**U19EC046 | ML | LAB 3**

**AIM**

Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

**ALGORITHM**

1. Import necessary libraries
2. Read the dataset using pandas
3. Split the dataset into features and results
4. Split the features and results into training and testing dataset using scikit learn

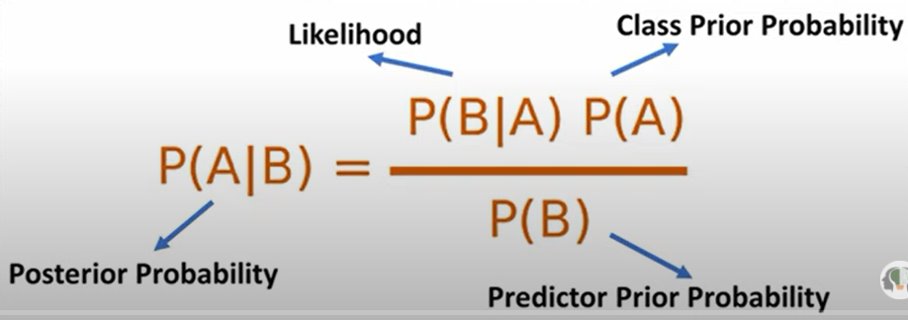
test\_train\_split

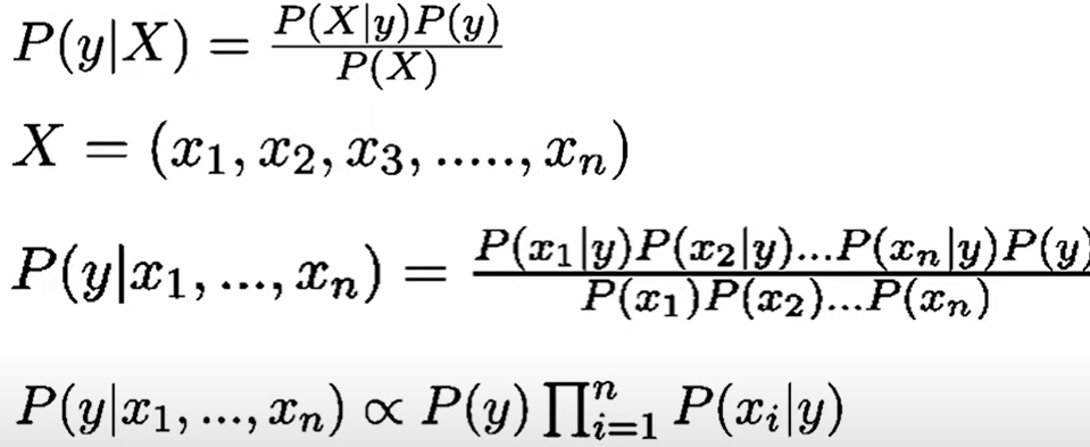
1. Apply pre-processing if needed, use scikit-learn preprocessing class
2. Create an instance of gaussian naive bais classifier
3. Train the model using fit method
4. Predict the result using predict method on model
5. Find the accuracy of model using score method on tesing dataset

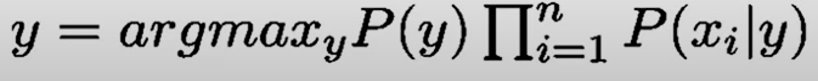
**THEORY:**

Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

**Formulas involved:**







**CODE**

1. Social Network Ads Database

|  |
| --- |
| **import pandas as pd**  **from sklearn.naive\_bayes import GaussianNB**  **from sklearn.model\_selection import train\_test\_split**  **df = pd.read\_csv('Social\_Network\_Ads.csv')**  **X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=3020)**  **from sklearn.preprocessing import StandardScaler**  **sc = StandardScaler()**  **X\_train = sc.fit\_transform(X\_train)**  **X\_test = sc.transform(X\_test)**  **classifier = GaussianNB()**  **classifier.fit(X\_train, Y\_train)**  **print(classifier.predict([X\_test[0]]))**  **classifier.score(X\_test, Y\_test)** |

1. Iris Dataset

|  |
| --- |
| **from sklearn.datasets import load\_iris**  **import numpy as np**  **data2=load\_iris()**  **df1=pd.DataFrame(np.c\_[data2.data, data2.target],columns=[list(data2.feature\_names)+['target']])**  **X2=df1.iloc[:,0:-1]**  **Y2=df1.iloc[:,-1]**  **from sklearn.model\_selection import train\_test\_split**  **X2\_train,X2\_test,Y2\_train,Y2\_test=train\_test\_split(X2,Y2,test\_size=0.15,random\_state=1000)**  **classifier.fit(X2\_train, Y2\_train)**  **classifier.score(X2\_test,Y2\_test)** |

1. Breast Cancer Dataset

|  |
| --- |
| **from sklearn.datasets import load\_breast\_cancer**  **import numpy as np**  **data2=load\_breast\_cancer()**  **df1=pd.DataFrame(np.c\_[data2.data, data2.target],columns=[list(data2.feature\_names)+['target']])**  **X2=df1.iloc[:,0:-1]**  **Y2=df1.iloc[:,-1]**  **from sklearn.model\_selection import train\_test\_split**  **X2\_train,X2\_test,Y2\_train,Y2\_test=train\_test\_split(X2,Y2,test\_size=0.15,random\_state=2020)**  **classifier.fit(X2\_train, Y2\_train)**  **classifier.score(X2\_test,Y2\_test)** |

**OUTPUT**

1. Social Network Ads database

* Accuracy : 91.25%

1. iris database

* Accuracy : 91.3%

1. Breast cancer database

* Accuracy : 97.37%

**CONCLUSION**

In this practical we have implemented code for naive baies Classifier and tested the prepared model’s accuracy on various datasets. It was observed that the accuracy of classifier increases if we perform pre-processing on dataset.