ASSIGNMENT DA-2

Roll No: 41205

Problem Statement:

Download Pima Indians Diabetes dataset. Use Naive Bayes Algorithm for classification Load the data from CSV file and split it into training and test datasets. Summarize the properties in the training dataset so that we can calculate probabilities and make predictions. Classify samples from a test dataset and a summarized training dataset.

Objective:

1. To under the Naïve Bayes algorithm
2. Implement Naïve Bayes classifier in Python

Outcome: One with be able to solve classification problems using probabilities with the Naïve Bayes algorithm

Pre-requisites:

1. 64-bit Linux OS
2. Programming Languages: Python

Hardware Specification:

1. x86_64 bit
2. 2/4 GB DDR RAM
3. 80 - 500 GB SATA HD
4. 1GB NIDIA TITAN X Graphics Card

Software Specification:

1. Ubuntu 14.04

Theory:

- Bayes Theorem: Bayes theorem is a way of finding a probability when we know certain other probabilities.
- Formula is:
  P(A|B) = P(A) P(B|A)P(B)
  P(A|B): how often A happens given that B happens, written P(A|B), P(B|A): how often B happens given that A happens, written P(B|A) P(A): and how likely A is on its own, written P(A)
  P(B): and how likely B is on its own, written P(B)
- Naive Bayes Classification: Naive Bayes is a simple, yet effective and commonly used, machine learning classifier.
- It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a

very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification and are a traditional solution for problems such as spam detection. Windows/Linux Operating Systems, RStudio, jdk.

- Confusion Matrix: A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most
- Class 1 : Positive
- Class 2 : Negative
- Definition of the Terms:
    - Positive (P) : Observation is positive (for example: is an apple).
    - Negative (N) : Observation is not positive (for example: is not an apple).
    - True Positive (TP) : Observation is positive, and is predicted to be positive.
    - False Negative (FN) : Observation is positive, but is predicted negative.
    - True Negative (TN) : Observation is negative, and is predicted to be negative.
    - False Positive (FP) : Observation is negative, but is predicted positive.
- Classification Rate/Accuracy:
    - Rate or Accuracy is given by the relation:
    - Accuracy = (TP + TN) / (TP + TN + FP + FN)

- Recall:
    - Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (small number of FN).
    - Recall is given by the relation:
    - Recall=TP / (TP + FN)= Recall=TP / (TP + FN)
- Precision:
    - To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labeled as positive is indeed positive (small number of FP).
    - Precision is given by the relation:
    - Precision = TP / (TP + FP)

Input: PIMA Indians Diabetes Dataset for classification

Output:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

	precision	recall	f1-score	support
0	0.83	0.84	0.83	51
1	0.68	0.65	0.67	26
accuracy			0.78	77
macro avg	0.75	0.75	0.75	77
weighted avg	0.78	0.78	0.78	77

Conclusion: Thus, we have created a naïve bayes classifier and classified the PIMA Indians dataset using it.