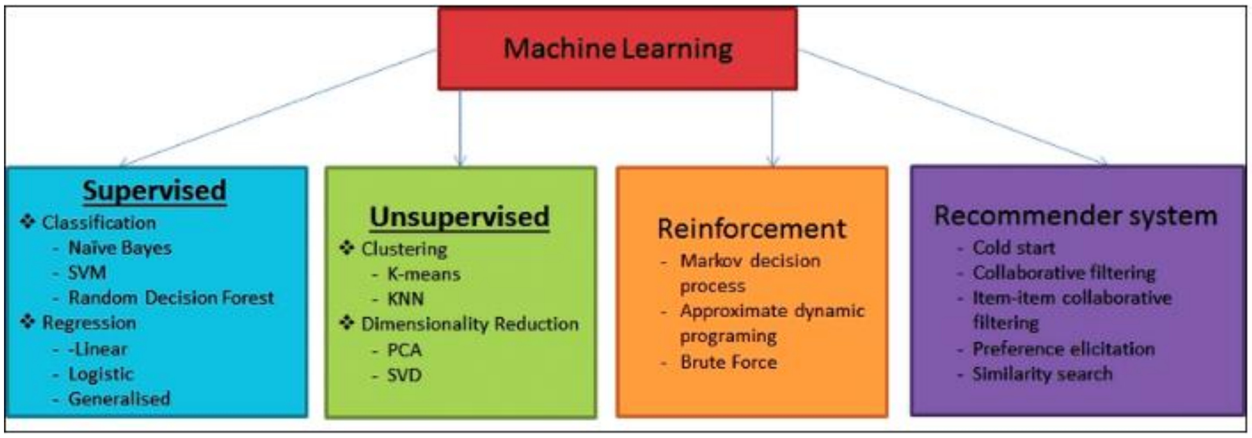
National University of Computer and Emerging Sciences

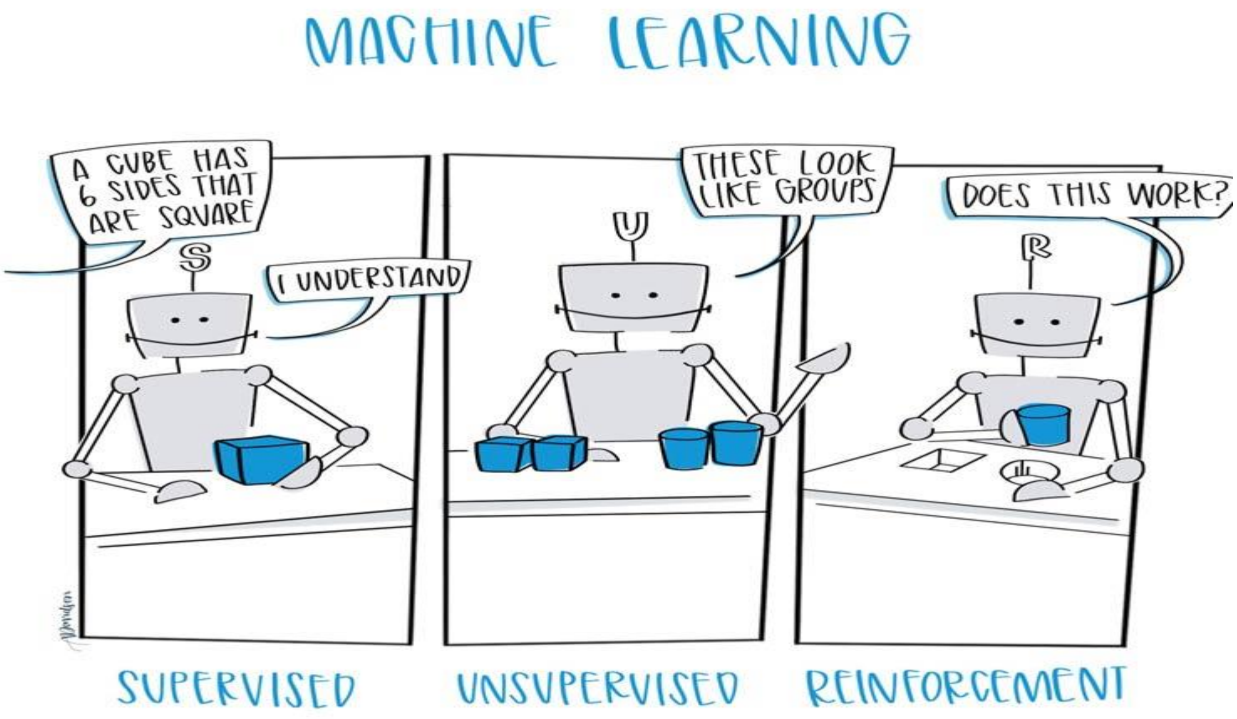
CL 461 Artificial Intelligence

Lab Manual 09

Introduction to Machine Learning Naïve Bayes

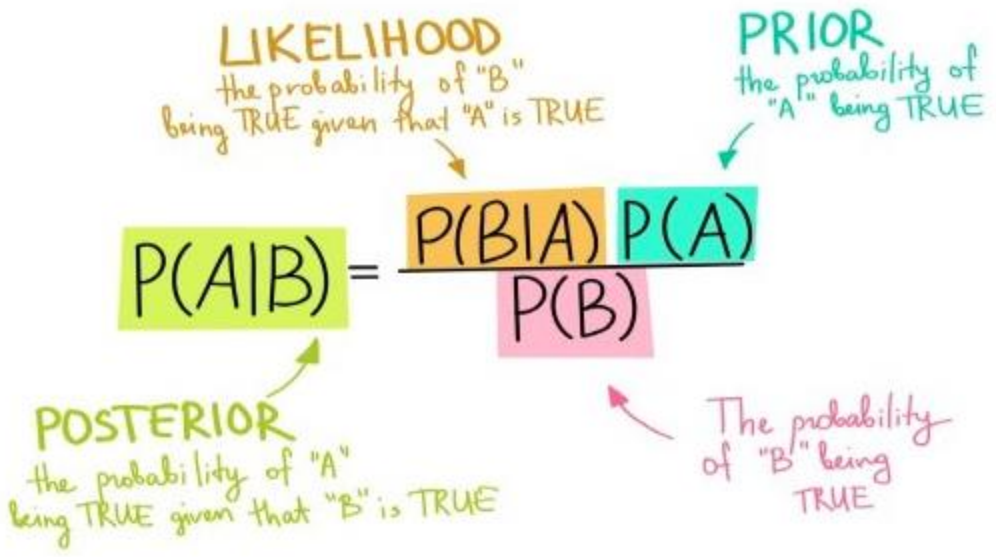
Machine Learning:





What is a Bayes Theorem?

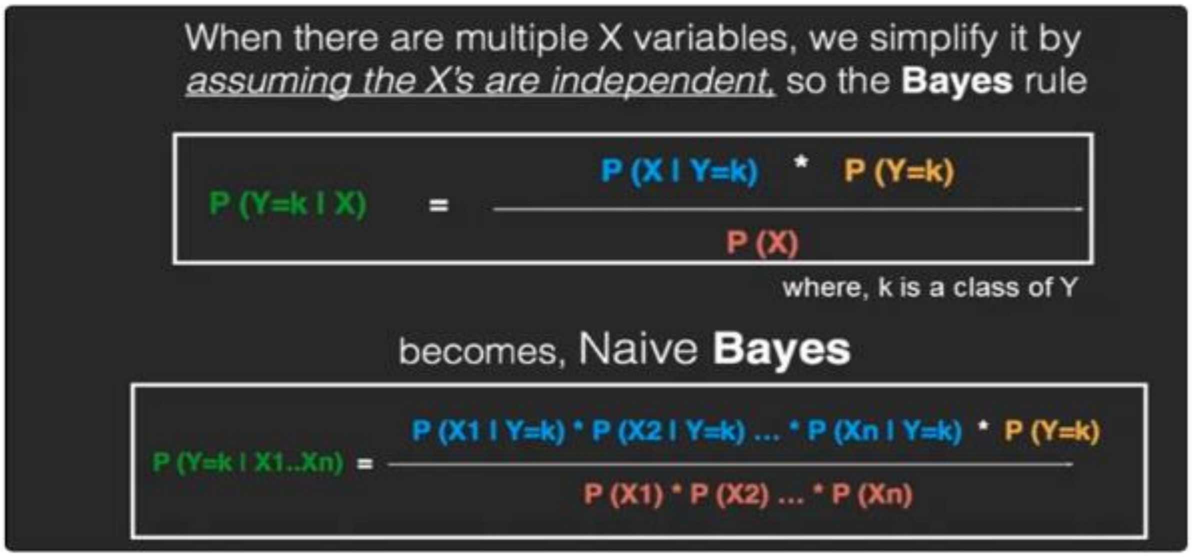
Bayes' Theorem is a way of finding a probability when we know certain other probabilities.


LIKELIHOOD
the probability of "B" being TRUE given that "A" is TRUE
PRIOR
the probability of "A" being TRUE
$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$
POSTERIOR
the probability of "A" being TRUE given that "B" is TRUE
The probability of "B" being TRUE

What is a Naïve Bayes algorithm?

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

It is called 'Naive' because of the naive assumption that the X's are independent of each other. Regardless of its name, it's a powerful formula.



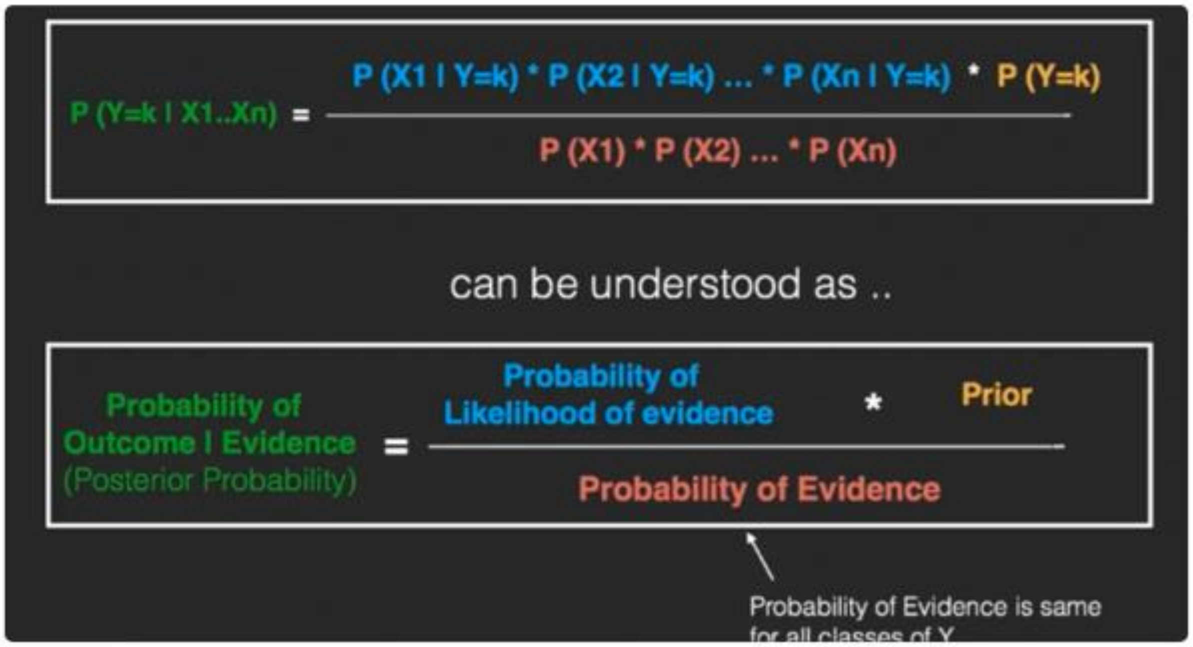
When there are multiple X variables, we simplify it by *assuming the X's are independent*, so the **Bayes** rule

$$\boxed{P(Y=k \mid X) = \frac{P(X \mid Y=k) * P(Y=k)}{P(X)}}$$

where, k is a class of Y

becomes, Naive **Bayes**

$$\boxed{P(Y=k \mid X1..Xn) = \frac{P(X1 \mid Y=k) * P(X2 \mid Y=k) \ldots * P(Xn \mid Y=k) * P(Y=k)}{P(X1) * P(X2) \ldots * P(Xn)}}$$



$$P(Y{=}k \mid X1..Xn) = \frac{P(X1 \mid Y{=}k) * P(X2 \mid Y{=}k) \ldots * P(Xn \mid Y{=}k) * P(Y{=}k)}{P(X1) * P(X2) \ldots * P(Xn)}$$

can be understood as ..

$$\text{Probability of Outcome} \mid \text{Evidence (Posterior Probability)} = \frac{\text{Probability of Likelihood of evidence} * \text{Prior}}{\text{Probability of Evidence}}$$

Probability of Evidence is same for all classes of Y

Naïve Bayes example by hand?

Say you have 1000 fruits which could be either ‘banana’, ‘orange’ or ‘other’. These are the 3 possible classes of the Y variable.

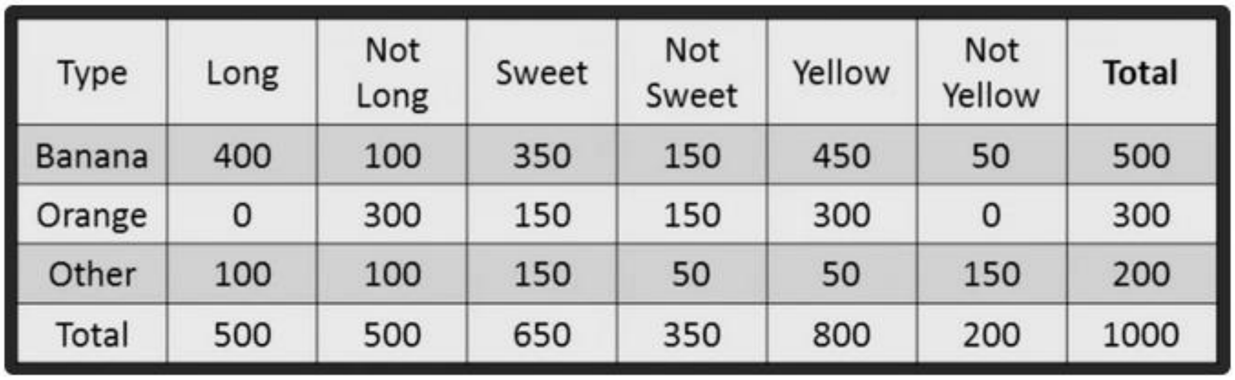
We have data for the following X variables, all of which are binary (1 or 0).

- Long
- Sweet
- Yellow

The first few rows of the training dataset look like this:

Fruit	Long (x1)	Sweet (x2)	Yellow (x3)
Orange	0	1	0
Banana	1	0	1
Banana	1	1	1
Other	1	1	0
..	..	..	..

For the sake of computing the probabilities, let’s aggregate the training data to form a counts table like this.



Type	Long	Not Long	Sweet	Not Sweet	Yellow	Not Yellow	**Total**
Banana	400	100	350	150	450	50	500
Orange	0	300	150	150	300	0	300
Other	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

So the objective of the classifier is to predict if a given fruit is a ‘Banana’ or ‘Orange’ or ‘Other’ when only the 3 features (long, sweet and yellow) are known.

The idea is to compute the 3 probabilities, that is the probability of the fruit being a banana, orange or other. Whichever fruit type gets the highest probability wins.

Compute Prior probabilities:

P(Y=Banana) = 500 / 1000 = 0.50

P(Y=Orange) = 300 / 1000 = 0.30

P(Y=Other) = 200 / 1000 = 0.20

Compute Probability of Evidence:

P(x1=Long) = 500 / 1000 = 0.50

P(x2=Sweet) = 650 / 1000 = 0.65

P(x3=Yellow) = 800 / 1000 = 0.80

Compute Likelihood probabilities:

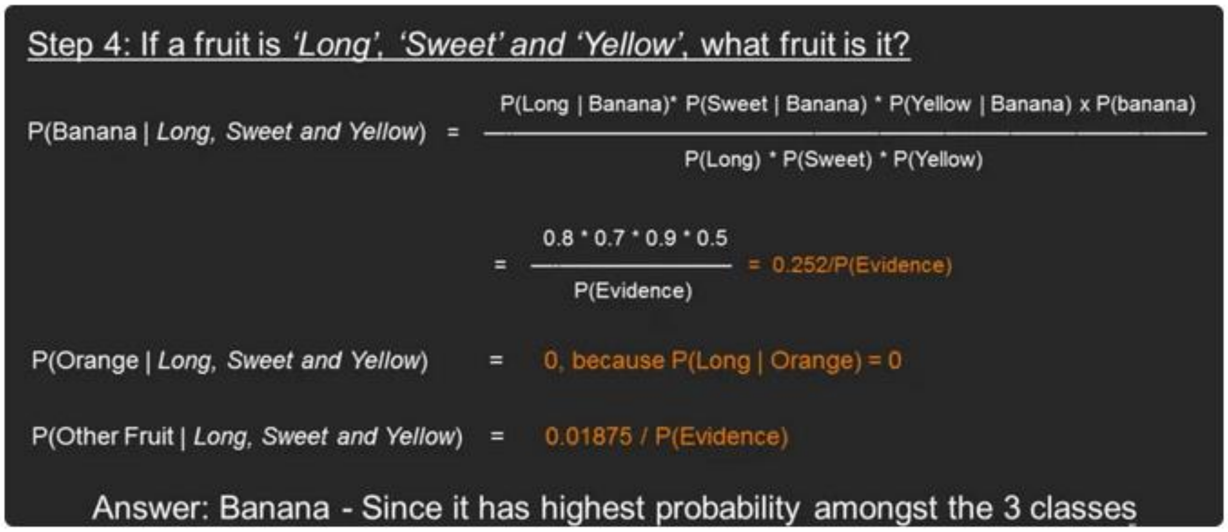
P(x1=Long | Y=Banana) = 400 / 500 = 0.80

P(x2=Sweet | Y=Banana) = 350 / 500 = 0.70

P(x3=Yellow | Y=Banana) = 450 / 500 = 0.90

So, the overall probability of Likelihood of evidence for Banana = 0.8 * 0.7 * 0.9 = 0.504

Substituting all:



Step 4: If a fruit is *'Long'*, *'Sweet' and 'Yellow'*, what fruit is it?

$$P(\text{Banana} \mid \textit{Long, Sweet and Yellow}) = \frac{P(\text{Long} \mid \text{Banana}) * P(\text{Sweet} \mid \text{Banana}) * P(\text{Yellow} \mid \text{Banana}) \times P(\text{banana})}{P(\text{Long}) * P(\text{Sweet}) * P(\text{Yellow})}$$

$$= \frac{0.8 * 0.7 * 0.9 * 0.5}{P(\text{Evidence})} = 0.252/P(\text{Evidence})$$

$P(\text{Orange} \mid \textit{Long, Sweet and Yellow}) =$ 0, because $P(\text{Long} \mid \text{Orange}) = 0$

$P(\text{Other Fruit} \mid \textit{Long, Sweet and Yellow}) = 0.01875 / P(\text{Evidence})$

Answer: Banana - Since it has highest probability amongst the 3 classes

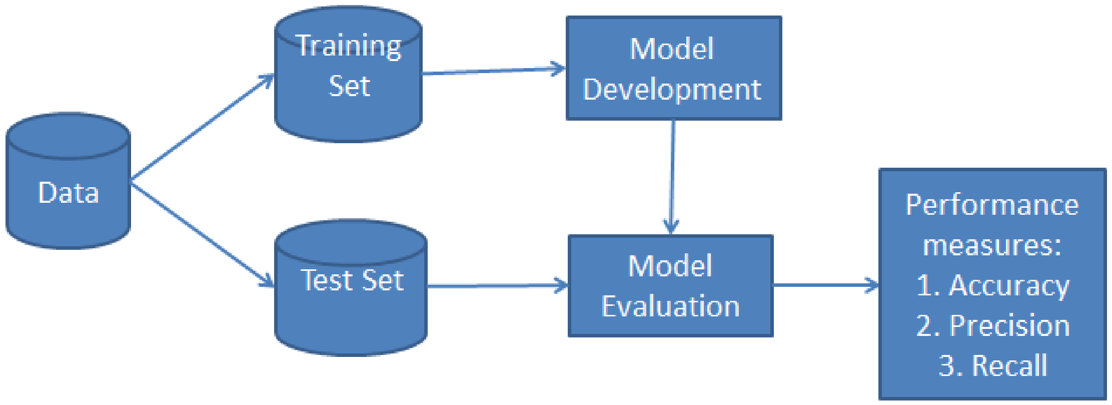
Naïve Bayes Classifier

Naive Bayes is the most straightforward and fast classification algorithm, which is suitable for a large chunk of data. Naive Bayes classifier is successfully used in various applications such as spam filtering, text classification, sentiment analysis, and recommender systems. It uses Bayes theorem of probability for prediction of unknown class.

Classification Workflow

Whenever you perform classification, the first step is to understand the problem and identify potential features and label. Features are those characteristics or attributes which affect the results of the label. For example, in the case of a loan distribution, bank’s manager identify customer’s occupation, income, age, location, previous loan history, transaction history, and credit score. These characteristics are known as features which help the model classify customers.

The classification has two phases, a learning phase, and the evaluation phase. In the learning phase, classifier trains its model on a given dataset and in the evaluation phase, it tests the classifier performance. Performance is evaluated on the basis of various parameters such as accuracy, error, precision, and recall.



Classifier Building in Scikit-learn

In this example, you can use the dummy dataset with three columns: weather, temperature, and play. The first two are features (weather, temperature) and play is the label/single feature.

For Python example see the attached Example#1(Naive Bayes).ipynb file.

Naive Bayes with Multiple Labels

Till now you have learned Naive Bayes classification with binary labels. Now you will learn about multiple class classification in Naive Bayes. Which is known as multinomial Naive Bayes classification. For example, if you want to classify a news article about technology, entertainment, politics, or sports.

In model building part, you can use wine dataset which is a very famous multi-class classification problem. "This dataset is the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars."

Dataset comprises of 13 features (alcohol, malic_acid, ash, alcalinity_of_ash, magnesium, total_phenols, flavanoids, nonflavanoid_phenols, proanthocyanins, color_intensity, hue, od280/od315_of_diluted_wines, and proline) and type of wine cultivar. This data has three type of wine Class_0, Class_1, and Class_3. Here you can build a model to classify the type of wine. (The dataset is available in the scikit-learn library)
For Python example see the attached Example#2(Naive Bayes).ipynb file.

LAB TASKS

- Task#1
  Implement the Naïve Bayes on the dataset given in this link.
  https://towardsdatascience.com/a-mathematical-explanation-of-naive-bayes-in-5-minutes-44adebcdb5f8

- Task#2
  Replicate this code on to a different dataset of your own choice.
  https://towardsdatascience.com/implementing-naive-bayes-algorithm-from-scratch-python-c6880cfc9c41

- Task#3
  Implement spam filter using Naïve Bayes. (with the help of internet)

- Task#4
  Reading Task
  https://towardsdatascience.com/bayes-theorem-101-example-solution-ff54147d6c7f