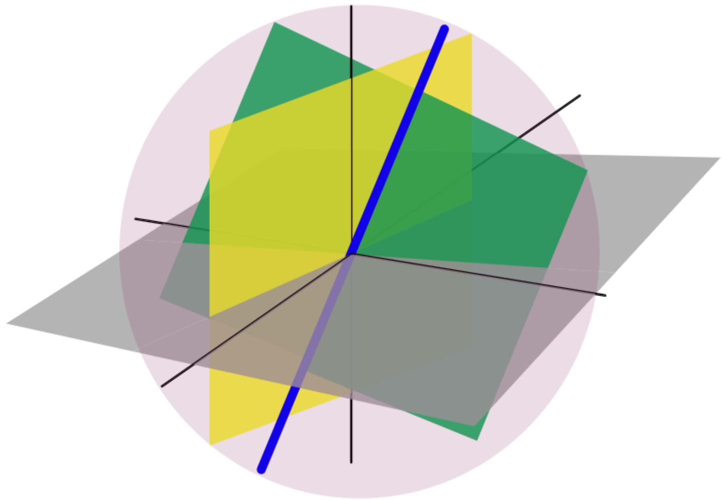
Information
Technology
Institute



# DATA MINING

Machine Learning

Prepared by Khaled Shaker

# Clustering:

Clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters.

Cluster analysis or simply clustering is the process of partitioning a set of data objects (or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters.

Different clustering methods may generate different clustering on the same data set.

Clustering is useful in that it can lead to the discovery of previously unknown groups within the data.

# Clustering:

Cluster analysis can be used as a standalone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, and to focus on a particular set of clusters for further analysis.

It may serve as a preprocessing step for other algorithms, such as characterization, attribute subset selection, and classification, which would then operate on the detected clusters and the selected attributes or features.

# Clustering:

## *k-means cluster*

k-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem.

The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters)

The main idea is to define k centers, one for each cluster.

These centers should be placed in a cunning way because of different location causes different result.

So, the better choice is to place them as much as possible far away from each other.

The next step is to take each point belonging to a given data set and associate it to the nearest center.

# Clustering:

***k-means cluster***

When no point is pending, the first step is completed and an early group age is done.

At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step.

After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center.

A loop has been generated.

As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more.

# Clustering:

***k-means cluster***

Example:

We have the following 5 points and we want to group them in 2 clusters:

| i | X | Y |
|---|---|---|
| A | 1 | 1 |
| B | 1 | 0 |
| C | 0 | 2 |
| D | 2 | 4 |
| E | 3 | 5 |

# Clustering:

### *k-means cluster*

Solution:

Choose 2 points to be the center of each cluster (selected Randomly) “A, C”

Step1: Calculate the distance between each point and the 2 selected point

$$length = \sqrt{(X1\ - X2)^2 + (Y1\ - Y2)^2}$$

| i | A (Cluster 1) | C (Cluster 2) |
|---|---|---|
| A | 0 | 1.4 |
| B | 1 | 2.2 |
| C | 1.4 | 0 |
| D | 3.2 | 2.8 |
| E | 4.5 | 4.2 |

# Clustering:

***k-means cluster***

Compare the distance between each point and the 2 selected groups. This point will belong to the cluster which has the smallest distance to it

Point B, belong to the Cluster of Point "A" (1 less than 2.2)

Point D, belong to the Cluster of Point "C" (2.8 less than 3.2)

Point E, belong to the Cluster of Point "C" (4.2 less than 4.5)

| i | X | Y | Cluster |
|---|---|---|---|
| A | 1 | 1 | 1 |
| B | 1 | 0 | 1 |
| C | 0 | 2 | 2 |
| D | 2 | 4 | 2 |
| E | 3 | 5 | 2 |

# Clustering:

***k-means cluster***

Calculate the mean of Cluster 1:

$X = (1 + 1) / 2 = 1$

$Y = (1 + 0) / 2 = 0.5$

Mean Cluster1 (1, 0.5)

Calculate the mean of Cluster 2:

$X = (0 + 2 + 3) / 3 = 1.7$

$Y = (2 + 4 + 5) / 3 = 3.7$

Mean Cluster2 (1.7, 3.7)

# Clustering:

## *k-means cluster*

Step2: Recalculate the distance from each point to the cluster means

| I | Cluster 1 | Cluster 2 |
|---|---|---|
| A | 0.5 | 2.7 |
| B | 0.5 | 3.7 |
| C | 1.8 | 2.4 |
| D | 3.6 | 0.5 |
| E | 4.9 | 1.9 |

Compare the distance between each point and the 2 cluster mean. This point will belong to the cluster which has the smallest distance to it

Point A, belong to the Cluster 1 (0.5 less than 2.7)

Point B, belong to the Cluster 1 (0.5 less than 3.7)

# Clustering:

***k-means cluster***

Point C, belong to the Cluster 1 (1.8 less than 2.4)

Point D, belong to the Cluster 2 (0.5 less than 3.6)

Point E, belong to the Cluster 2 (1.9 less than 4.9)

| i | X | Y | Cluster |
|---|---|---|---|
| A | 1 | 1 | 1 |
| B | 1 | 0 | 1 |
| C | 0 | 2 | 1 |
| D | 2 | 4 | 2 |
| E | 3 | 5 | 2 |

# Clustering:

***k-means cluster***

Calculate the mean of Cluster 1: (0.7, 1)

Calculate the mean of Cluster 2: (2.5, 4.5)

Step3: Recalculate the distance from each point to the cluster means. In this example we will find no change, so it is the final solution

# Clustering:

***k-means cluster***

Calculate the mean of Cluster 1: (0.7, 1)

Calculate the mean of Cluster 2: (2.5, 4.5)

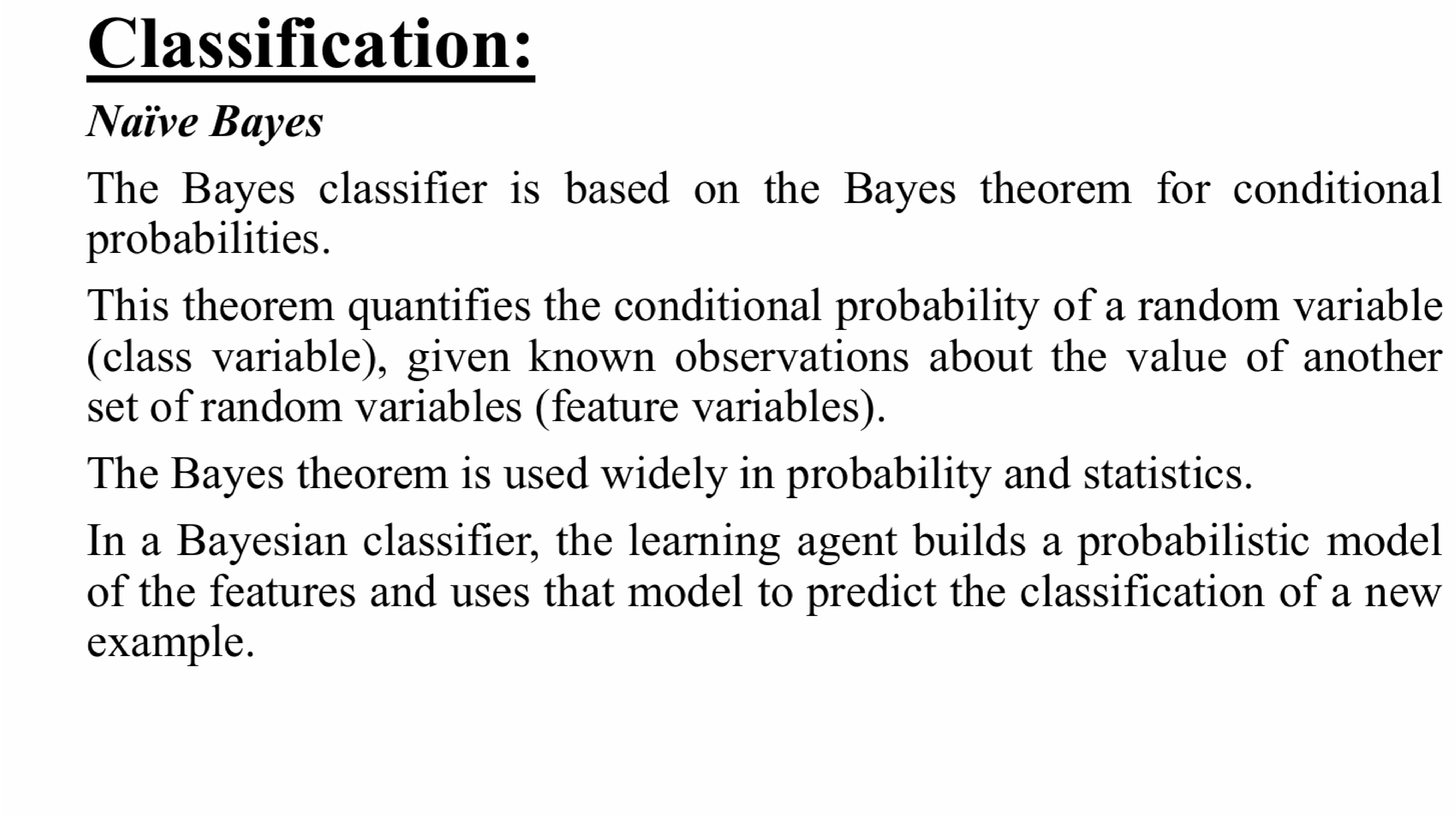
Step3: Recalculate the distance from each point to the cluster means. In this example we will find no change, so it is the final solution

From the definition of conditional probability,

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

Now, considering the second and last terms in the preceding expression, we can write

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \quad \text{for} \quad P(B) > 0$$



# Classification:

***Naïve Bayes***

The Bayes classifier is based on the Bayes theorem for conditional probabilities.

This theorem quantifies the conditional probability of a random variable (class variable), given known observations about the value of another set of random variables (feature variables).

The Bayes theorem is used widely in probability and statistics.

In a Bayesian classifier, the learning agent builds a probabilistic model of the features and uses that model to predict the classification of a new example.

# Classification:

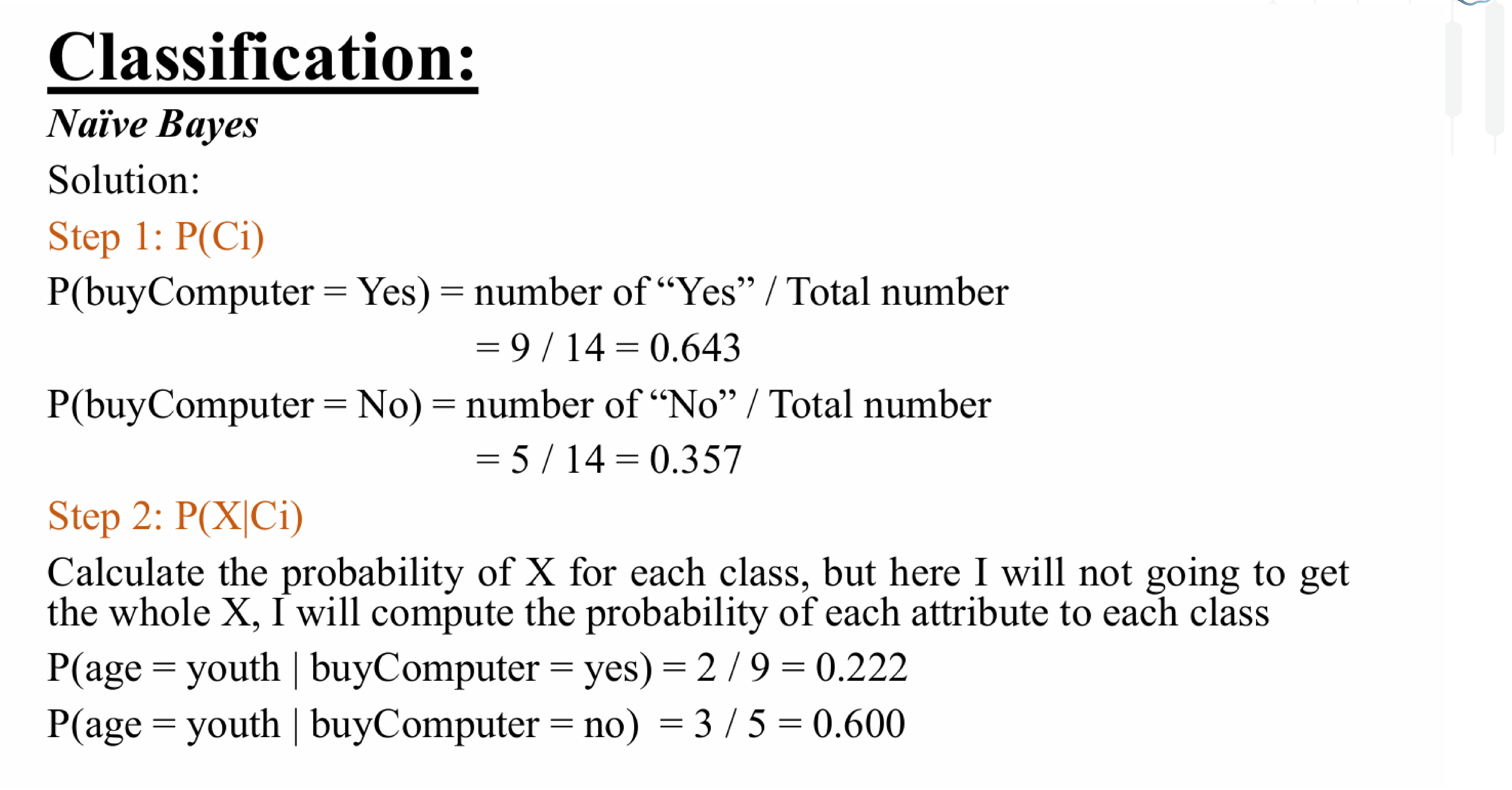
***Naïve Bayes***

Example:

| RID | age | Income | student | Credit-rating | Class: buyComputer |
|---|---|---|---|---|---|
| 1 | Youth | High | No | fair | No |
| 2 | Youth | High | No | Excellent | No |
| 3 | Middle | High | No | Fair | Yes |
| 4 | Senior | Medium | No | Fair | Yes |
| 5 | Senior | Low | Yes | Fair | Yes |
| 6 | Senior | Low | Yes | Excellent | No |
| 7 | Middle | Low | Yes | Excellent | Yes |
| 8 | Youth | Medium | No | Fair | No |
| 9 | Youth | Low | Yes | Fair | Yes |
| 10 | Senior | Medium | Yes | Fair | Yes |
| 11 | Youth | Medium | Yes | Excellent | Yes |
| 12 | Middle | Medium | No | Excellent | Yes |
| 13 | Middle | High | Yes | Fair | Yes |
| 14 | senior | Medium | no | Excellent | No |

Tuple to Classify is:

X(age = youth, income = medium, student = yes, credit = fair), Maximize P(X|Ci) P(Ci)



# Classification:

*Naïve Bayes*

Solution:

Step 1: P(Ci)

P(buyComputer = Yes) = number of "Yes" / Total number

= 9 / 14 = 0.643

P(buyComputer = No) = number of "No" / Total number

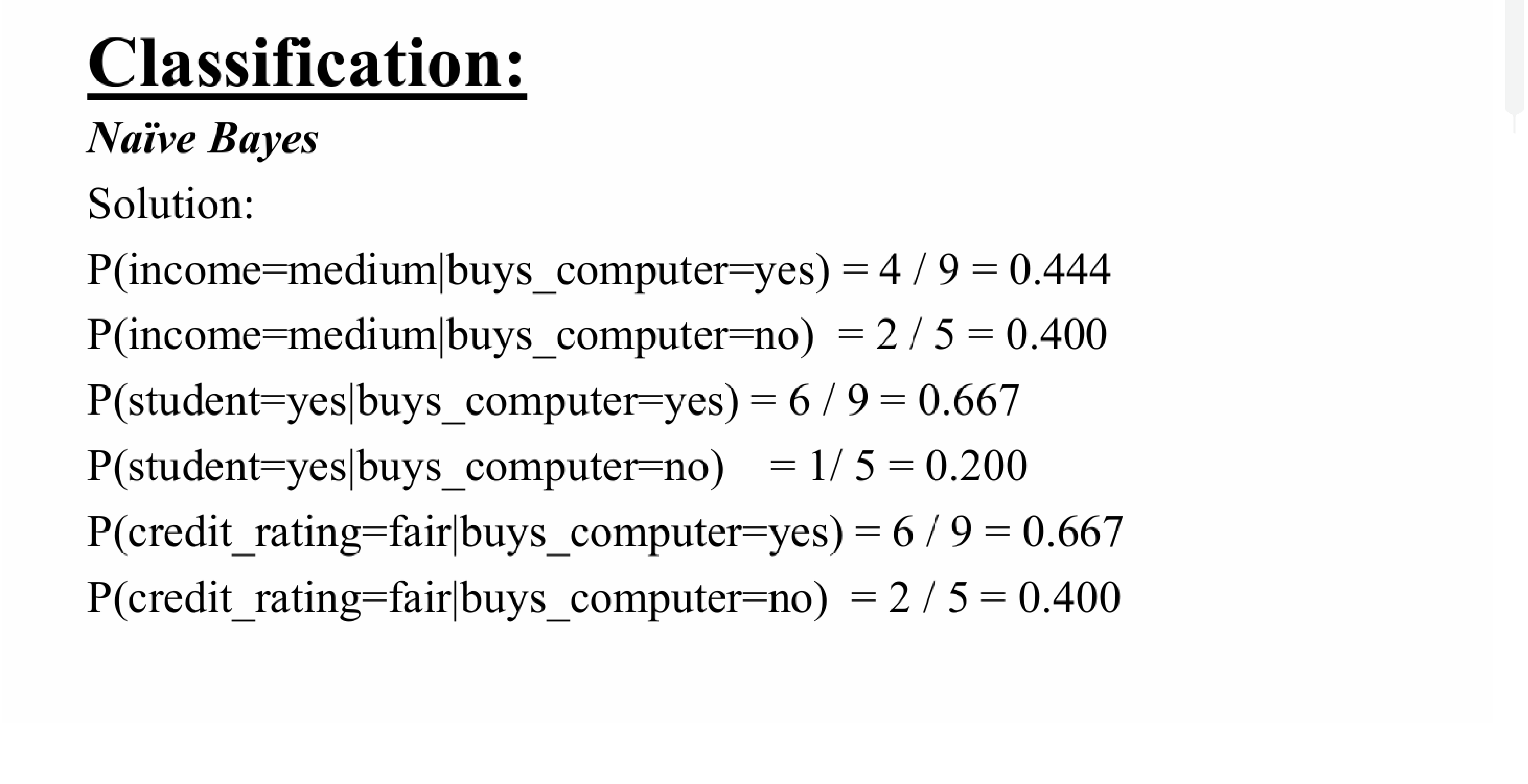
= 5 / 14 = 0.357

Step 2: P(X|Ci)

Calculate the probability of X for each class, but here I will not going to get the whole X, I will compute the probability of each attribute to each class

P(age = youth | buyComputer = yes) = 2 / 9 = 0.222

P(age = youth | buyComputer = no) = 3 / 5 = 0.600



# Classification:

***Naïve Bayes***

Solution:

P(income=medium|buys_computer=yes) = 4 / 9 = 0.444

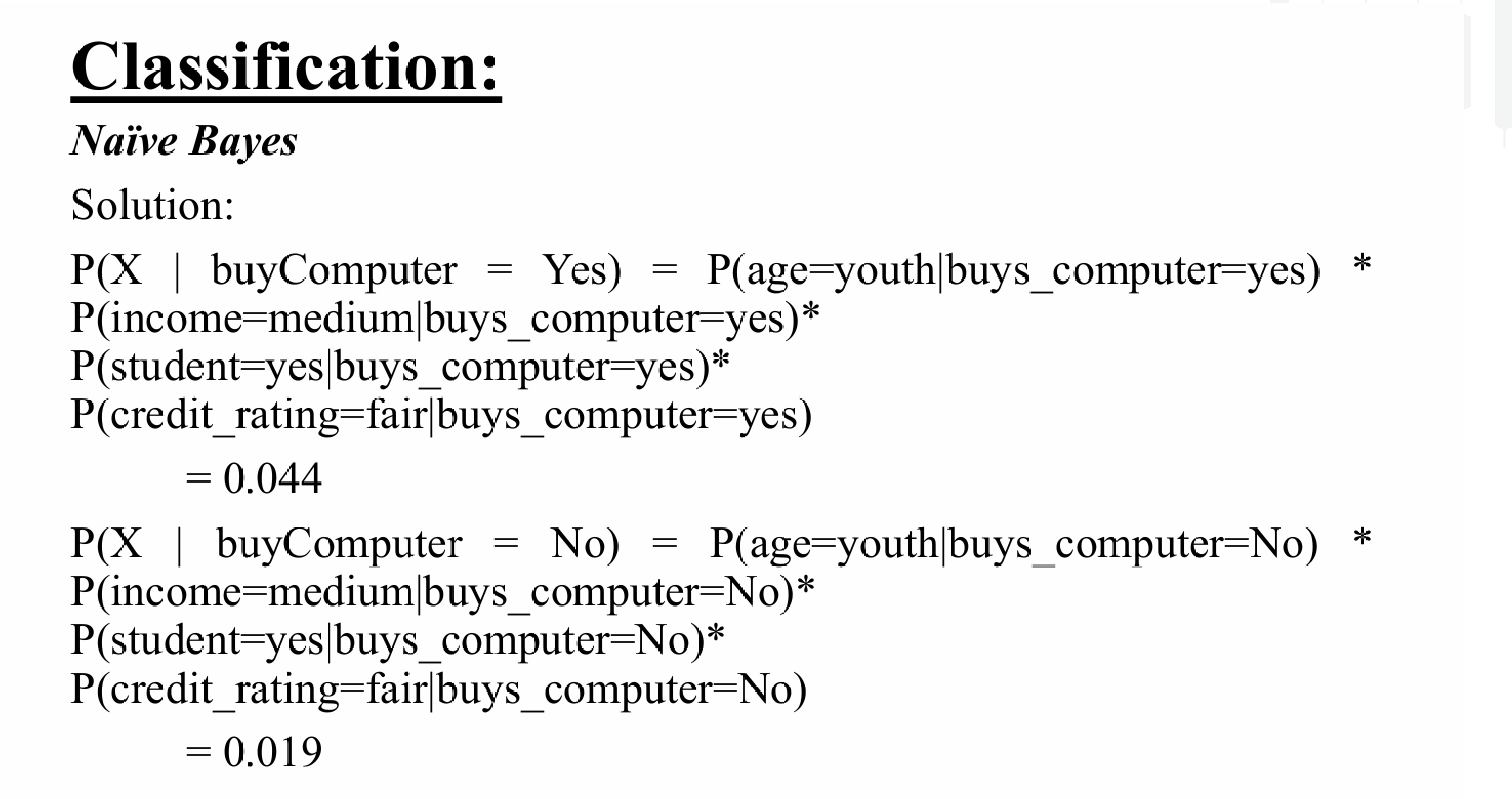
P(income=medium|buys_computer=no) = 2 / 5 = 0.400

P(student=yes|buys_computer=yes) = 6 / 9 = 0.667

P(student=yes|buys_computer=no) = 1/ 5 = 0.200

P(credit_rating=fair|buys_computer=yes) = 6 / 9 = 0.667

P(credit_rating=fair|buys_computer=no) = 2 / 5 = 0.400



# Classification:

***Naïve Bayes***

Solution:

P(X | buyComputer = Yes) = P(age=youth|buys_computer=yes) * P(income=medium|buys_computer=yes)* P(student=yes|buys_computer=yes)* P(credit_rating=fair|buys_computer=yes)

= 0.044

P(X | buyComputer = No) = P(age=youth|buys_computer=No) * P(income=medium|buys_computer=No)* P(student=yes|buys_computer=No)* P(credit_rating=fair|buys_computer=No)

= 0.019

# Classification:

***Naïve Bayes***

Solution:

Step 3: P(X|Ci) P(Ci)

P(X|buys_computer=yes)P(buys_computer=yes) = 0.044 * 0.643

= 0.028

P(X|buys_computer=no)P(buys_computer=no) = 0.019 * 0.357

= 0.007

**The naïve Bayesian Classifier predicts buys_computer=yes for tuple X**

The **Naïve Bayes classifier** is a probabilistic machine learning model based on **Bayes' Theorem**. It is widely used for classification tasks, especially in text classification, spam filtering, and sentiment analysis. The model is called "naïve" because it makes a strong assumption that all features (variables) are **conditionally independent** of each other given the class label. This assumption simplifies computation but may not always reflect real-world scenarios.

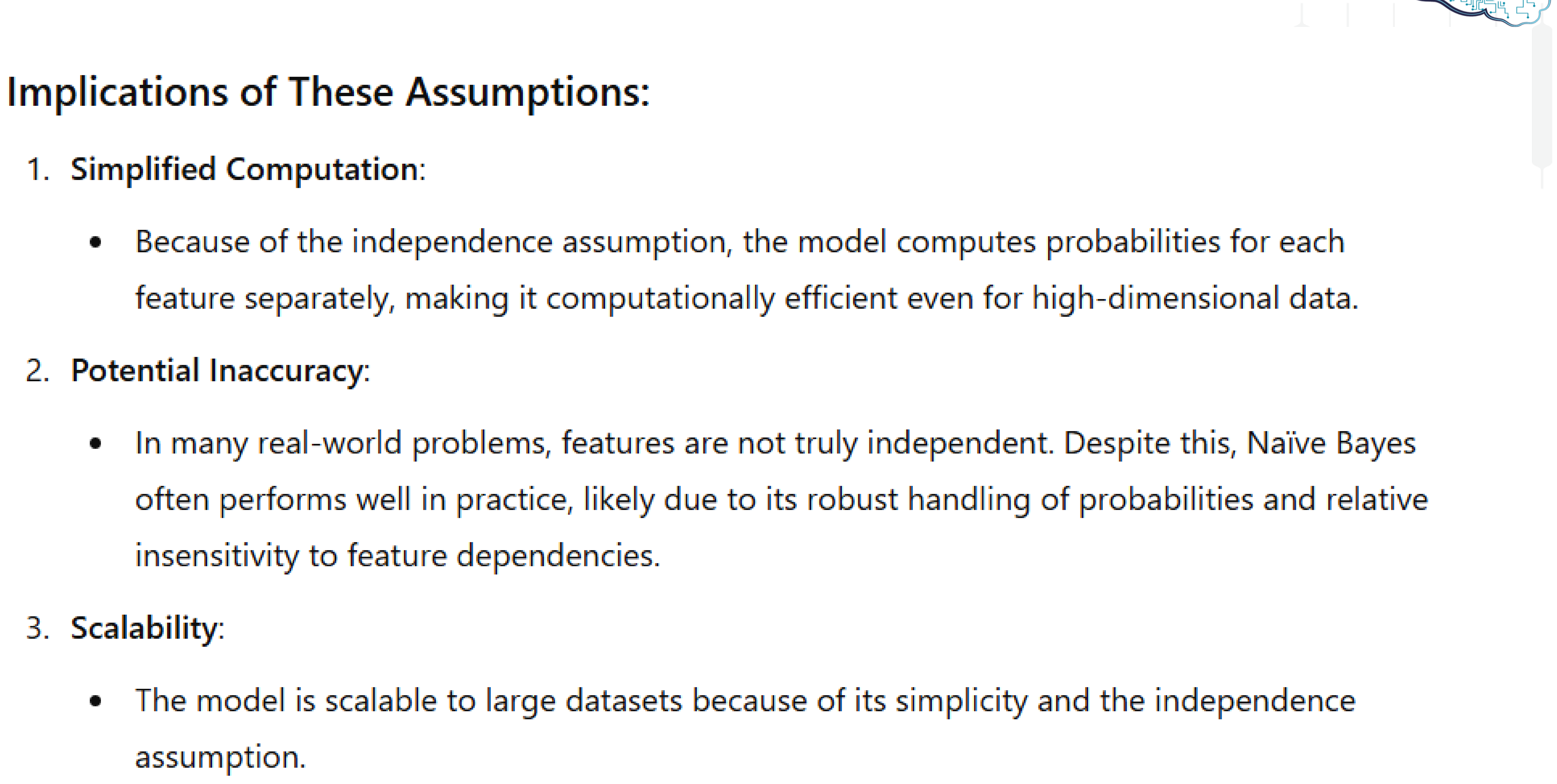
## Key Assumptions of the Naïve Bayes Classifier:

1. **Conditional Independence**:

   - It assumes that the value of a feature is independent of the value of any other feature given the class label.
   - Mathematically, for features $x_1, x_2, \ldots, x_n$ and class $C$:

$$P(x_1, x_2, \ldots, x_n | C) = P(x_1 | C) \cdot P(x_2 | C) \cdot \ldots \cdot P(x_n | C)$$

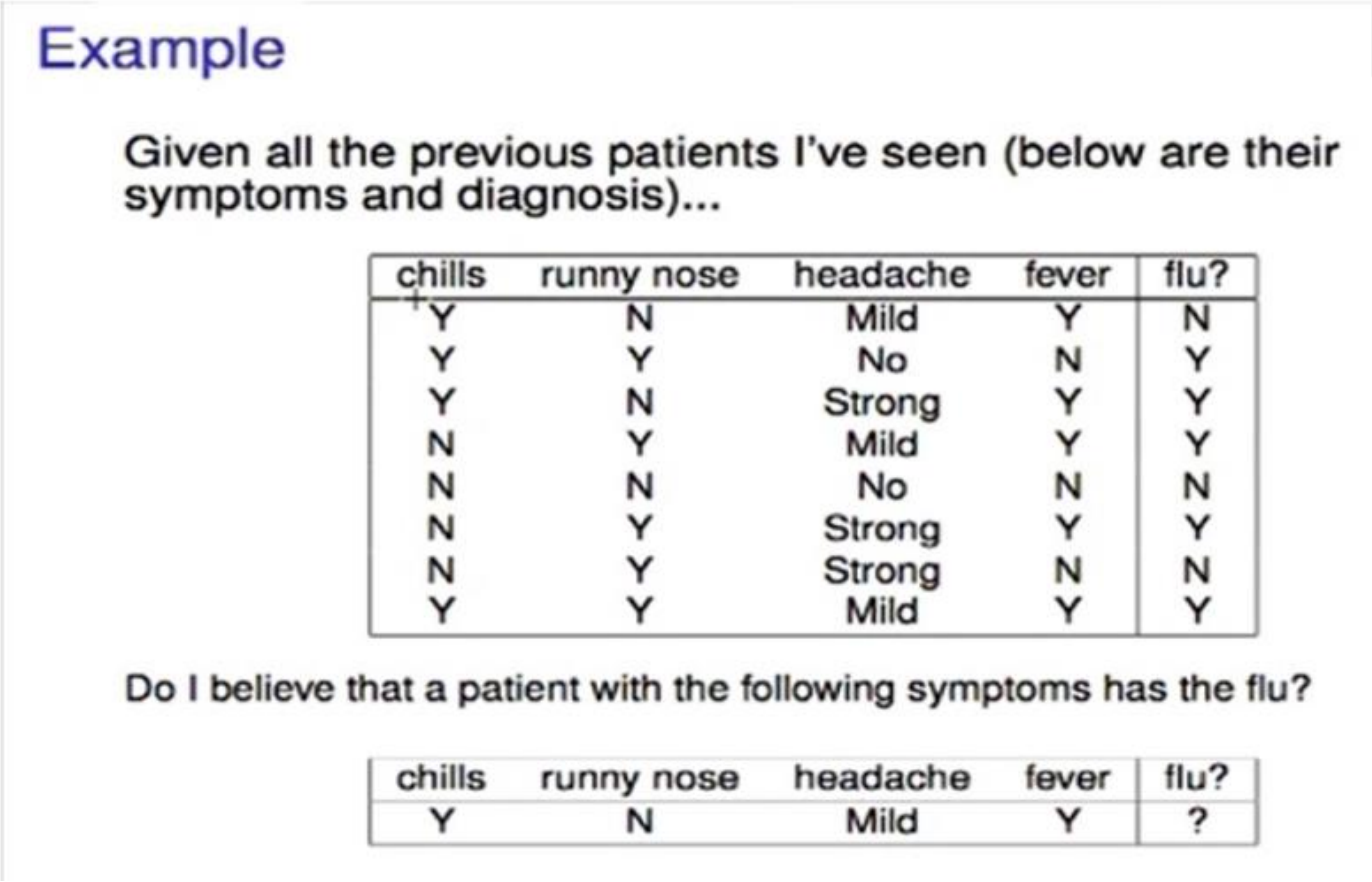
   - This assumption significantly simplifies the computation of joint probabilities.



## Implications of These Assumptions:

1. **Simplified Computation**:
   - Because of the independence assumption, the model computes probabilities for each feature separately, making it computationally efficient even for high-dimensional data.
2. **Potential Inaccuracy**:
   - In many real-world problems, features are not truly independent. Despite this, Naïve Bayes often performs well in practice, likely due to its robust handling of probabilities and relative insensitivity to feature dependencies.
3. **Scalability**:
   - The model is scalable to large datasets because of its simplicity and the independence assumption.

# Assignment



## Example

Given all the previous patients I've seen (below are their symptoms and diagnosis)...

| chills | runny nose | headache | fever | flu? |
|---|---|---|---|---|
| Y | N | Mild | Y | N |
| Y | Y | No | N | Y |
| Y | N | Strong | Y | Y |
| N | Y | Mild | Y | Y |
| N | N | No | N | N |
| N | Y | Strong | Y | Y |
| N | Y | Strong | N | N |
| Y | Y | Mild | Y | Y |

Do I believe that a patient with the following symptoms has the flu?

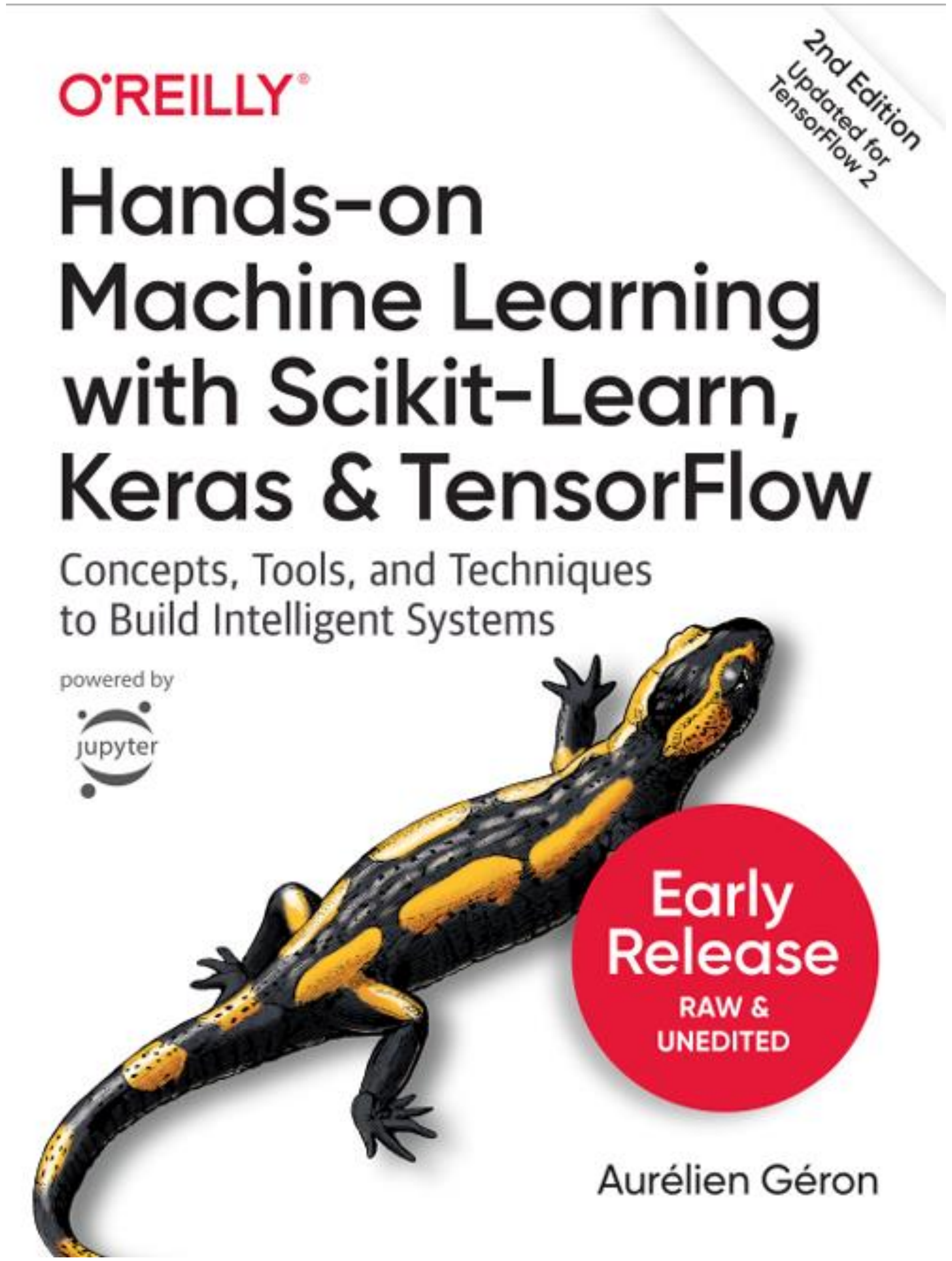
| chills | runny nose | headache | fever | flu? |
|---|---|---|---|---|
| Y | N | Mild | Y | ? |

- **Bernoulli Naive Bayes** : It assumes that all our features are binary such that they take only two values. Means 0s can represent “word does not occur in the document” and 1s as "word occurs in the document" .

- **Multinomial Naive Bayes**: It is used when we have discrete data (e.g. movie ratings ranging from 1 to 5 as each rating will have a certain frequency to represent). In text learning we have the count of each word to predict the class or label.

- **Gaussian Naive Bayes** : Because of the assumption of the normal distribution, Gaussian Naive Bayes is used in cases when all our features are continuous. For example in Iris dataset features are sepal width, petal width, sepal length, petal length. So its features can have different values in data set as width and length can vary. We can’t represent features in terms of their occurrences. This means data is continuous. Hence we use Gaussian Naive Bayes here

# Extra Mile



O'REILLY®

2nd Edition
Updated for TensorFlow 2

Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow

Concepts, Tools, and Techniques to Build Intelligent Systems

powered by jupyter

Early Release
RAW & UNEDITED

Aurélien Géron

شرح الكتاب بالعربي فيديوهات

# Thank you

khaledgama4@gmail.com