Assignment No. B12

1 Problem Definition

Implement Naive Bayes for Concurrent/Distributed application. Approach should handle categorical and continuous data.

2 Learning Objectives:

1. To understand the Naive Bayes Classification algorithm.
2. To learn handling classification of categorical and continuous data.

3 Software and Hardware requirements:

1. Open source 64 bit OS.
2. Text editor.
3. Python 2.7

4 Theory

The objective of classification is to analyze the input data and to develop an accurate description or model for each class using the features present in the data. This model is used to classify test data for which the class descriptions are not known. The input data, also called the training set, consists of multiple records each having multiple attributes or features. Each record is tagged with a class label.
The data analysis task is called as classification, where a model or classifier is constructed to predict categorical labels. (e.g. safe or risky, yes or no) Data classification is a two-step process 1. Learning 2. Classification

1. Learning: In this step, a classifier is built describing a predetermined set of data classes or concepts. This first step of the classification process can also be viewed as the learning of a mapping or function, y = f(X), that can predict the associated class label y of a given tuple X. This mapping is represented in the form of classification rules, decision trees or mathematical formula.
2. Classification: Model created in previous steps are used for classification. Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples. Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class.Bayesian classification is based on Bayes theorem, Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.
Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered naive. (Hence it is Bayesian classifier is also called as Naive Bayes Classifier)

Baye's Theorem

Bayes theorem is named after Thomas Bayes, who did early work in probability and decision theory during the 18th century. Let X be a data tuple. In Bayesian terms, X is considered evidence.
Let H be some hypothesis, such as that the data tuple X belongs to a specified class C. For classification problems, we want to determine P(H—X), the probability that the hypothesis H holds given the evidence or observed data tuple X.
In other words, we are looking for the probability that tuple X belongs to class C, given that we know the attribute description of X.
1 P(H) is the posterior probability, or a posteriori probability, of H conditioned on X.
i.e. P(H—X) : Probability of H given X
P(H—X) reflects the probability that customer X will buy a computer given that we know the customers age and income. 2) P(X—H) is the posterior probability of X conditioned on H.
i.e P(X—H) : Probability of X given H
That is, it is the probability that customers, X, is 35 years old and earns dollar 40,000, given that we know the customer will buy a computer.

3) P(H) is the prior probability, or a priori probability, of H.

i.e. P(H) : Prior probability of hypothesis H
For example, this is the probability that any given customer will buy a computer, regardless of age, income, or any other information, for that matter.
4) P(X) is the prior probability of X.

i.e. P(X) : Prior probability of data tuple X

For example, it is the probability that a person from our set of customers is 35 years old and earns dollar 40,000.
Bayes theorem is useful in that it provides a way of calculating the posterior probability, P(H—X), from P(H), P(X—H), and P(X). Bayes theorem is given below:
P (H—X)=P(X —H)P(H)/P(X)

Bayesian classifier
The nave Bayesian classifier, or simple Bayesian classifier, works as follows:
1) Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector X = (x1, x2, , xn)
2) Suppose there are m classes C1, C2, , Cm. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the nave Bayesian classifier predicts that tuple X belongs to the class Ci if and only if,

$$P(C_i|\boldsymbol{X}) > P(C_j|\boldsymbol{X}) \quad \text{for } 1 \leq j \leq m, j \neq i.$$

Thus we maximize P(Ci—X). The class Ci for which P(Ci—X) is maximized is called the maximum posteriori hypothesis. This can be derived from Bayes theorem :

$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})}.$$

As P(X) is constant for all classes, only P(XCi)*P(Ci) need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, P(C1) = P(C2) = = P(Cm), and we would therefore maximize P(X— Ci).
Otherwise, we maximize P(X— Ci)*P(Ci).

4) Given data sets with many attributes, it would be extremely computationally expensive to compute P(X— Ci). Thus,

$$\begin{aligned} P(\boldsymbol{X}|C_i) &= \prod_{k=1}^{n} P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i). \end{aligned}$$

Here xk refers to the value of attribute Ak for tuple X. For each attribute, we look at whether the attribute is categorical or continuous-valued.
For instance, to compute P(X—Ci), we consider the following:
a) If Ak is categorical, then P(xk —Ci) is the number of tuples of class Ci in D having the value xk for Ak , divided by -Ci,D-, the number of tuples of class Ci in D.
b) If Ak is continuous-valued, then A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ.
5) In order to predict the class label of X, P(X—Ci)P(Ci) is evaluated for each class Ci. The classifier predicts that the class label of tuple X is the class Ci if and only if P(X—Ci)P(Ci) ¿ P(X—Cj)P(Cj) for 1 ¡= j ¡= m, j = i.
In other words, the predicted class label is the class Ci for which P(X—Ci)P(Ci) is the maximum.

Following formulas are used for predicting the accurate worktype for a person.

1. Probability density formula.
   $f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$.

2. formula for calculating mean.
   $$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

3. formula for calculating standard deviation.
   $$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2}$$

5 Related Mathematics

Let S be the solution perspective of the given problem.
The set S is defined as:
$S = \{\ s, e, X, Y, F, DD, NDD | \varnothing_s\}$
Where,
s= Start point
s= A, W, P
where,
A = set of item ids.
W = corresponding set of weights.
P = corresponding set of profit.
e= End point
e= maximum possible profit within the weight limit found.
F= Set of main functions

F = $\{f_{bound}, f_{knapsack}\}$

f_{bound} :function to calculate bounds for item.

$f_{knapsack}$:function to add items in the knapsack for maximum profit.

X= Input Set.
$X = \{\ x_1, ..., x_n, p_1..., p_n, w_1...w_n\}$
where,
x_i= items
p_i= corresponding profits
w_i= corresponding weights

Y = $\{K, prof\}$
where,
prof = total profit of the knapsack.
K = set of final knapsack contents.
DD= set of deterministic data
NDD= set of non deterministic data

DD = $\{x_1, x_2, ..., x_n, k\}$
NDD = $\{initialcentroids\}$

6 Conclusion

Thus we implemented Naive Bayes classification for both categorical and continuous data.