ML1

Dieudonne Ouedraogo

3/11/2018

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import mglearn
import pandas as pd
    # create a simple dataset of people
data = {'Name': ["John", "Anna", "Peter", "Linda"],
            'Location' : ["New York", "Paris", "Berlin", "London"],
            'Age' : [24, 13, 53, 33]
}
data_pandas = pd.DataFrame(data)
print(data_pandas)
# Select all rows that have an age column greater than 30
print(data_pandas[data_pandas.Age > 30])

from sklearn.datasets import load_iris
iris_dataset = load_iris()
print("Keys of iris_dataset: \n{}".format(iris_dataset.keys()))
print(iris_dataset['DESCR'][:193] + "\n...")
print("Target names: {}".format(iris_dataset['target_names']))
print("Feature names: \n{}".format(iris_dataset['feature_names']))
print("Type of data: {}".format(type(iris_dataset['data'])))
print("Shape of data: {}".format(iris_dataset['data'].shape))
print("First five columns of data:\n{}".format(iris_dataset['data'][:5]))
print("Type of target: {}".format(type(iris_dataset['target'])))
print("Shape of target: {}".format(iris_dataset['target'].shape))
print("Target:\n{}".format(iris_dataset['target']))
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(iris_dataset['data'], iris_dataset['target'], rando
print("X_train shape: {}".format(X_train.shape))
print("y_train shape: {}".format(y_train.shape))
print("X_test shape: {}".format(X_test.shape))
print("y_test shape: {}".format(y_test.shape))
# create dataframe from data in X_train
# label the columns using the strings in iris_dataset.feature_names
iris_dataframe = pd.DataFrame(X_train, columns=iris_dataset.feature_names)
# create a scatter matrix from the dataframe, color by y_train
grr = pd.scatter_matrix(iris_dataframe, c=y_train, figsize=(15, 15), marker='o',
                            hist_kwds={'bins': 20}, s=60, alpha=.8, cmap=mglearn.cm3)
grr
#plt.show()
```

```
## -c:39: FutureWarning: pandas.scatter_matrix is deprecated. Use pandas.plotting.scatter_matrix instead
##    Age  Location   Name
## 0   24  New York   John
## 1   13     Paris   Anna
## 2   53    Berlin  Peter
```

```
## 3   33    London  Linda
##    Age Location   Name
## 2   53   Berlin  Peter
## 3   33   London  Linda
## Keys of iris_dataset:
## dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names'])
## Iris Plants Database
## ====================
##
## Notes
## -----
## Data Set Characteristics:
##     :Number of Instances: 150 (50 in each of three classes)
##     :Number of Attributes: 4 numeric, predictive att
## ...
## Target names: ['setosa' 'versicolor' 'virginica']
## Feature names:
## ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
## Type of data: <class 'numpy.ndarray'>
## Shape of data: (150, 4)
## First five columns of data:
## [[ 5.1  3.5  1.4  0.2]
##  [ 4.9  3.   1.4  0.2]
##  [ 4.7  3.2  1.3  0.2]
##  [ 4.6  3.1  1.5  0.2]
##  [ 5.   3.6  1.4  0.2]]
## Type of target: <class 'numpy.ndarray'>
## Shape of target: (150,)
## Target:
## [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
##  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  2 2]
## X_train shape: (112, 4)
## y_train shape: (112,)
## X_test shape: (38, 4)
## y_test shape: (38,)
```