Naive Bayes Classifier

March 10, 2019

```
In [26]: %matplotlib inline
         import numpy as np
         from matplotlib import pyplot as plt
         from sklearn.datasets import load_boston
         import pandas as pd
         boston = load_boston()

In [27]: type(boston)
         type(boston.data)
         boston.feature_names
         print(boston.DESCR)
```

```
.. _boston_dataset:

Boston house prices dataset
---------------------------

**Data Set Characteristics:**

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usu

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
        - LSTAT    % lower status of the population
```

```
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/


This dataset was taken from the StatLib library which is maintained at Carnegie Mellon Universit

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980.   N.B. Various transformations are used in the table on
pages 244-261 of the latter.

The Boston house-price data has been used in many machine learning papers that address regressio
problems.

.. topic:: References

   - Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential Data and Sources of
   - Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the
```

In [28]:
```python
bos = pd.DataFrame(boston.data)
bos.columns = boston.feature_names
bos.head()
```

Out[28]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	\
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	

	PTRATIO	B	LSTAT
0	15.3	396.90	4.98
1	17.8	396.90	9.14
2	17.8	392.83	4.03
3	18.7	394.63	2.94
4	18.7	396.90	5.33

In [33]:
```python
# Here, CHAS is collection of distinct classes namely 0.0 and 1.0.
```

```
# Import train_test_split function
from sklearn.model_selection import train_test_split

# Split dataset into training set and test set
X_train, X_test = train_test_split(bos, test_size=0.3,random_state=109) # 70% training
```

```
{0.0, 1.0}
```

```
In [35]: # Instantiate the classifier
         from sklearn.naive_bayes import GaussianNB
         gnb = GaussianNB()
         used_features =list(bos)
         used_features.pop(used_features.index('RAD'))
         print(", ".join(used_features))

         feat = 'CHAS'


         # Train classifier
         gnb.fit(
             X_train[used_features].values,
             X_train[feat]
         )
         y_pred = gnb.predict(X_test[used_features])
         #print(y_pred)
         #print(X_test.species)
         y1= (X_test[feat] != y_pred).sum()
         # Print results
         print("Number of mislabeled points")
         print(y1)
```

```
CRIM, ZN, INDUS, CHAS, NOX, RM, AGE, DIS, TAX, PTRATIO, B, LSTAT
Number of mislabeled points
0
```

```
In [31]: from sklearn.metrics import accuracy_score
         print('Accurecy of "GaussianNB" :' ,accuracy_score(X_test[feat] , y_pred))
```

```
Accurecy of "GaussianNB" : 1.0
```

```
In [32]: from sklearn.metrics import confusion_matrix
         bcm = confusion_matrix(X_test[feat], y_pred)
         print(bcm)
```

```
[[138   0]
 [  0  14]]
```