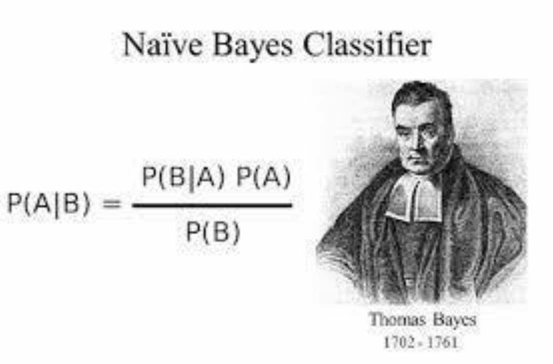
Naïve Bayes Algorithm

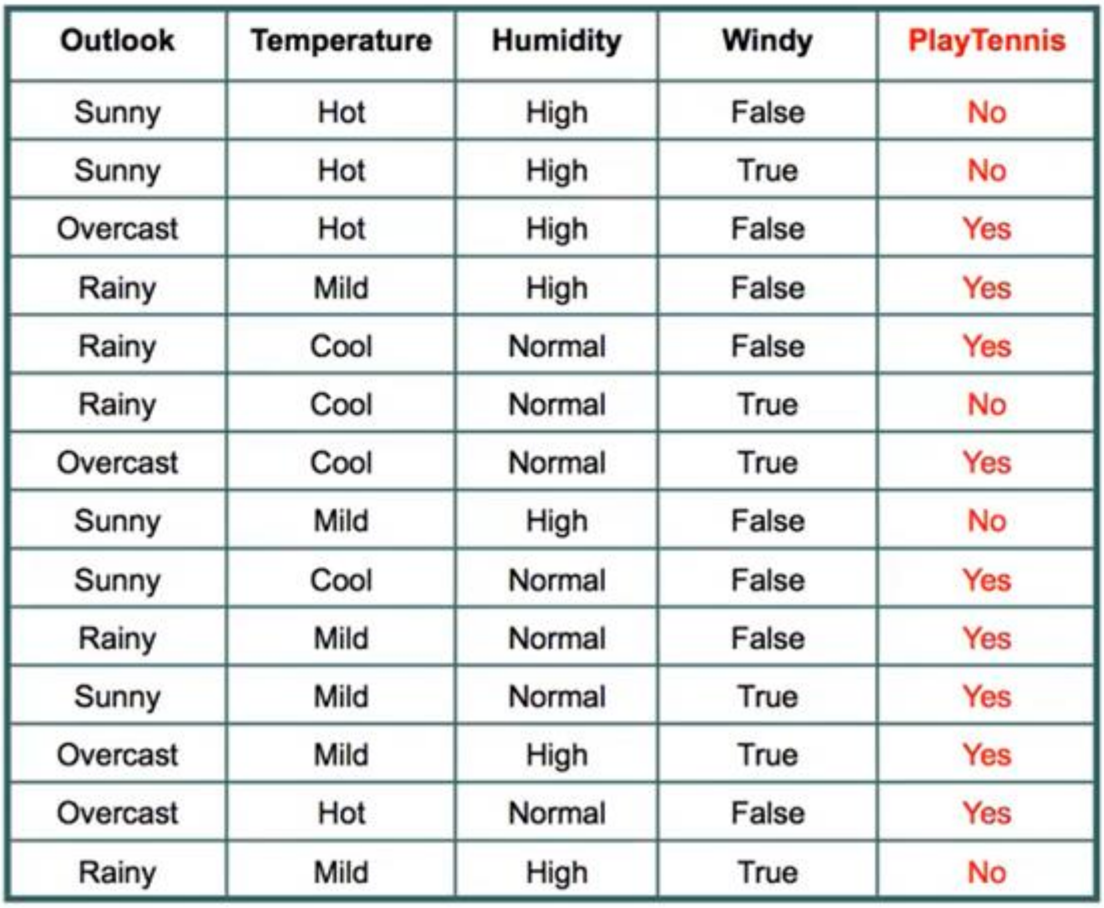
The naïve Bayes algorithm is a family of probabilistic classification algorithms used for tasks like text classification, such as spam filtering and sentiment analysis.

It assumes that features are independent of each other, meaning the presence or absence of one feature doesn’t impact the probability of another feature.

This algorithm is based on bayes theorem in Probability.



Intuition of Naïve bayes algorithm:



Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

(see its solution in - https://youtu.be/Vlj6xS937E4?si=XnaPFtRaNuLKfW2C)

What happens inside naïve bayes during training phase?

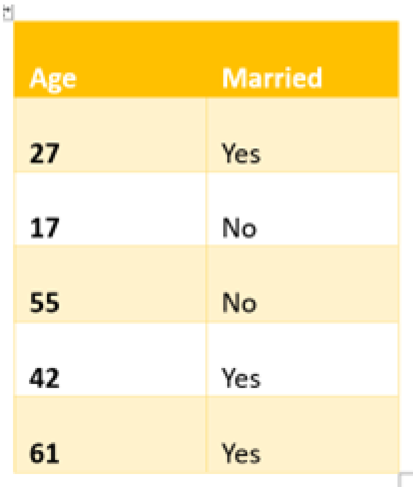
During the training phase of the Naïve Bayes algorithm, probabilities for all possible combinations of feature values and classes are calculated and stored in a hashing format.

In the testing phase, the algorithm retrieves the corresponding probabilities based on the observed feature values, multiplies them together, and provides the final output, indicating the predicted class.

By precomputing and storing the probabilities during training, the testing phase becomes more efficient.

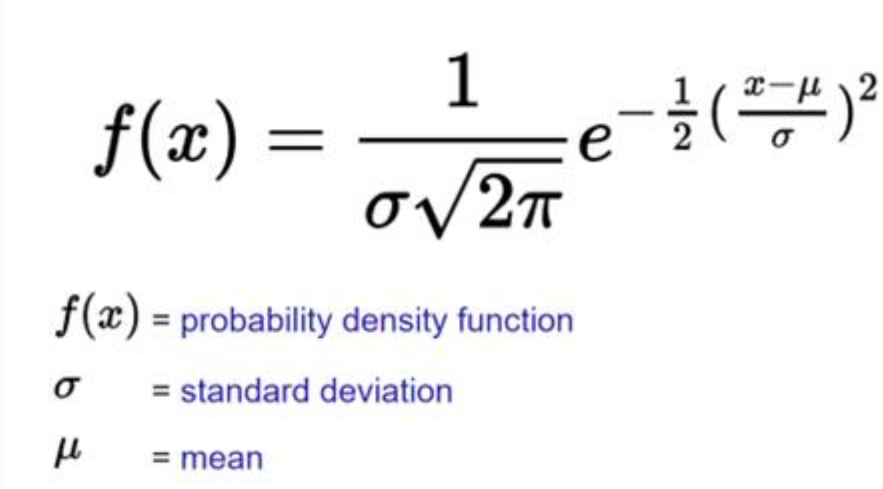
How naïve bayes handles numerical data ?

Suppose I have a dataset in which age and whether the person is married or not given.



Age	Married
27	Yes
17	No
55	No
42	Yes
61	Yes

I have a age 46 and I want to predict whether the person is married or not using naïve bayes. So I need to calculate P(Y | 46) and P(N | 46). But in given table may be the given age not present then that time these probabilities becomes zero. To resolve this issue we assume that our Age columns follows a gaussian distribution and based on this distribution we calculate μ and σ and x= 46 and we put all the values in gaussian distribution function:



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$f(x)$ = probability density function

σ = standard deviation

μ = mean

Now we calculate f(x) which gives the probability of corresponding age. So in this way using Probability density function we can use naïve bayes algorithm for numerical data.

What if data distribution is not Gaussian Distribution?

1. **Data Transformation:** Depending on the nature of your data, you could apply a transformation to make it more normally distributed. Common transformations include the logarithm, square root, and reciprocal transformations.

2. **Alternative Distributions:** If you know or suspect that your data follow a specific non-normal distribution (e.g., exponential, Poisson, etc.), you can modify the Naïve Bayes algorithm to assume that specific distribution when calculating the likelihoods.

3. **Discretization:** You can turn your continuous data into categorical data by binning the values. There are various ways to decide on the bins, including equal width bins, equal frequency bins, or using a more sophisticated method like k-means clustering. Once your
data is binned, you can use the standard Multinomial or Bernoulli Naïve Bayes methods.

4. **Kernel Density Estimation:** A non-parametric way to estimate the probability density function of a random variable. Kernel density estimation can be used when the distribution is unknown.

5. **Use other models:** If none of the above options work well, it may be best to consider a different classification algorithm that doesn’t make strong assumptions about the distributions of the features, such as Decision Trees, Random Forests, or Support Vector Machines

Zero Probability Problem in naïve bayes: (Laplace Additive Smoothing)

Laplace additive smoothing is a technique commonly used in Naïve Bayes algorithms to handle the issue of zero probabilities.

It is applied to avoid the problem of encountering unseen combinations of features and classes during classification.

In Naïve Bayes, when calculating probabilities, there is a possibility of encountering feature values in the test data that were not observed in the training data.

As a result, the conditional probability for such combinations would be zero, which can lead to inaccurate predictions.

Laplace additive smoothing addresses this problem by adding a small constant (usually 1) to both the numerator and the denominator when estimating probabilities. This way, even if a specific feature value has not been observed for a particular class in the training data, it will still have a non-zero probability.

P(feature value | class) = (count of feature value given the class + α) / (count of class + n * α)

Here α is generally taken 1 and value of n depends on which type of naïve bayes you used.

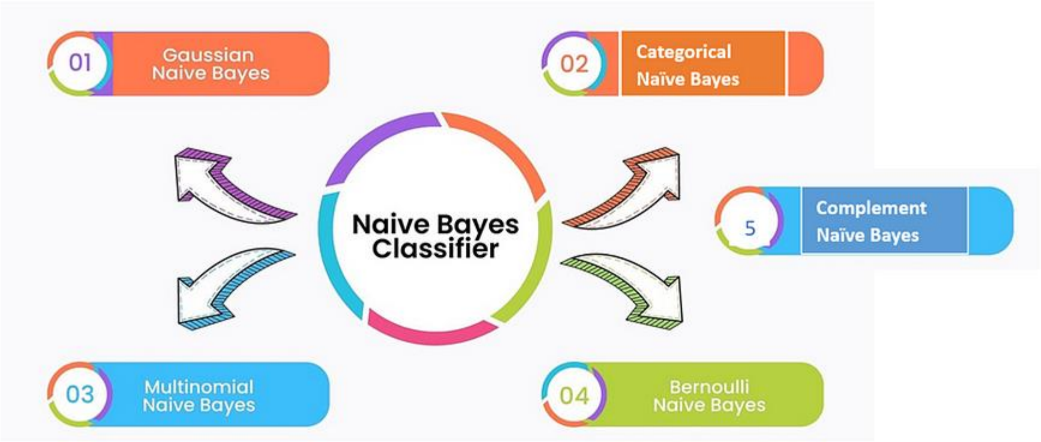
(study detailed info about this later)

Bias-Variance Trade-Off in naïve bayes classifier using Laplacian additive smoothing coefficient ‘α’:

An extremely large alpha value in Laplace smoothing results in over smoothing, where all feature occurrences are assigned similar probabilities. This leads to an underfitting scenario, where the model becomes less sensitive to the observed data and performs poorly in generalizing to new, unseen instances.

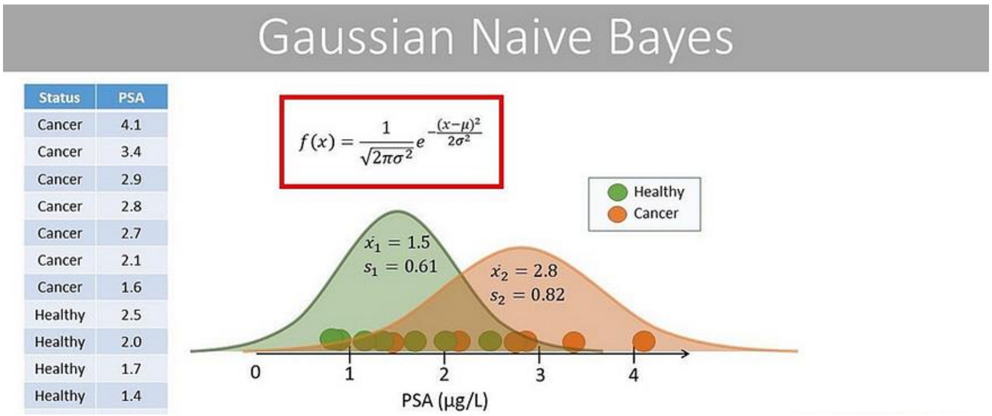
(related reading - https://medium.com/@sachinsoni600517/na%C3%AFve-bayes-machine-learning-algorithm-from-basic-to-advanced-91a8fb749ee3)

Types of naïve Bayes:



Gaussian Naïve Bayes:

When all the input features are numerical, then that time use Gaussian Naïve Bayes. Laplace Additive smoothing is not applicable on Gaussian Naïve Bayes because here probability is never zero for any input feature.



Gaussian Naive Bayes

Status	PSA
Cancer	4.1
Cancer	3.4
Cancer	2.9
Cancer	2.8
Cancer	2.7
Cancer	2.1
Cancer	1.6
Healthy	2.5
Healthy	2.0
Healthy	1.7
Healthy	1.4

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Categorical Naïve Bayes:

When all the input features are categorical, then that time use Categorical Naïve Bayes. When applying Laplace additive smoothing in this case **’n’** indicates number of unique feature values or categories for a specific input column.

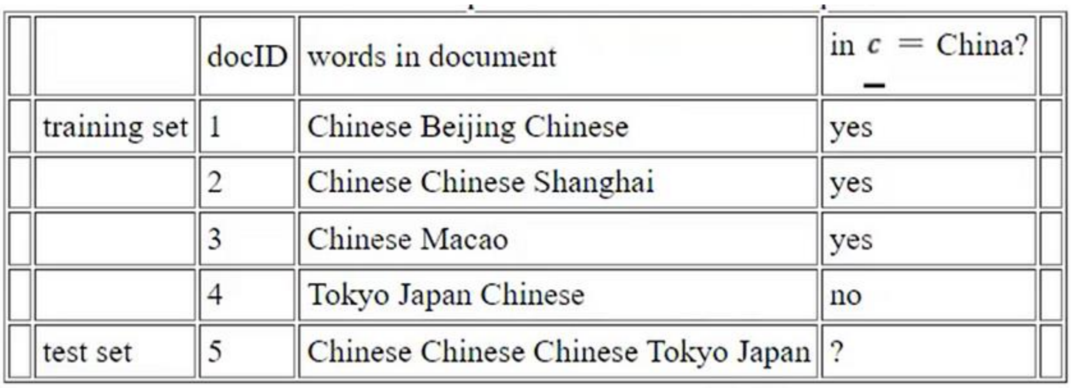
Multinomial Naïve Bayes:

Multinomial Naïve Bayes is a variant of the Naïve Bayes algorithm that is specifically designed for text classification problems where the features represent the frequency or occurrence of words in a document.

It is commonly used in natural language processing tasks, such as sentiment analysis, spam filtering, topic classification, and document categorization.

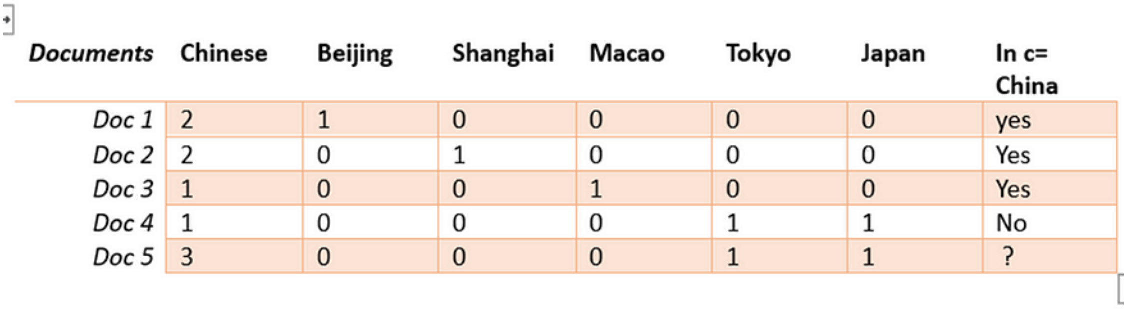
In Multinomial Naïve Bayes, the features are typically represented as term frequencies, such as the number of times a word appears in a document, or as TF-IDF (Term Frequency-Inverse Document Frequency) values, which take into account both the frequency of a word in a document and its rarity across the entire dataset.

Let’s take an example to understand Multinomial Naïve bayes.



	docID	words in document	in c = China?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

The above table represents docID with different words are coming in that document. Training set contains four different documents and on the basis of words, we need to predict whether docID 5 is from China or not. Now applying count based bag-of-words I am converting each document into word frequency table, where the columns represent different words and the rows represent each document.



Documents	Chinese	Beijing	Shanghai	Macao	Tokyo	Japan	In c= China
Doc 1	2	1	0	0	0	0	yes
Doc 2	2	0	1	0	0	0	Yes
Doc 3	1	0	0	1	0	0	Yes
Doc 4	1	0	0	0	1	1	No
Doc 5	3	0	0	0	1	1	?

Now for Document 5, we need to predict whether it is c= China or not.

So, for predicting above we need to calculate,
P(Yes | Chinese=3, Beijing= 0, Shanghai=0, Macao= 0, Tokyo=1, Japan=1) and P(No | Chinese=3, Beijing= 0,Shanghai=0,Macao= 0, Tokyo=1, Japan=1)

Let say, B=(Chinese=3, Beijing= 0, Shanghai=0, Macao= 0, Tokyo=1,Japan=1)

P(Yes | B) = P(Yes) * P(B | Yes)
P(Yes)= 3/4,

P(Chinese | Yes) = 5 + 1/8+6 = 6/14, Here adding 1 in numerator and 6 in denominator indicates Laplace additive smoothing and alpha=1 and n=6 because n indicates total number of different words.

P(Beijing | Yes)= 1+1/8+6 = 2/14
P(Shanghai | Yes) = 1+1/8+6 = 2/14
P(Macao | Yes) = 1+1/8+6 = 2/14

P(Tokyo | Yes) = 0+1/8+6= 1/14
P(Japan | Yes) = 0+1/8+6= 1/14

Similarly all these Probabilities is calculated for No option also,

P(No) = 1/4
P(Beijing | No)= 0+1/3+6 = 1/9
P(Shanghai | No) = 1+1/3+6 = 1/9
P(Macao | No) = 0+1/3+6 = 1/9
P(Tokyo | No) = 1+1/3+6= 2/9
P(Japan | No) = 1+1/3+6= 2/9
P(Chinese | No) = 1+1/3+6 = 2/9

P(Yes | B) = 3/4 * $(6/14)^3$ *1/14 * 1/14 = 0.0003
P(No | B) = 1/4 * $(2/9)^3$ * 2/9 * 2/9 = 0.0001

So for document 5 is from China because P(Yes | B) > P(No | B).

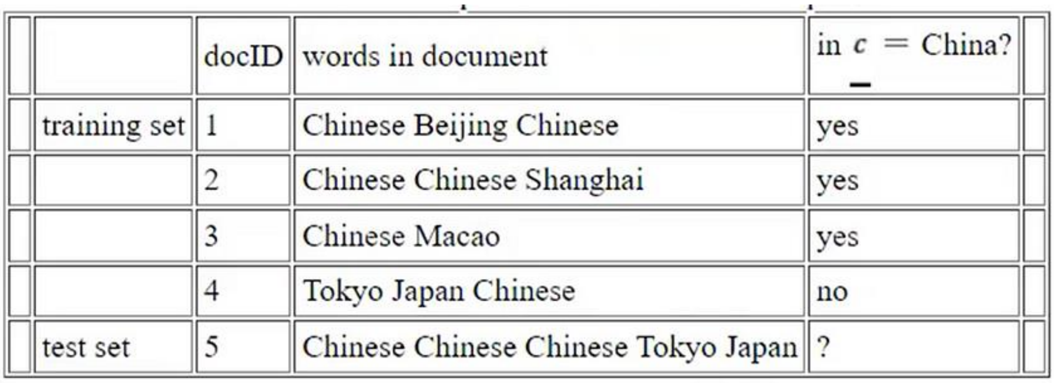
Bernoulli Naïve Bayes:

Bernoulli Naïve Bayes is commonly used in scenarios where the features are binary or Boolean variables.

In Bernoulli Naïve Bayes, the input data is represented as a binary feature vector, where each feature represents the presence or absence of a particular attribute.

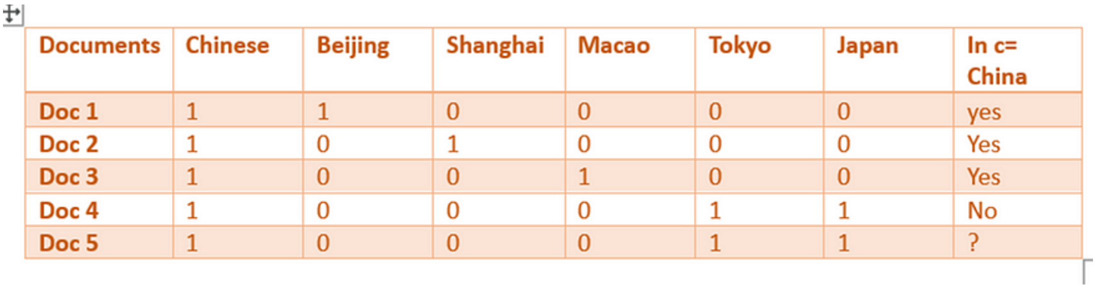
For example, in text classification, each feature could correspond to the presence or absence of a specific word in a document.

Let’s take an example to understand Bernoulli Naïve bayes.



		docID	words in document	in c = China?	
	training set	1	Chinese Beijing Chinese	yes	
		2	Chinese Chinese Shanghai	yes	
		3	Chinese Macao	yes	
		4	Tokyo Japan Chinese	no	
	test set	5	Chinese Chinese Chinese Tokyo Japan	?	

The above table represents docID with different words are coming in that document. Training set contains four different documents and on the basis of words, we need to predict whether docID 5 is from China or not. Now applying binary bag-of-words I am converting each document into binary word frequency table, where the columns represent different words and the rows represent each document.



Documents	Chinese	Beijing	Shanghai	Macao	Tokyo	Japan	In c= China
Doc 1	1	1	0	0	0	0	yes
Doc 2	1	0	1	0	0	0	Yes
Doc 3	1	0	0	1	0	0	Yes
Doc 4	1	0	0	0	1	1	No
Doc 5	1	0	0	0	1	1	?

Now for Document 5, we need to predict whether it is c= China or not.

So, for predicting above we need to calculate,
P(Yes | Chinese=1, Beijing= 0, Shanghai=0, Macao= 0, Tokyo=1, Japan=1) and P(No | Chinese=1, Beijing= 0,Shanghai=0,Macao= 0, Tokyo=1, Japan=1)

Let say, B=(Chinese=1, Beijing= 0, Shanghai=0, Macao= 0, Tokyo=1,Japan=1)

P(Yes | B) = P(Yes) * P(B | Yes)
P(Yes)= 3/4,

Probability of Bernoulli random variable is given by,
P(X=K) = PK + (1-P)(1-K)

P(Chinese=1 | Yes) = 3 +1/3+2 =4/5 , because here K=1
and P(X=1) = P*1 +0 = P ,and P indicates probability of 1 and also I apply Laplace Additive Smoothing on probabilities.

P(Beijing =0 | Yes)= 2+1/3+2 = 3/5
P(Shanghai=0 | Yes) = 2+1/3+2 = 3/5
P(Macao=0 | Yes) = 2+1/3+2 = 3/5
P(Tokyo=1 | Yes) = 0+1/3+2 = 1/5
P(Japan=1 | Yes) = 0+1/3+2 = 1/5

Similarly all these Probabilities is calculated for No option also,

P(No) = 1/4
P(Beijing =0| No)= 1+1/1+2 = 2/3
P(Shanghai=0 | No) = 1+1/1+2=2/3
P(Macao=0 | No) = 1+1/1+2=2/3
P(Tokyo=1 | No) = 1+1/1+2=2/3
P(Japan=1 | No) = 1+1/1+2=2/3
P(Chinese=1 | No) = 1+1/1+2=2/3

P(Yes | B) =3/4 *4/5 * 3/5 * 3/5 * 3/5 *1/5 *1/5 = 0.005
P(No | B) = 1/4 * 2/3 * 2/3 * 2/3 * 2/3 * 2/3 * 2/3= 0.022

So for document 5 is not from China because P(No| B) > P(Yes | B).