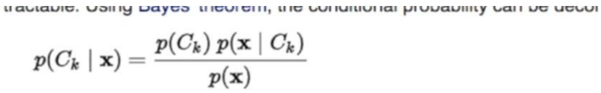
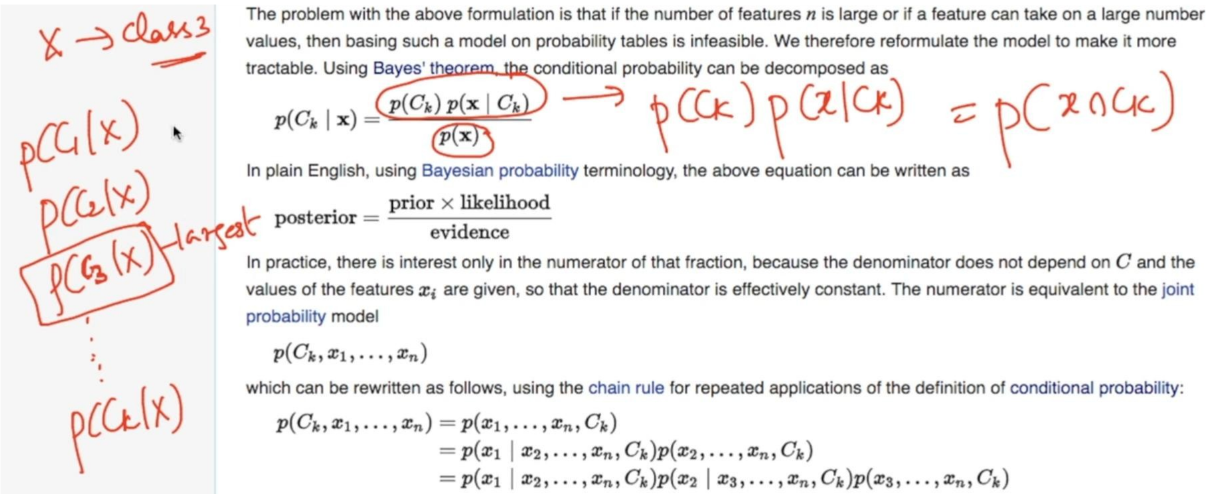
We need to find probability of a class label given X

X is n-d



tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$



The problem with the above formulation is that if the number of features n is large or if a feature can take on a large number values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

In plain English, using Bayesian probability terminology, the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on C and the values of the features x_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

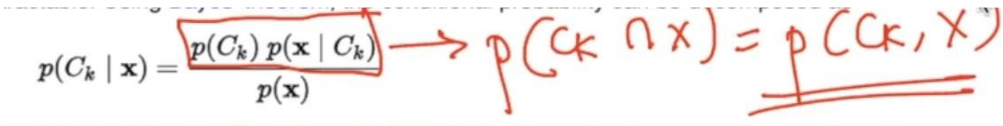
$$p(C_k, x_1, \ldots, x_n)$$

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned} p(C_k, x_1, \ldots, x_n) &= p(x_1, \ldots, x_n, C_k) \\ &= p(x_1 \mid x_2, \ldots, x_n, C_k)\, p(x_2, \ldots, x_n, C_k) \\ &= p(x_1 \mid x_2, \ldots, x_n, C_k)\, p(x_2 \mid x_3, \ldots, x_n, C_k)\, p(x_3, \ldots, x_n, C_k) \end{aligned}$$

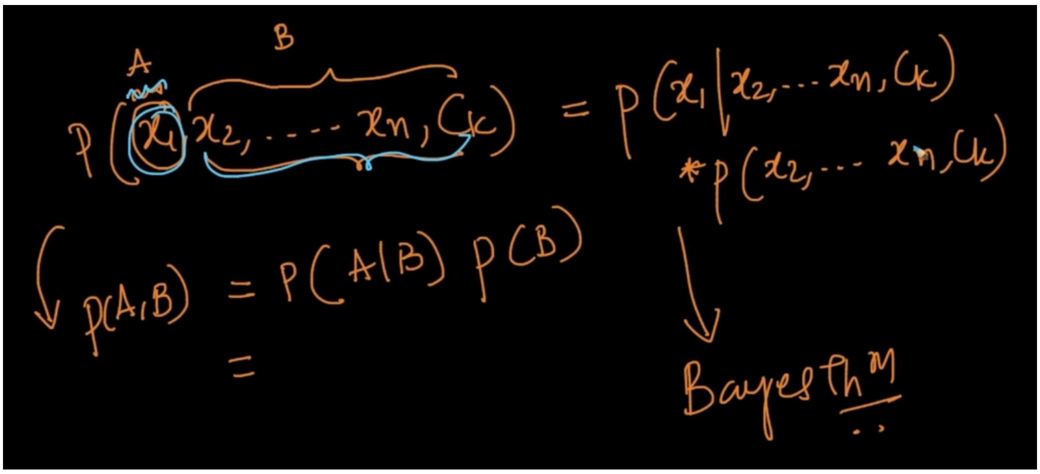
whichever class label's probability is the highest, we'll pick that one

denominator is constant, we can ignore it



$$p(C_k \mid \mathbf{x}) = \frac{\boxed{p(C_k)\, p(\mathbf{x} \mid C_k)}}{p(\mathbf{x})}$$

the above is called a joint probability



$$p(C_k, x_1, \dots, x_n)$$

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \checkmark \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2, \dots, x_n, C_k) \checkmark \rightarrow \text{defn. of cond} \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k)\, p(x_3, \dots, x_n, C_k) \\ &= \dots \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k) \dots p(x_{n-1} \mid x_n, C_k)\, p(x_n \mid C_k)\, p(C_k) \end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature x_i is conditionally independent of every other feature x_j for $j \neq i$, given the category C. This means that

$$p(x_i \mid x_{i+1}, \dots, x_n, C_k) = p(x_i \mid C_k)\,.$$

Thus, the joint model can be expressed as

$$\begin{aligned} p(C_k \mid x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\ &\propto p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k) \cdots \\ &\propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,. \end{aligned}$$

$$p(\overset{A}{x_2}, \underbrace{x_3, \dots, x_n, C_k}_{B})$$

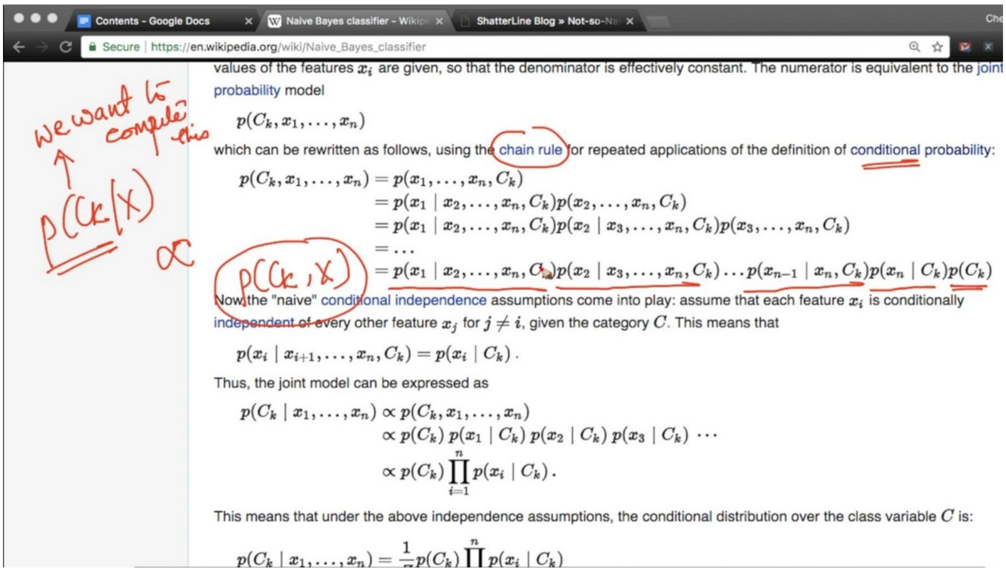
$$= p(x_2 \mid x_3, x_4, \dots x_n, C_k)$$

$$* p(x_3, x_4, \dots x$$

we want to compute the term in red

-------------->

which is proprotional to what we just computed



values of the features x_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$p(C_k, x_1, \dots, x_n)$$

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k)\, p(x_3, \dots, x_n, C_k) \\ &= \dots \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k) \dots p(x_{n-1} \mid x_n, C_k)\, p(x_n \mid C_k)\, p(C_k) \end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature x_i is conditionally independent of every other feature x_j for $j \neq i$, given the category C. This means that

$$p(x_i \mid x_{i+1}, \dots, x_n, C_k) = p(x_i \mid C_k)\,.$$

Thus, the joint model can be expressed as

$$\begin{aligned} p(C_k \mid x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\ &\propto p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k) \cdots \\ &\propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,. \end{aligned}$$

This means that under the above independence assumptions, the conditional distribution over the class variable C is:

$$p(C_k \mid x_1, \dots, x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

we assume conditional independence

for ex if we assume conditional independence among A and B

means that

$$P(A \mid B) = P(A)$$

$$P(A \mid \underbrace{B, C}_{B \cap C}) = P(A \mid C)$$

$$p(C_k, x_1, \ldots, x_n)$$

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned}
p(C_k, x_1, \ldots, x_n) &= p(x_1, \ldots, x_n, C_k) \\
&= p(x_1 \mid x_2, \ldots, x_n, C_k) p(x_2, \ldots, x_n, C_k) \\
&= p(x_1 \mid x_2, \ldots, x_n, C_k) p(x_2 \mid x_3, \ldots, x_n, C_k) p(x_3, \ldots, x_n, C_k) \\
&= \ldots \\
&= p(x_1 \mid x_2, \ldots, x_n, C_k) p(x_2 \mid x_3, \ldots, x_n, C_k) \ldots p(x_{n-1} \mid x_n, C_k) p(x_n \mid C_k) p(C_k)
\end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature x_i is conditionally independent of every other feature x_j for $j \neq i$, given the category C. This means that

$$p(x_i \mid x_{i+1}, \ldots, x_n, C_k) = p(x_i \mid C_k).$$ ⟹ x_i is indep (cond) of $x_{i+1}, x_{i+2}, \ldots x_n$ given C_k

Thus, the joint model can be expressed as

$$\begin{aligned}
p(C_k \mid x_1, \ldots, x_n) &\propto p(C_k, x_1, \ldots, x_n) \\
&\propto p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k) \cdots \\
&\propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).
\end{aligned}$$

$$p(x_i \mid x_{i+1}, \ldots, x_n, C_k) = p(x_i \mid C_k).$$

Thus, the joint model can be expressed as

$$\begin{aligned}
p(C_k \mid x_1, \ldots, x_n) &\propto p(C_k, x_1, \ldots, x_n) \\
&\propto p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k) \cdots \\
&\propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).
\end{aligned}$$

This means that under the above independence assumptions, the conditional distribution over the class variable C is:

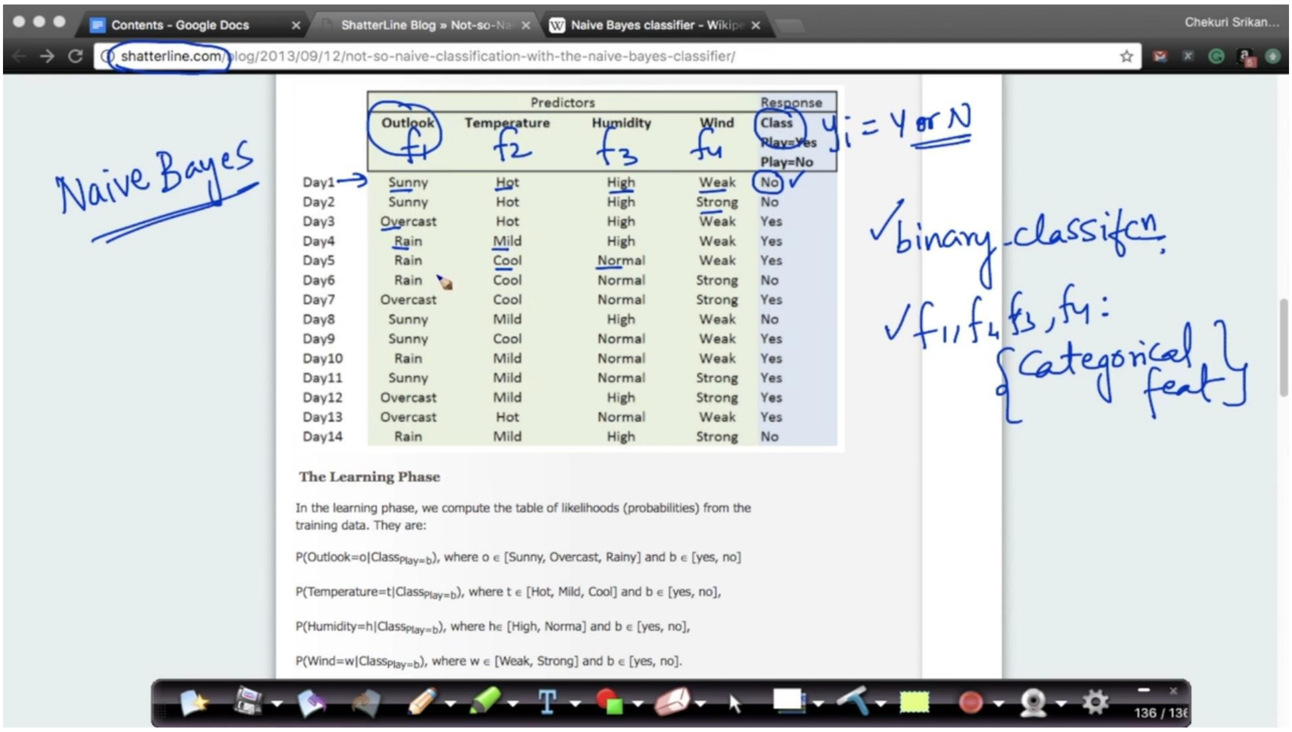
$$p(C_k \mid x_1, \ldots, x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

where the evidence $Z = p(\mathbf{x}) = \sum_k p(C_k)\, p(\mathbf{x} \mid C_k)$ is a scaling factor dependent only on $x_1, \ldots, x_n$, that is, a constant if the values of the feature variables are known.

Constructing a classifier from the probability model [edit]

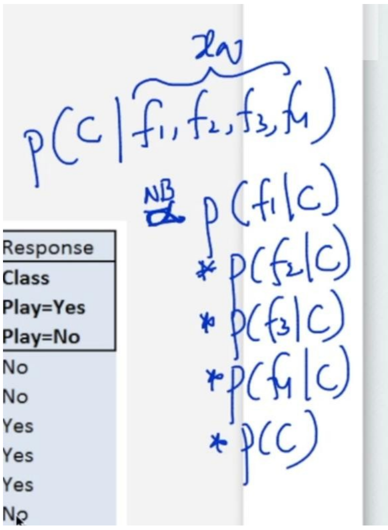
The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posteriori* or *MAP* decision rule. The corresponding classifier, a Bayes classifier, is the function that assigns a class label $\hat{y} = C_k$ for some k as follows:

$$\hat{y} = \underset{k \in \{1, \ldots, K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$

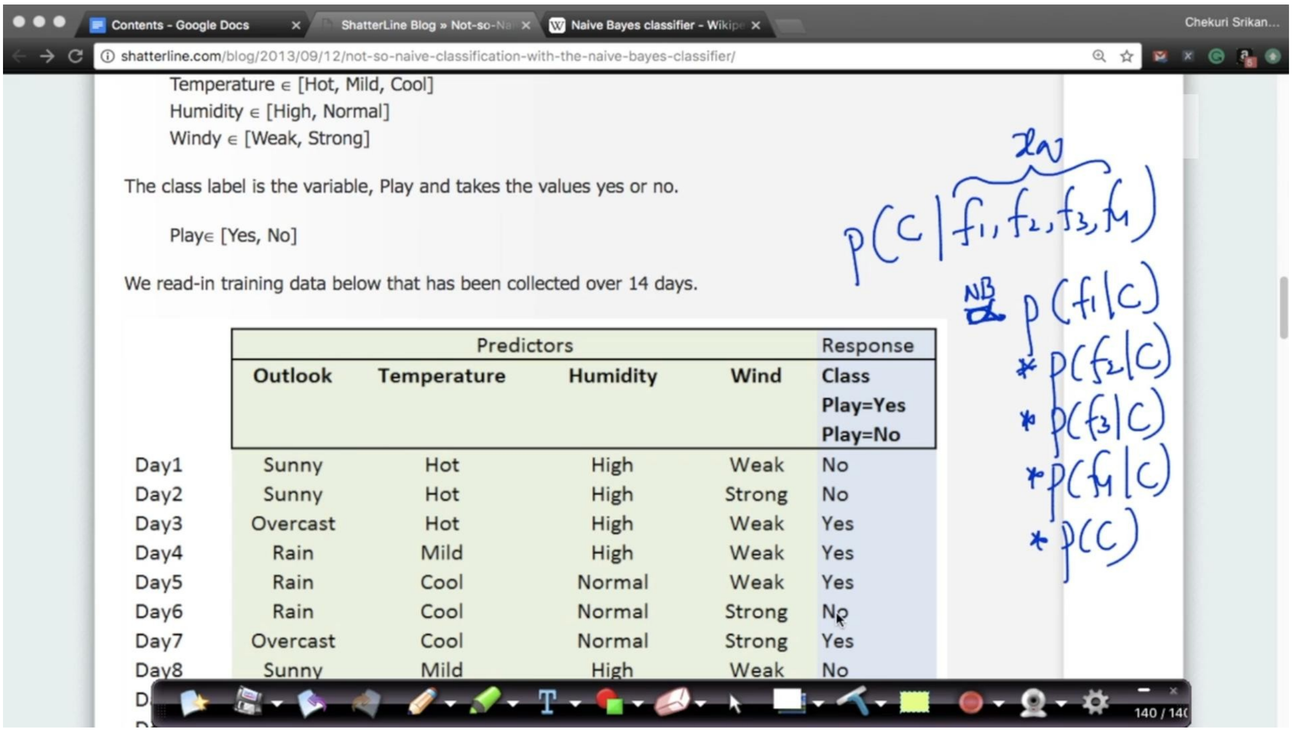


we compute stuff in learning phase

remember we need to compute this, (NB just stands for naive bayes)



P(c) is easy to compute, just count how many yes or no are there



Temperature ∈ [Hot, Mild, Cool]
Humidity ∈ [High, Normal]
Windy ∈ [Weak, Strong]

The class label is the variable, Play and takes the values yes or no.

Play∈ [Yes, No]

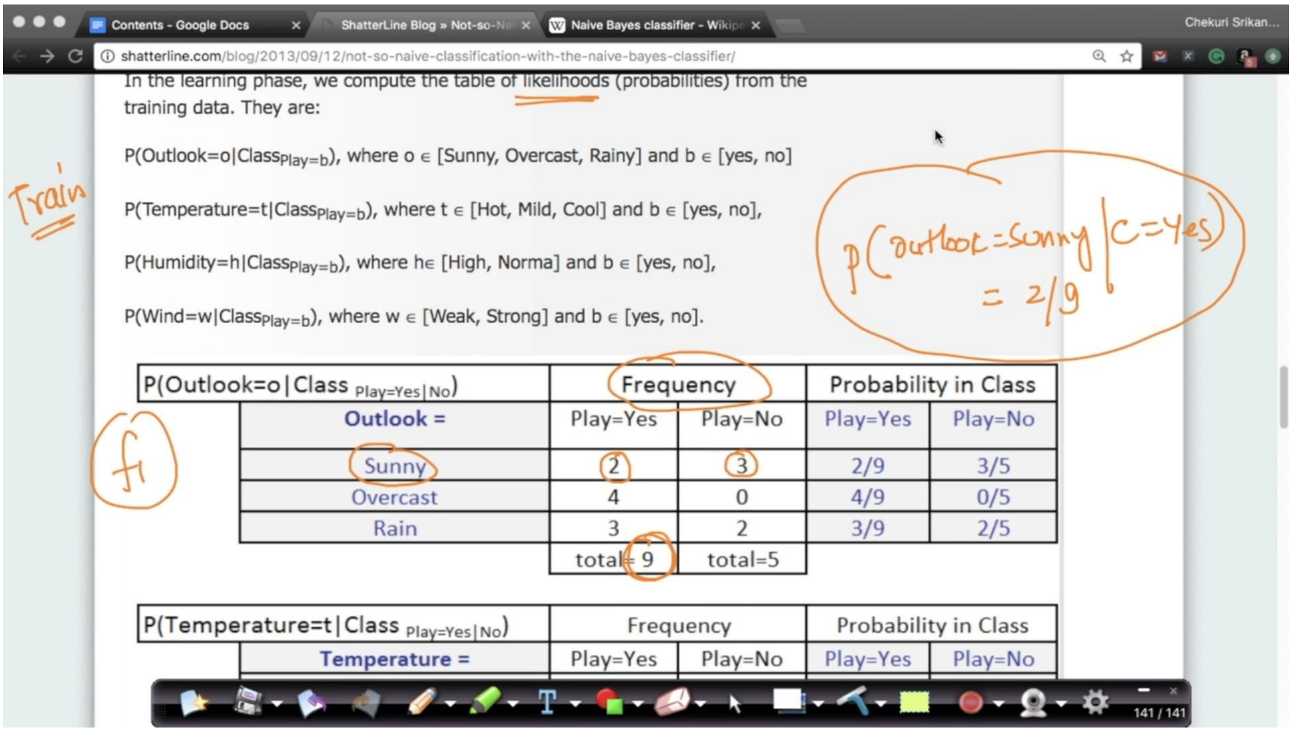
We read-in training data below that has been collected over 14 days.

	Predictors				Response
	Outlook	Temperature	Humidity	Wind	Class Play=Yes Play=No
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No

$$p(C \mid \underbrace{f_1, f_2, f_3, f_4}_{x_N})$$

NB $p(f_1|C) * p(f_2|C) * p(f_3|C) * p(f_4|C) * p(C)$

for each feature count this shit below



In the learning phase, we compute the table of likelihoods (probabilities) from the training data. They are:

Train

$P(Outlook=o|Class_{Play=b})$, where o ∈ [Sunny, Overcast, Rainy] and b ∈ [yes, no]

$P(Temperature=t|Class_{Play=b})$, where t ∈ [Hot, Mild, Cool] and b ∈ [yes, no],

$P(Humidity=h|Class_{Play=b})$, where h∈ [High, Norma] and b ∈ [yes, no],

$P(Wind=w|Class_{Play=b})$, where w ∈ [Weak, Strong] and b ∈ [yes, no].

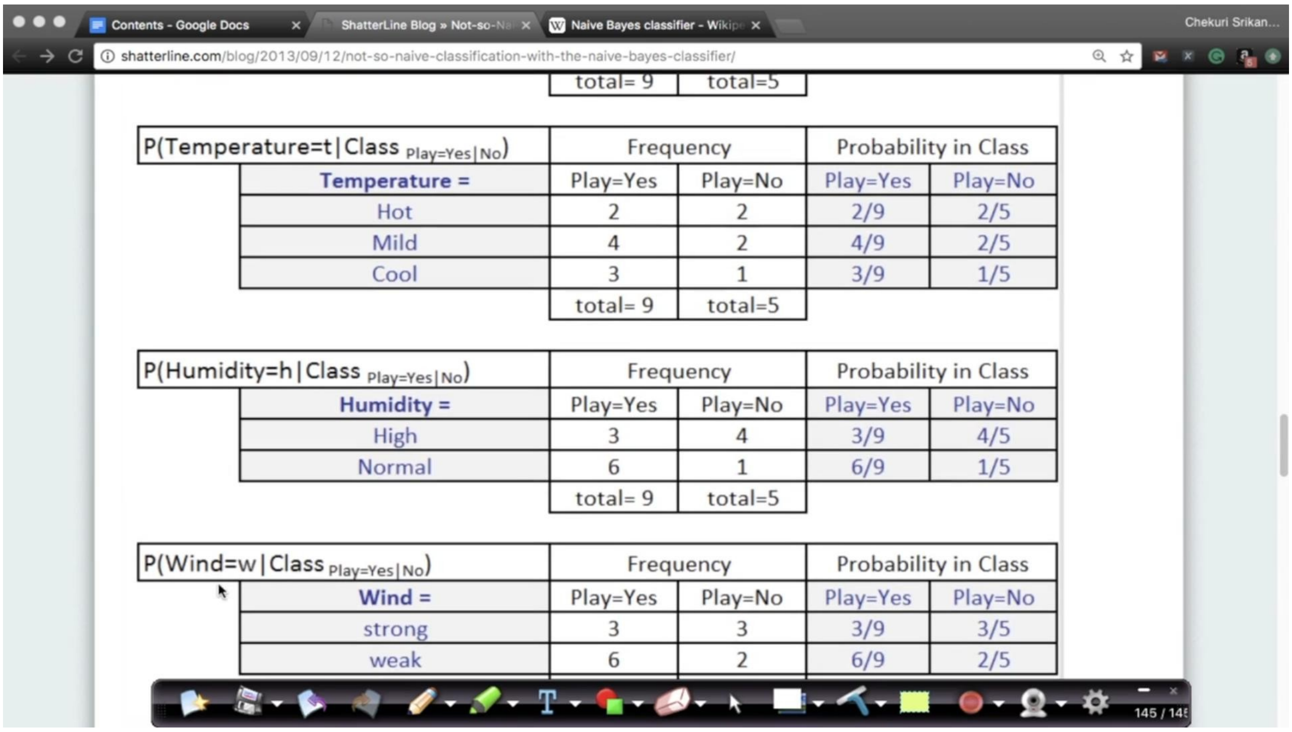
$p(outlook=Sunny \mid C=Yes) = 2/9$

f_1

P(Outlook=o|Class $_{Play=Yes\|No}$)	Frequency		Probability in Class	
Outlook =	Play=Yes	Play=No	Play=Yes	Play=No
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rain	3	2	3/9	2/5
	total= 9	total=5		

P(Temperature=t|Class $_{Play=Yes\|No}$)	Frequency		Probability in Class	
Temperature =	Play=Yes	Play=No	Play=Yes	Play=No

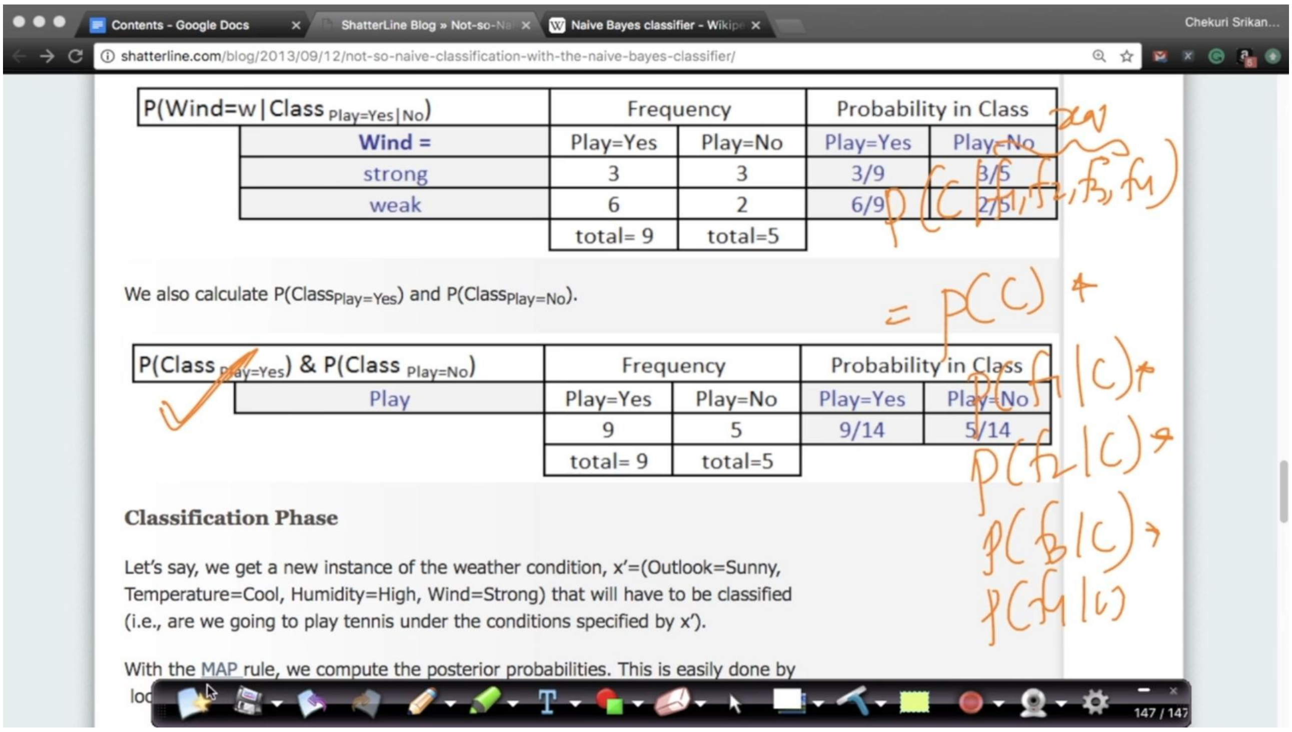
similarly count for all features



P(Temperature=t|Class $_{Play=Yes\|No}$)	Frequency		Probability in Class	
Temperature =	Play=Yes	Play=No	Play=Yes	Play=No
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
	total= 9	total=5		

P(Humidity=h|Class $_{Play=Yes\|No}$)	Frequency		Probability in Class	
Humidity =	Play=Yes	Play=No	Play=Yes	Play=No
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
	total= 9	total=5		

P(Wind=w|Class $_{Play=Yes\|No}$)	Frequency		Probability in Class	
Wind =	Play=Yes	Play=No	Play=Yes	Play=No
strong	3	3	3/9	3/5
weak	6	2	6/9	2/5



P(Wind=w|Class $_{Play=Yes\|No}$)	Frequency		Probability in Class	
Wind =	Play=Yes	Play=No	Play=Yes	Play=No
strong	3	3	3/9	3/5
weak	6	2	6/9	2/5
	total= 9	total=5		

We also calculate P(Class$_{Play=Yes}$) and P(Class$_{Play=No}$).

P(Class $_{Play=Yes}$) & P(Class $_{Play=No}$)	Frequency		Probability in Class	
Play	Play=Yes	Play=No	Play=Yes	Play=No
	9	5	9/14	5/14
	total= 9	total=5		

Classification Phase

Let's say, we get a new instance of the weather condition, x'=(Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong) that will have to be classified (i.e., are we going to play tennis under the conditions specified by x').

With the MAP rule, we compute the posterior probabilities. This is easily done by lo

if we assume number of categories each feature takes is small, o(1)

then time complexity for training is o(ndc)

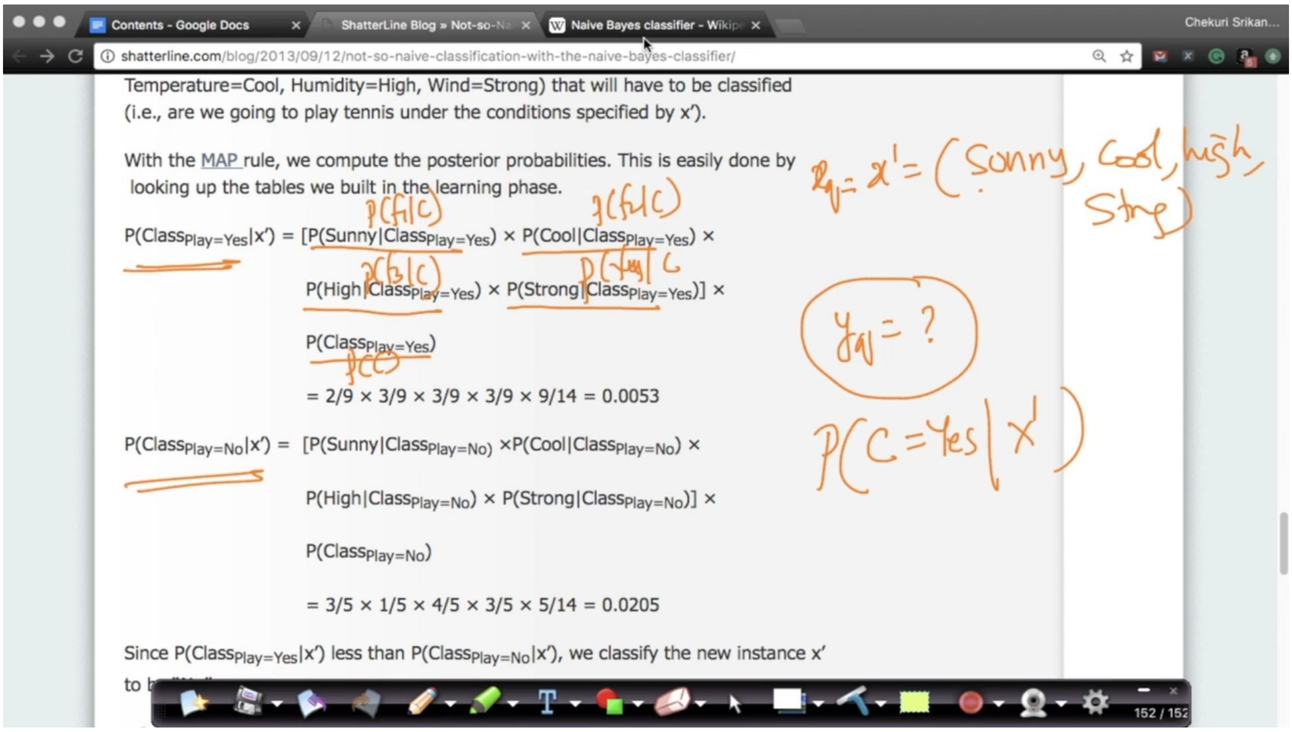
	Outlook	Temperature	Humidity	Wind	Class Play=Yes Play=No
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

The Learning Phase

space complexity :

if there are C classes, order of C space

d*c space because , d features, c classes



Temperature=Cool, Humidity=High, Wind=Strong) that will have to be classified (i.e., are we going to play tennis under the conditions specified by x').

With the MAP rule, we compute the posterior probabilities. This is easily done by looking up the tables we built in the learning phase.

$P(Class_{Play=Yes}|x') = [P(Sunny|Class_{Play=Yes}) \times P(Cool|Class_{Play=Yes}) \times P(High|Class_{Play=Yes}) \times P(Strong|Class_{Play=Yes})] \times P(Class_{Play=Yes})$

$= 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$

$P(Class_{Play=No}|x') = [P(Sunny|Class_{Play=No}) \times P(Cool|Class_{Play=No}) \times P(High|Class_{Play=No}) \times P(Strong|Class_{Play=No})] \times P(Class_{Play=No})$

$= 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0205$

Since $P(Class_{Play=Yes}|x')$ less than $P(Class_{Play=No}|x')$, we classify the new instance x'

we’ll store all data required for this in a dictionary test complexity :

for c classes, we need to lookup each feature, time = d*c

if d and c is small, training time is v v small

space is also small as compared to k-nn