keras

In [1]:

```
import tensorflow as tf
```

In [2]:

```
from sklearn.datasets import load_wine
```

In [3]:

```
vino = load_wine()
print(vino['DESCR'])
```

```
.. _wine_dataset:

Wine recognition dataset
------------------------

**Data Set Characteristics:**

    :Number of Instances: 178 (50 in each of three classes)
    :Number of Attributes: 13 numeric, predictive attributes and the class
    :Attribute Information:
 		- Alcohol
 		- Malic acid
 		- Ash
		- Alcalinity of ash
 		- Magnesium
		- Total phenols
 		- Flavanoids
 		- Nonflavanoid phenols
 		- Proanthocyanins
		- Color intensity
 		- Hue
 		- OD280/OD315 of diluted wines
 		- Proline

    - class:
            - class_0
            - class_1
            - class_2

    :Summary Statistics:

    ============================= ==== ===== ======= =====
                                   Min   Max   Mean     SD
    ============================= ==== ===== ======= =====
    Alcohol:                      11.0  14.8    13.0   0.8
    Malic Acid:                   0.74  5.80    2.34  1.12
    Ash:                          1.36  3.23    2.36  0.27
    Alcalinity of Ash:            10.6  30.0    19.5   3.3
    Magnesium:                    70.0 162.0    99.7  14.3
    Total Phenols:                0.98  3.88    2.29  0.63
    Flavanoids:                   0.34  5.08    2.03  1.00
    Nonflavanoid Phenols:         0.13  0.66    0.36  0.12
    Proanthocyanins:              0.41  3.58    1.59  0.57
    Colour Intensity:              1.3  13.0     5.1   2.3
```

```
    Hue:                          0.48  1.71    0.96  0.23
    OD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71
    Proline:                       278  1680     746   315
    ============================= ==== ===== ======= =====

    :Missing Attribute Values: None
    :Class Distribution: class_0 (59), class_1 (71), class_2 (48)
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988

This is a copy of UCI ML Wine recognition datasets.
https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data

The data is the results of a chemical analysis of wines grown in the same
region in Italy by three different cultivators. There are thirteen different
measurements taken for different constituents found in the three types of
wine.

Original Owners:

Forina, M. et al, PARVUS -
An Extendible Package for Data Exploration, Classification and Correlation.
Institute of Pharmaceutical and Food Analysis and Technologies,
Via Brigata Salerno, 16147 Genoa, Italy.

Citation:

Lichman, M. (2013). UCI Machine Learning Repository
[https://archive.ics.uci.edu/ml]. Irvine, CA: University of California,
School of Information and Computer Science.

.. topic:: References

  (1) S. Aeberhard, D. Coomans and O. de Vel,
  Comparison of Classifiers in High Dimensional Settings,
  Tech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of
  Mathematics and Statistics, James Cook University of North Queensland.
  (Also submitted to Technometrics).

  The data was used with many others for comparing various
  classifiers. The classes are separable, though only RDA
  has achieved 100% correct classification.
  (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data))
  (All results using the leave-one-out technique)

  (2) S. Aeberhard, D. Coomans and O. de Vel,
  "THE CLASSIFICATION PERFORMANCE OF RDA"
  Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of
  Mathematics and Statistics, James Cook University of North Queensland.
  (Also submitted to Journal of Chemometrics).
```

In [4]:

```python
caracteristicas = vino['data']
objetivo = vino['target']
```

In [5]:

```python
from sklearn.model_selection import train_test_split
```

In [6]:

```
X_train, X_test, y_train, y_test = train_test_split(caracteristicas, objetivo, test_size=0.3)
```

In [8]:

```
from sklearn.preprocessing import MinMaxScaler
```

In [10]:

```
normalizador = MinMaxScaler()
```

In [11]:

```
x_train_normalizado = normalizador.fit_transform(X_train)
x_test_normalizado = normalizador.transform(X_test)
```

In [13]:

```
from tensorflow.contrib.keras import models, layers, losses, optimizers, metrics, activations
```

In [14]:

```
modelo = models.Sequential()
```

In [15]:

```
modelo.add(layers.Dense(units=13, input_dim=13, activation='relu'))
```

```
WARNING:tensorflow:From C:\Users\SARA\anaconda3\envs\pruebasTensorflow\lib\site-packages\tensorflow_c
ore\python\ops\resource_variable_ops.py:1630: calling BaseResourceVariable.__init__ (from tensorflow.python
.ops.resource_variable_ops) with constraint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
```

In [16]:

```
modelo.add(layers.Dense(units=13, activation='relu'))
```

In [17]:

```
modelo.add(layers.Dense(units=13, activation='relu'))
```

In [18]:

```
modelo.add(layers.Dense(units=13, activation='softmax'))
```

In [20]:

```
modelo.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

In [22]:

```
modelo.fit(x_train_normalizado, y_train, epochs=60)
```

```
Train on 124 samples
Epoch 1/60
124/124 [==============================] - 1s 9ms/sample - loss: 2.6100 - acc: 0.2258
Epoch 2/60
124/124 [==============================] - 0s 290us/sample - loss: 2.5668 - acc: 0.2581
Epoch 3/60
124/124 [==============================] - 0s 258us/sample - loss: 2.5274 - acc: 0.2581
```

```
124/124 [==============================] - 0s 258us/sample - loss: 2.5274 - acc: 0.2581
Epoch 4/60
124/124 [==============================] - 0s 258us/sample - loss: 2.4880 - acc: 0.2742
Epoch 5/60
124/124 [==============================] - 0s 258us/sample - loss: 2.4486 - acc: 0.2742
Epoch 6/60
124/124 [==============================] - 0s 226us/sample - loss: 2.4096 - acc: 0.2742
Epoch 7/60
124/124 [==============================] - 0s 194us/sample - loss: 2.3691 - acc: 0.2742
Epoch 8/60
124/124 [==============================] - 0s 258us/sample - loss: 2.3275 - acc: 0.2742
Epoch 9/60
124/124 [==============================] - 0s 194us/sample - loss: 2.2859 - acc: 0.2742
Epoch 10/60
124/124 [==============================] - 0s 194us/sample - loss: 2.2433 - acc: 0.2742
Epoch 11/60
124/124 [==============================] - 0s 226us/sample - loss: 2.1976 - acc: 0.2742
Epoch 12/60
124/124 [==============================] - 0s 258us/sample - loss: 2.1492 - acc: 0.2742
Epoch 13/60
124/124 [==============================] - 0s 258us/sample - loss: 2.1019 - acc: 0.2742
Epoch 14/60
124/124 [==============================] - 0s 581us/sample - loss: 2.0494 - acc: 0.2742
Epoch 15/60
124/124 [==============================] - 0s 194us/sample - loss: 1.9989 - acc: 0.2742
Epoch 16/60
124/124 [==============================] - 0s 258us/sample - loss: 1.9459 - acc: 0.2742
Epoch 17/60
124/124 [==============================] - 0s 226us/sample - loss: 1.8944 - acc: 0.2742
Epoch 18/60
124/124 [==============================] - 0s 387us/sample - loss: 1.8414 - acc: 0.2742
Epoch 19/60
124/124 [==============================] - 0s 258us/sample - loss: 1.7916 - acc: 0.2742
Epoch 20/60
124/124 [==============================] - 0s 258us/sample - loss: 1.7419 - acc: 0.2742
Epoch 21/60
124/124 [==============================] - 0s 323us/sample - loss: 1.6913 - acc: 0.2742
Epoch 22/60
124/124 [==============================] - 0s 258us/sample - loss: 1.6415 - acc: 0.2742
Epoch 23/60
124/124 [==============================] - 0s 258us/sample - loss: 1.5906 - acc: 0.2742
Epoch 24/60
124/124 [==============================] - 0s 194us/sample - loss: 1.5395 - acc: 0.2742
Epoch 25/60
124/124 [==============================] - 0s 194us/sample - loss: 1.4850 - acc: 0.2742
Epoch 26/60
124/124 [==============================] - 0s 290us/sample - loss: 1.4332 - acc: 0.2742
Epoch 27/60
124/124 [==============================] - 0s 161us/sample - loss: 1.3778 - acc: 0.2742
Epoch 28/60
124/124 [==============================] - 0s 194us/sample - loss: 1.3204 - acc: 0.2742
Epoch 29/60
124/124 [==============================] - 0s 226us/sample - loss: 1.2653 - acc: 0.2742
Epoch 30/60
124/124 [==============================] - 0s 194us/sample - loss: 1.2080 - acc: 0.4274
Epoch 31/60
124/124 [==============================] - 0s 226us/sample - loss: 1.1541 - acc: 0.5565
Epoch 32/60
124/124 [==============================] - 0s 194us/sample - loss: 1.0988 - acc: 0.6290
Epoch 33/60
124/124 [==============================] - 0s 194us/sample - loss: 1.0510 - acc: 0.6532
Epoch 34/60
124/124 [==============================] - 0s 161us/sample - loss: 1.0048 - acc: 0.6532
```

```
124/124 [==============================] - 0s 161us/sample - loss: 1.0048 - acc: 0.6532
Epoch 35/60
124/124 [==============================] - 0s 161us/sample - loss: 0.9639 - acc: 0.6532
Epoch 36/60
124/124 [==============================] - 0s 226us/sample - loss: 0.9289 - acc: 0.6532
Epoch 37/60
124/124 [==============================] - 0s 290us/sample - loss: 0.8945 - acc: 0.6532
Epoch 38/60
124/124 [==============================] - 0s 194us/sample - loss: 0.8646 - acc: 0.6532
Epoch 39/60
124/124 [==============================] - 0s 290us/sample - loss: 0.8341 - acc: 0.6532
Epoch 40/60
124/124 [==============================] - 0s 194us/sample - loss: 0.8065 - acc: 0.6532
Epoch 41/60
124/124 [==============================] - 0s 194us/sample - loss: 0.7809 - acc: 0.6532
Epoch 42/60
124/124 [==============================] - 0s 226us/sample - loss: 0.7558 - acc: 0.6532
Epoch 43/60
124/124 [==============================] - 0s 290us/sample - loss: 0.7332 - acc: 0.6613
Epoch 44/60
124/124 [==============================] - 0s 161us/sample - loss: 0.7109 - acc: 0.6613
Epoch 45/60
124/124 [==============================] - ETA: 0s - loss: 0.7078 - acc: 0.625 - 0s 194us/sample - loss: 0.6894 - acc: 0.6694
Epoch 46/60
124/124 [==============================] - 0s 161us/sample - loss: 0.6700 - acc: 0.6694
Epoch 47/60
124/124 [==============================] - 0s 226us/sample - loss: 0.6509 - acc: 0.6694
Epoch 48/60
124/124 [==============================] - 0s 194us/sample - loss: 0.6333 - acc: 0.6694
Epoch 49/60
124/124 [==============================] - 0s 194us/sample - loss: 0.6165 - acc: 0.6935
Epoch 50/60
124/124 [==============================] - 0s 226us/sample - loss: 0.6010 - acc: 0.7016
Epoch 51/60
124/124 [==============================] - 0s 226us/sample - loss: 0.5866 - acc: 0.7016
Epoch 52/60
124/124 [==============================] - 0s 290us/sample - loss: 0.5738 - acc: 0.7177
Epoch 53/60
124/124 [==============================] - 0s 194us/sample - loss: 0.5605 - acc: 0.7500
Epoch 54/60
124/124 [==============================] - 0s 226us/sample - loss: 0.5488 - acc: 0.7661
Epoch 55/60
124/124 [==============================] - ETA: 0s - loss: 0.5234 - acc: 0.875 - 0s 365us/sample - loss: 0.5369 - acc: 0.7661
Epoch 56/60
124/124 [==============================] - 0s 194us/sample - loss: 0.5269 - acc: 0.7742
Epoch 57/60
124/124 [==============================] - 0s 161us/sample - loss: 0.5161 - acc: 0.8145
Epoch 58/60
124/124 [==============================] - 0s 226us/sample - loss: 0.5061 - acc: 0.8226
Epoch 59/60
124/124 [==============================] - 0s 194us/sample - loss: 0.4969 - acc: 0.8226
Epoch 60/60
124/124 [==============================] - 0s 161us/sample - loss: 0.4882 - acc: 0.8226
```

Out[22]:

```
<tensorflow.python.keras.callbacks.History at 0x26cd41ece48>
```

In [23]:

```
predicciones = modelo.predict_classes(x_test_normalizado)
```

In [25]:

```
from sklearn.metrics import classification_report
print(classification_report(y_test,predicciones))
```

```
              precision    recall  f1-score   support

           0       0.50      0.92      0.65        12
           1       0.94      0.54      0.68        28
           2       0.88      1.00      0.93        14

    accuracy                           0.74        54
   macro avg       0.77      0.82      0.75        54
weighted avg       0.82      0.74      0.74        54
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: