Machine learning and pattern recognition

Introduction Lab

Rapid Miner for Data analysis, Machine learning, Pattern Recognition

In this exercise, we would like to demonstrate to you how rapidminer can perform the tasks that can be done by python/C++ in a very user-friendly way.

The below task is about loading and understanding Iris flower data for further processing.

1. Visit https://colab.research.google.com/notebooks/intro.ipynb
2. Run the commands below

```
# scipy
import scipy
print('scipy: %s' % scipy.__version__)
# numpy
import numpy
print('numpy: %s' % numpy.__version__)
# matplotlib
import matplotlib
print('matplotlib: %s' % matplotlib.__version__)
# pandas
import pandas
print('pandas: %s' % pandas.__version__)
# statsmodels
import statsmodels
print('statsmodels: %s' % statsmodels.__version__)
# scikit-learn
import sklearn
print('sklearn: %s' % sklearn.__version__)
```

3. Run the commands below

```
# Load libraries
from pandas import read_csv
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

4. Using pandas to load the data

```
import pandas as pd

# Load dataset

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"

names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']

dataset = pd.read_csv(url, names=names)

print(dataset.shape)
```

5. Let's look at the first 15 rows

```
print(dataset.head(15)).
```

6. Statistical Summary of each attribute.

```
print(dataset.describe())
```

output:

```
       sepal-length        sepal-width  petal-length        petal-width

count         150.000000          150.000000   150.000000          150.000000

mean          5.843333            3.054000      3.758667           1.198667

std             0.828066          0.433594      1.764420           0.763161

min             4.300000          2.000000      1.000000           0.100000

25%             5.100000          2.800000      1.600000           0.300000

50%             5.800000          3.000000      4.350000           1.300000

75%             6.400000          3.300000      5.100000           1.800000

max             7.900000          4.400000      6.900000           2.500000

>>>
```

7. Class Distribution

Let's look at the number of instances (rows) that belong to each class.

```
print(dataset.groupby('class').size())
```

output:

```
class

Iris-setosa         50

Iris-versicolor     50

Iris-virginica      50

dtype: int64
```

8. Note each class has the same number of instances

```
print(dataset.groupby('class').size())
```

9. Data Visualization

Univariate plots to better understand each attribute.

Create a histogram of each input variable to get an idea of the distribution.

```
dataset.hist()
```

output:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000002C5AF506888>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002C5AF539288>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000002C5AF56BF48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002C5AF5A5DC8>]],
       dtype=object)
```

Multivariate Plots - Look at the interactions between the variables.

Scatterplots of all pairs of attributes. This is useful to spot structured relationships between input variables.

```
from pandas.plotting import scatter_matrix

scatter_matrix(dataset)
```

Note the diagonal grouping of some pairs of attributes. This suggests a high correlation and a predictable relationship.

How can we do all the previous using RapidMiner?

Read about 10-fold cross validation : https://machinelearningmastery.com/k-fold-cross-validation/

References: https://machinelearningmastery.com/