

Lab 3 - Naive Bayes Classifier

Author: Krishna Swaroop

181CO125, NITK Surathkal

▼ Introduction

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

▼ Dataset

The iris dataset is a classic and very easy multi-class classification dataset. The dataset is described below

Features	Quantity
Classes	3
Samples per Class	50
Samples Total	150
Dimensionality	4

The classes are numbered 0/1/2 and the features are real valued attributes

▼ Naive Bayes Classifier

▼ 1) Import Libraries

```
from sklearn.datasets import load_iris
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd
```

▼ 2) Load data

```
iris = load_iris()
X, y = load_iris(return_X_y=True)
```

Print first five values of numpy array to verify dataset values

```
print("Features of first five values are: ")
print(X[0:5])
print("Classes of first five values are: ", y[0:5])
```

```
Features of first five values are:
[[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]]
Classes of first five values are:  [0 0 0 0 0]
```

▼ 3) Split data

Use `train_test_split()` to split the X and y to training and testing dataset. Here, 20% of the dataset is reserved to test our algorithm

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_
```

▼ 4) Scale data proportionately

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

▼ 5) Fit model

```
from sklearn.naive_bayes import GaussianNB
nvclassifier = GaussianNB()
nvclassifier.fit(X_train, y_train)

GaussianNB(priors=None, var_smoothing=1e-09)
```

▼ 6) Predict

```
y_pred = nvclassifier.predict(X_test)
print(y_pred)

[2 2 0 0 0 2 1 1 1 1 1 2 0 0 0 0 2 1 0 1 0 2 0 2 2 1 2 0 2 1]
```

▼ 7) Calculate Accuracy

```
print("Accuracy score of Naive Bayes Model: ", nvclassifier.score(X_test, y_test))
```

```
Accuracy score of Naive Bayes Model:  0.9333333333333333
```

Therefore, we can see that Naive Bayes gives an accuracy score of 93.33%