PROBLEM STATEMENT :

Implement Naive Bayes to predict the work type for a person with following parameters: age: 30,Qualification: MTech, Experience: 8
Following table provides the details of the available data:

Work Type	Age	Qualification	Experience
Consultancy	30	Ph.D.	9
Service	21	MTech.	1
Research	26	MTech.	2
Service	28	BTech.	10
Consultancy	40	MTech.	14
Research	35	Ph.D.	10
Research	27	BTech.	6
Service	32	MTech.	9
Consultancy	45	Btech.	17
Research	36	Ph.D.	7

OBJECTIVE :

• To apply algorithmic strategies while solving problems
• To develop time and space efficient algorithms
• To study algorithmic examples in distributed, concurrent and parallel environments

THEORY :

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

1. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X = (x 1 , x 2 , . . . , x n), depicting n measurements made on the tuple from n attributes, respectively, A 1 , A 2 , . . . , A n .

2. Suppose that there are m classes, C 1 , C 2 , . . . , C m . 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 naïve Bayesian classifier predicts that tuple X belongs to the class C i if and only if

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

Thus we maximize P(C i |X). The class C i for which P(C i |X) is maximized is called the maximum posteriori hypothesis. By Bayes' theorem,

$$P(C_i|X) = [P(X|C_i)P(C_i)]/P(X)$$

As P(X) is constant for all classes, only P(X|C i)P(C i) 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(C 1) = P(C 2) = · · · = P(C m), and we would therefore maximize P(X|C i). Otherwise, we maximize P(X|C i)P(C i). Note that the class prior probabilities may be estimated by P(C i) = |C i , D |/|D|, where |C i , D | is the number of training tuples of class C i in D.

Given data sets with many attributes, it would be extremely computationally expensive to compute P(X|C i). In order to reduce computation in evaluating P(X|C i), the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$\mathbf{P(X|C\,i) = \prod_{k=1}^{n} P(x\,k|C\,i)}$$
$$\mathbf{= P(x\,1|C\,i) \times P(x\,2|C\,i) \times \cdots \times P(x\,n|C\,i).}$$

We can easily estimate the probabilities P(x 1 |C i), P(x 2 |C i), . . . , P(x n |C i) from the training tuples. Recall that here x k refers to the value of attribute A k for tuple X. For each attribute, we look at whether the attribute is categorical or continuous-valued. For instance, to compute P(X|C i), we consider the following:

(a) If A k is categorical, then P(x k |C i) is the number of tuples of class C i in D having the value x k for A k , divided by |C i , D |, the number of tuples of class C i in D.

(b) If A k is continuous-valued, then we need to do a bit more work, but the calculation is pretty straightforward. A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ, defined by

so that,

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$
$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

These equations may appear daunting, but hold on! We need to compute μ C i and σ C i , which are the mean (i.e., average) and standard deviation, respectively, of the values of attribute A k for training tuples of class C i . We then plug these two quantities into Equation (6.13), together with x k , in order to estimate P(x k |C i).

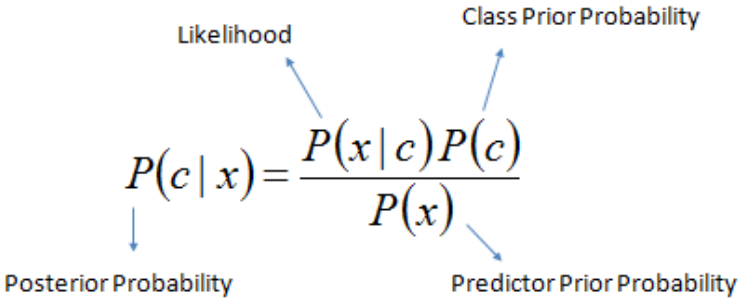
In order to predict the class label of X, P(X|C i)P(C i) is evaluated for each class C i . The classifier predicts that the class label of tuple X is the class C i if and only if

P(X|C i)P(C i) > P(X|C j)P(C j) for 1 ≤ j ≤ m, j = i.

In other words, the predicted class label is the class C i for which P(X|C i)P(C i) is the maximum.

ALGORITHM :

Bayes theorem provides a way of calculating the posterior probability, $P(c|x)$, from $P(c)$, $P(x)$, and $P(x|c)$. Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*

INPUT :
Enter age , Qualification and Experience

EXPECTED OUTPUT :

Predicted worktype

MATHMODEL

D : Set of tuples
- Each Tuple is an 'n' dimensional attribute vector
- X : (x1,x2,x3,.... xn)

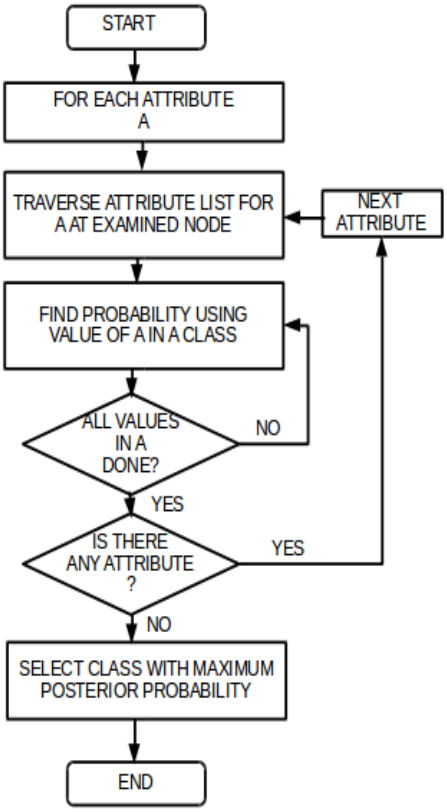
Let there be 'm' Classes : C1,C2,C3...Cm
Naïve Bayes classifier predicts X belongs to Class Ci iff
- P (Ci/X) > P(Cj/X) for 1<= j <= m , j <> i

Maximum Posteriori Hypothesis
- P(Ci/X) = P(X/Ci) P(Ci) / P(X)
- Maximize P(X/Ci) P(Ci) as P(X) is constant

With many attributes, it is computationally expensive to evaluate P(X/Ci).
Naïve Assumption of "class conditional independence"

- $P (X / . Ci) = \prod_{k=1}^{n} P (x k / Ci)$
- P(X/Ci) = P(x1/Ci) * P(x2/Ci) *...* P(xn/ Ci)

FLOWCHART :



TEST CASES :

TEST CASE	INPUT	EXPECTED OUTPUT	OUTPUT ACHIEVED	REMARKS
1.	ENTER AGE :30 ENTER QUALIFICATION :M.Tech ENTER EXPERIENCE :8	Research	Research	Correct
2.	ENTER AGE :40 ENTER QUALIFICATION :M.Tech ENTER EXPERIENCE :15	Consultancy	Consultancy	Correct
3.	ENTER AGE :40 ENTER QUALIFICATION :P.hD ENTER EXPERIENCE :8	Research	Research	Correct

SPACE AND TIME COMPLEXITIES :

In Baye's classification we use three nested loops which iterate over the entire length of the dataset. Hence time complexity is $O(N^3)$.

CONCLUSION :

Hence we have successfully implemented the naive bayes algorithm to predict the work type of the person.

OUTCOMES ACHIEVED :

COURSE OUTCOME	**ACHIEVED(√)**
Problem solving abilities for smart devices.	
Problem solving abilities for gamifications.	
Problem solving abilities of pervasiveness,embedded security and NLP.	
To solve problems for multicore or distributed,concurrent/Parallel environments	√