Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exe24 - Naive Bayes Classification Exercise

Name: Phua Yan Han

IC Number: 050824070059

Date :

Introduction :

Conclusion : ¶

Naive Bayes exercise

Naive Bayes classification walkthrough

In [14]:
```
# Import scikit-learn dataset library
import sklearn as sk
from sklearn import datasets

#Load dataset
wine = datasets.load_wine()
# print the wine data features (top 5 records)
```

Out[14]:

```
{'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
         1.065e+03],
        [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
         1.050e+03],
        [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
         1.185e+03],
        ...,
        [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
         8.350e+02],
        [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
         8.400e+02],
        [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
         5.600e+02]]),
 'target': array([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, 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,
        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]),
 'frame': None,
 'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
 'DESCR': '.. _wine_dataset:\n\nWine recognition dataset\n----------
-----\n\n**Data Set Characteristics:**\n\n    :Number of Instances: 178\n
:Number of Attributes: 13 numeric, predictive attributes and the class\n    :
Attribute Information:\n \t\t- Alcohol\n \t\t- Malic acid\n \t\t- Ash\n\t\t-
Alcalinity of ash  \n \t\t- Magnesium\n\t\t- Total phenols\n \t\t- Flavanoids
\n \t\t- Nonflavanoid phenols\n \t\t- Proanthocyanins\n\t\t- Color intensity
\n \t\t- Hue\n \t\t- OD280/OD315 of diluted wines\n \t\t- Proline\n\n    - cl
ass:\n            - class_0\n            - class_1\n            - class_2\n\t
\t\n    :Summary Statistics:\n    \n    ============================== ==== ==
=== ======= =====\n                                   Min   Max   Mean     SD
\n    ============================== ==== ===== ======= =====\n    Alcohol:
11.0  14.8    13.0   0.8\n    Malic Acid:                    0.74  5.80    2.3
4  1.12\n    Ash:                           1.36  3.23    2.36  0.27\n    Alca
linity of Ash:            10.6  30.0    19.5   3.3\n    Magnesium:
70.0 162.0    99.7  14.3\n    Total Phenols:                 0.98  3.88    2.2
9  0.63\n    Flavanoids:                    0.34  5.08    2.03  1.00\n    Nonf
lavanoid Phenols:         0.13  0.66    0.36  0.12\n    Proanthocyanins:
0.41  3.58    1.59  0.57\n    Colour Intensity:              1.3  13.0     5.
1   2.3\n    Hue:                           0.48  1.71    0.96  0.23\n    OD28
0/OD315 of diluted wines: 1.27  4.00    2.61  0.71\n    Proline:
278  1680     746   315\n    ============================== ==== ===== =======
=====\n\n    :Missing Attribute Values: None\n    :Class Distribution: class_
0 (59), class_1 (71), class_2 (48)\n    :Creator: R.A. Fisher\n    :Donor: Mi
chael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n    :Date: July, 1988\n\nThis
is a copy of UCI ML Wine recognition datasets.\nhttps://archive.ics.uci.edu/m
l/machine-learning-databases/wine/wine.data\n\nThe data is the results of a c
hemical analysis of wines grown in the same\nregion in Italy by three differe
nt cultivators. There are thirteen different\nmeasurements taken for differen
t constituents found in the three types of\nwine.\n\nOriginal Owners: \n\nFor
ina, M. et al, PARVUS - \nAn Extendible Package for Data Exploration, Classif
ication and Correlation. \nInstitute of Pharmaceutical and Food Analysis and
Technologies,\nVia Brigata Salerno, 16147 Genoa, Italy.\n\nCitation:\n\nLichm
```

```
an, M. (2013). UCI Machine Learning Repository\n[https://archive.ics.uci.edu/
ml]. Irvine, CA: University of California,\nSchool of Information and Compute
r Science. \n\n.. topic:: References\n\n  (1) S. Aeberhard, D. Coomans and O.
de Vel, \n  Comparison of Classifiers in High Dimensional Settings, \n  Tech.
Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of  \n  Mathemati
cs and Statistics, James Cook University of North Queensland. \n  (Also submi
tted to Technometrics). \n\n  The data was used with many others for comparin
g various \n  classifiers. The classes are separable, though only RDA \n  has
achieved 100% correct classification. \n  (RDA : 100%, QDA 99.4%, LDA 98.9%,
1NN 96.1% (z-transformed data)) \n  (All results using the leave-one-out tech
nique) \n\n  (2) S. Aeberhard, D. Coomans and O. de Vel, \n  "THE CLASSIFICAT
ION PERFORMANCE OF RDA" \n  Tech. Rep. no. 92-01, (1992), Dept. of Computer S
cience and Dept. of \n  Mathematics and Statistics, James Cook University of
North Queensland. \n  (Also submitted to Journal of Chemometrics).\n',
 'feature_names': ['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline']}
```

In [ ]:
```
# print the names of the 13 features
print(wine.feature_names)
# print the label type of wine(class_0, class_1, class_2)
print(wine.target_names)
```

In [ ]:
```
# print data(feature)shape
print(wine.data.shape)
```

In [15]:
```
# print the wine data features (top 5 records)
print(wine.data[:5])
```

```
[[1.423e+01 1.710e+00 2.430e+00 1.560e+01 1.270e+02 2.800e+00 3.060e+00
  2.800e-01 2.290e+00 5.640e+00 1.040e+00 3.920e+00 1.065e+03]
 [1.320e+01 1.780e+00 2.140e+00 1.120e+01 1.000e+02 2.650e+00 2.760e+00
  2.600e-01 1.280e+00 4.380e+00 1.050e+00 3.400e+00 1.050e+03]
 [1.316e+01 2.360e+00 2.670e+00 1.860e+01 1.010e+02 2.800e+00 3.240e+00
  3.000e-01 2.810e+00 5.680e+00 1.030e+00 3.170e+00 1.185e+03]
 [1.437e+01 1.950e+00 2.500e+00 1.680e+01 1.130e+02 3.850e+00 3.490e+00
  2.400e-01 2.180e+00 7.800e+00 8.600e-01 3.450e+00 1.480e+03]
 [1.324e+01 2.590e+00 2.870e+00 2.100e+01 1.180e+02 2.800e+00 2.690e+00
  3.900e-01 1.820e+00 4.320e+00 1.040e+00 2.930e+00 7.350e+02]]
```

In [ ]:
```
# print the wine labels (0:Class_0, 1:class_2, 2:class_2)
print(wine.target_names)
```

In [17]:
```
# Import train_test_split function
from sklearn.model_selection import train_test_split

# Split dataset into training set and test set
X = wine.data
y = wine.target
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_s
```

In [21]:
```
#Import Gaussian Naive Bayes model
from sklearn.naive_bayes import GaussianNB

#Create a Gaussian Classifier
model = GaussianNB()
#Train the model using the training sets
model.fit(X_train,y_train)
#Predict the response for test dataset
y_pred = model.predict(X_test)
```

In [22]:
```
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics,tree

# Model Accuracy, how often is the classifier correct?
print("accuracy", metrics.accuracy_score(y_test,y_pred))
```

```
accuracy 1.0
```

Exercise 1 : Perform NB classification using the Iris dataset

In [4]:
```
## Exercise 1 : Perform NB classification using the iris dataset

# Load libraries
from sklearn import datasets
import matplotlib.pyplot as plt

# Load iris dataset
iris = datasets.load_iris()

# Create feature matrix


# Create target vector

# View the first observation's feature values
```

Exercise 2 : Perform NB classification using the Titanic dataset

In [ ]: