd.DataFrame(W_train_samples).to_csv("wine_train_samples.csv")
pd.DataFrame(W_test_samples).to_csv("wine_test_samples.csv")
```

Durch den folgenden Befehl wird der eben erstellte Unterordner wieder verlassen und die Datenvorbereitung ist abgeschlossen.

[12]:
```
#change the working directory back to the original working directory
os.chdir(current_wd)
```