# Naïve Bayes

Bayes Theorem & Intuition

# Naïve Bayes

- Bayes Theorem
- Naïve Bayes Classifier

# Bayes Theorem

# Bayes Theorem - Introduction

Let's say we are doing some Analytics for a Factory and there are two machines.
Both the machines produces some wrenches which are same.
The goal of the workers at the end of the day is to figure out the defective Spanners by the end of the day.

**So the question here is:**
**What is the probability of Machine1 or Machine2 producing the defective Spanner/wrench ?**

# Bayes Theorem - Introduction

**Question:**
**What is the probability that the Machine 1 or Machine 2 produces defective Spanner ?**
**i.e.**
If we pick up a defective wrench from the produced lot. What is the probability that it is produced by Machine 1 or Machine 2 ?

# Bayes Theorem - Introduction

**Question:**
**What is the probability that the Machine 1 or Machine 2 produces defective Spanner ?**
**i.e.**
If we pick up a defective wrench from the produced lot. What is the probability that it is produced by Machine 1 or Machine 2 ?

The mathematical concept that we will be using to get this probability is called Bayes Theorem.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

# Bayes Theorem - Introduction

**Problem Statement:**
Machine 1: 30 wrenches/hour
Machine 2: 20 wrenches/hour

Out of all produced parts:
**1% is defective**

Out of all defective parts:
**50% comes from Machine 1**
**50% comes from Machine 2**

**Question is:**
**What is the probability that a part produced by Machine 2 is defective ?**

**Interpretation:**
**Total wrenches produced per hour is 50**
-> P(Mach1) = 30/50 = 0.6
-> P(Mach2) = 20/50 = 0.4
So if we pick a wrench from the lot of 50, the probability of it being produced by Machine 1 is 0.6 or 60%

-> P(defect) = 1%

-> P(Mach1 | Defect) = 50%
-> P(Mach2 | Defect) = 50%

We want is:
**P(Defect | Mach 2) = ?**

# Bayes Theorem - Introduction

**Question is:** Say if machine 2 produces 1000 wrenches, then what portion (percentage) of that quantity produced by Machine 2 is defective (e.g. 5% wrenches are defective, 7% wrenches are defective etc.)

**Without Bayes Theorem:**
This all is very intuitive. We need not any theorem for this, we can do this by sheer calculation. Let's do this without Bayes theorem first.

<u>Example Solution:</u>

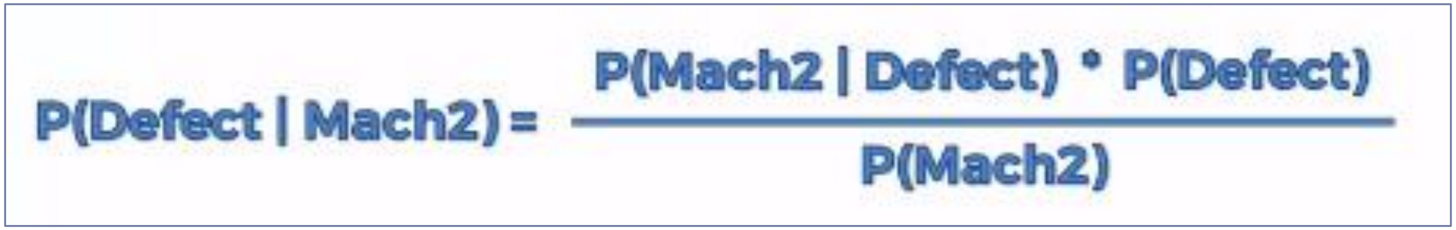
- Total - 1000 wrenches
- 400 came from Machine 2
- 1% have a defect = 10
- 50% of them came from Machine 2 = 5 wrenches

**Question is: What is % defective parts from Machine 2**
**Hence**
**% defective Part = 5/400 = 1.25%**

# Bayes Theorem - Introduction

**With Bayes Theorem**

Let us use Bayes theorem to get this probability:



$$P(\text{Defect} \mid \text{Mach2}) = \frac{P(\text{Mach2} \mid \text{Defect}) * P(\text{Defect})}{P(\text{Mach2})}$$

**P (Defect | Mach2) = (0.5 * 0.01) / 0.4 = 0.0125**

**i.e. 1.25% probability that the machine 2 will produce a defective part.**

Basically,

**If Machine 2 produces 10 thousand wrenches then 125 of them are defective.**

# Bayes Theorem - Introduction

**Quick Exercise :**
**What is the probability of a part being defective, given that it came from Machine 1 ?**

**P(Defect | Mach1) = ?**

**Answer :**

# Bayes Theorem - Introduction

**Why to use this Bayes Theorem ?**

The only question that arises is, why to go through all these complexities ? Why cannot we simply count the number of wrenches and calculate the probabilities and get the same result ?

**Obvious Question :**
**If the items are labeled, why couldn’t we just count the number of defective wrenches that came from Machine 2 and divide by the total number that came from Machine 2 ? Because that is exactly what we did without theorem.**

Well, it might be time consuming.
Sometimes you might not have access to those numbers/information

# Naïve Bayes Classifier

# Naïve Bayes Classifier

**Naïve Bayes Classifier is a Probabilistic type of Classifier.**
**Here first the probabilities are calculated based on Bayes theorem;**
**Based on those probabilities, the new observation points are classified into categories accordingly.**

So lets see how we are going to apply Bayes Theorem to create a machine Learning algorithm.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

# Naïve Bayes Classifier

Let’s say we have a dataset of 30 observations of Salary vs Age.

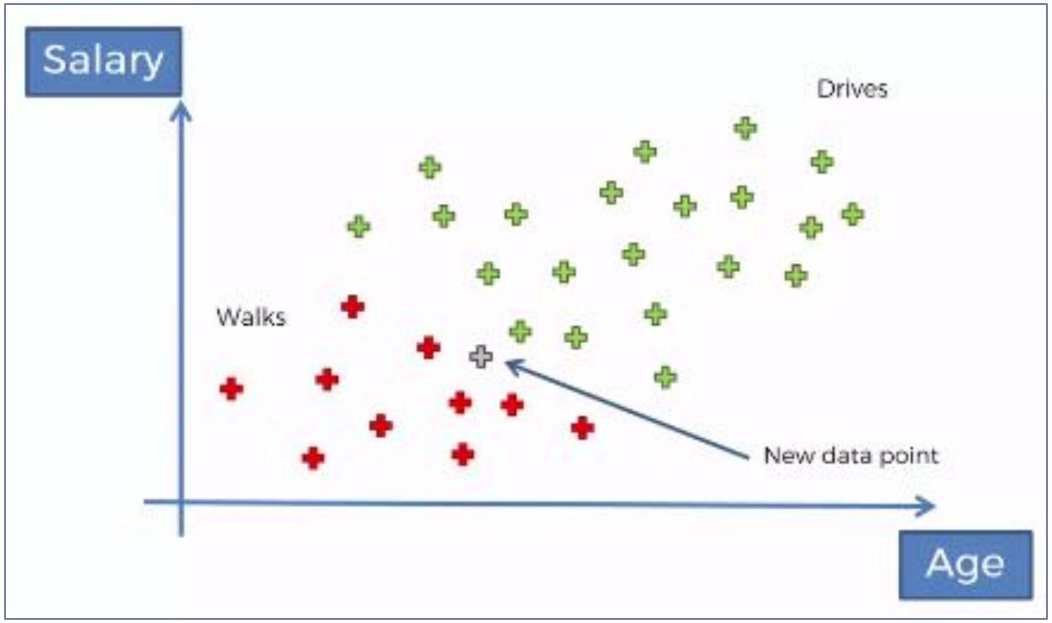
So we have 2 categories:

**Red – Person walks to work**

**Green – Person drives to work**

And the question here is,

What happens if we add a new observation to this dataset ? How would it be classified ?



# Naïve Bayes Classifier – How it works ?

How do we apply Bayes Theorem here and create a classifier ?

**We take the Bayes Theorem and apply it twice.**

**STEP 1:**

**First time:** To find out what is the probability that the person (new observation) walks given his features.

X over here is the features of the new observation (person)

Here we have only Age and Salary as features, but in reality there can be many.

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

# Naïve Bayes Classifier – How it works ?

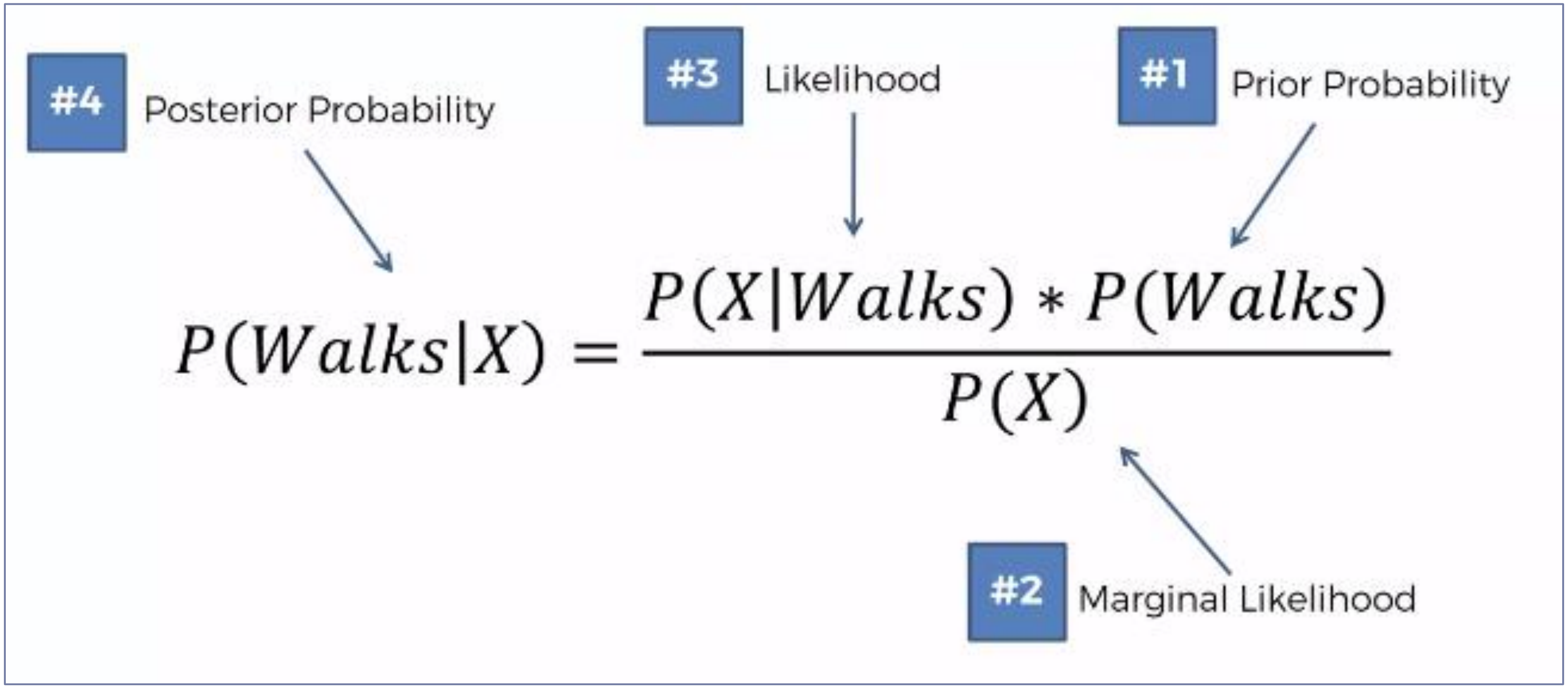
How do we apply Bayes Theorem here and create a classifier ?

**STEP 2:**

**Second time:** To find out what is the probability that the person (new observation) drives given his features.

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

# Naïve Bayes Classifier – How it works ?



$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

# Naïve Bayes Classifier – How it works ?

How do we apply Bayes Theorem here and create a classifier ?

**STEP 3:**

Compare

P(Walks | X) vs. P(Drives | X)

# Naïve Bayes Classifier – How it works ?

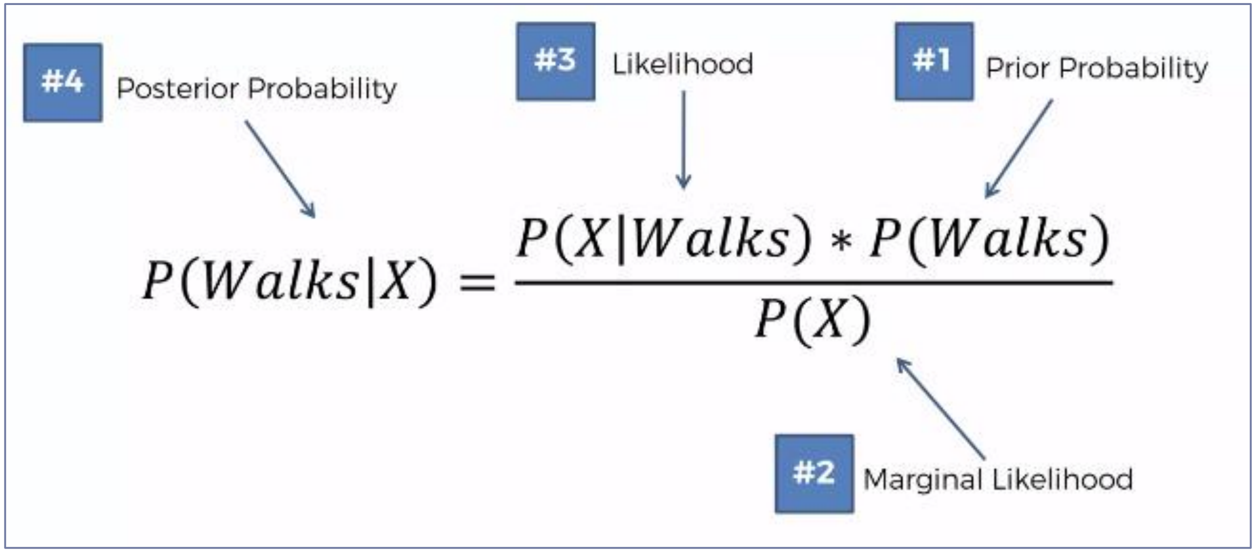
Let’s perform those steps and classify the new observation point.

**Step 1: Probability of Person walks to work / P(Walks | X)**

Here we calculate 3 types of probability :

1. **Prior Probability**
2. **Marginal Likelihood**
3. **Likelihood**

Based on above three, we get Posterior Probability

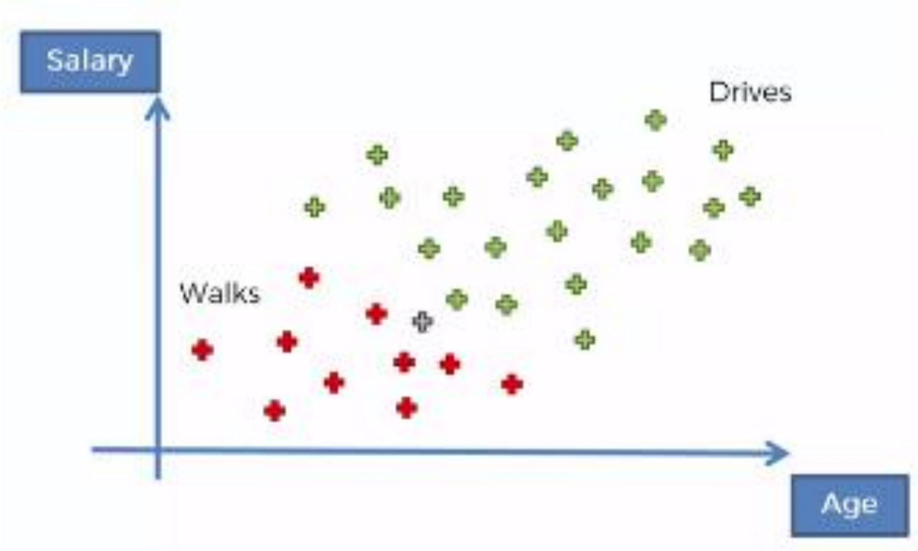


$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

# Naïve Bayes Classifier – How it works ?

**Step 1: Probability of Person walks to work / P(Walks | X)**

**1.1 Prior Probability / P(Walks)**



**P (Walks) = Number of Walkers / Total Observations**

P(Walks) = 10/30

# Naïve Bayes Classifier – How it works ?

**Step 1: Probability of Person walks to work / P(Walks | X)**

**1.2 Marginal Likelihood/ P(X):**

**Tells you the likelihood of the new point falling inside the circle**

- Select a radius and draw a circle around some observations.
- This radius we need to select on our own and we need to decide this. This is the input parameter for the algorithm.
- Now we look all the points inside the radius and deem them similar in terms of features.
- Hence the new features will be classified similar to the points in the circle.
- And now we calculate the probability (Marginal Likelihood) for the observations within the circle.

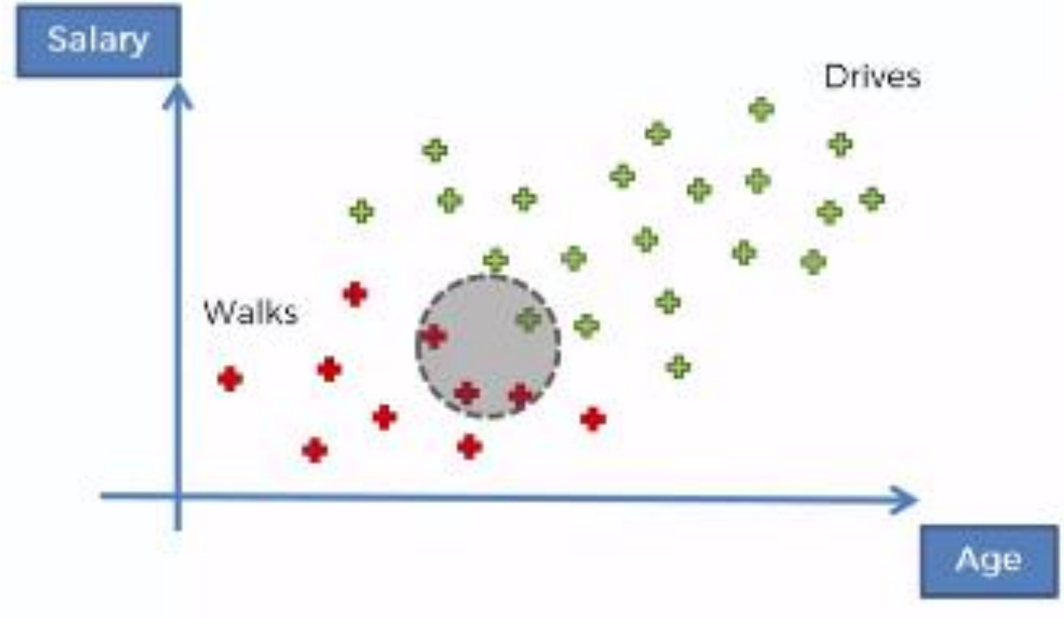
$$P(X) = \frac{Number\ of\ Similar\ Observations}{Total\ Observations}$$

# Naïve Bayes Classifier – How it works ?

**Step 1: Probability of Person walks to work / P(Walks | X)**

**1.2 Marginal Likelihood/ P(X):**

**Tells you the likelihood of the new point falling inside the circle**



**P (X) = Number of Similar Observations / Total Observations**

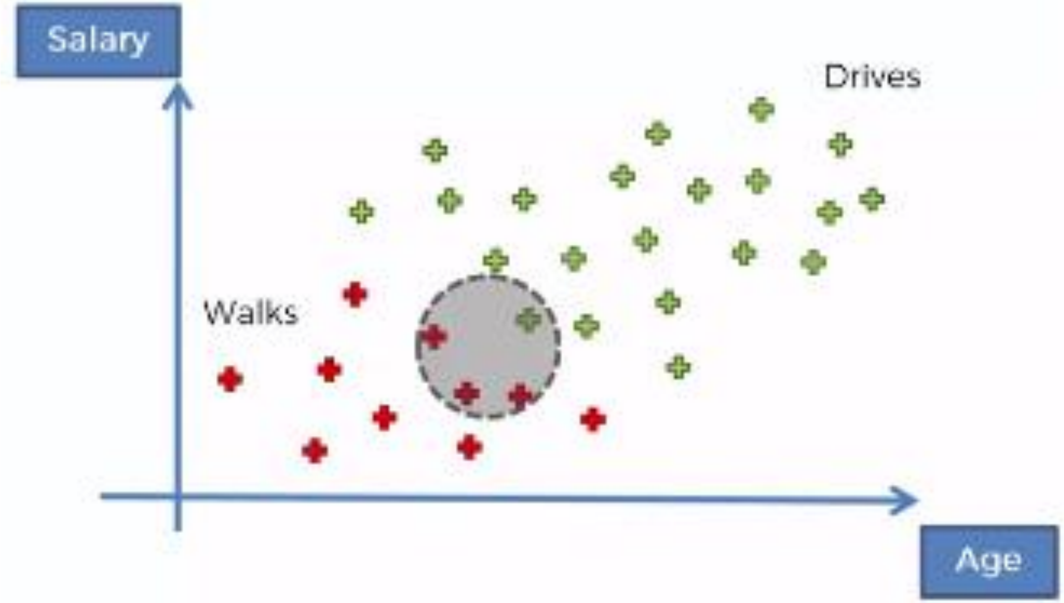
P(Walks) = 4 / 30

# Naïve Bayes Classifier – How it works ?

**Step 1: Probability of Person walks to work / P(Walks | X)**

**1.3 Likelihood/ P(X | Walks):**

**Tells you the likelihood of the new person who walks exhibits features X.**



**P (X | Walks) = Number of Walking Observations (in circle) / Total walkers**

P(Walks) = 3 / 10

# Naïve Bayes Classifier – How it works ?

**Step 1: Probability of Person walks to work / P(Walks | X)**

Now calculating the Posterior Probability:

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

**P (Walks | X) = [(3/10) * (10/30)] / 4/30**
P (Walks | X) = 0.75

**STEP 1 - Done**

# Naïve Bayes Classifier – How it works ?

**Step 2: Probability of Person drives to work / P(Drives | X)**

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

**P (Drives | X) = [(1/20) * (20/30)] / 4/30**
P (Drives | X) = 0.25

**STEP 2 - Done**

**Hint: If you have only 2 classes to categorize, then you need not to calculate the step 2 probability.**
You already get some probability in Step 1 and if it is less than 50% then the new observations will be marked as other category otherwise if it is greater than 50% then the same category.

# Naïve Bayes Classifier – How it works ?

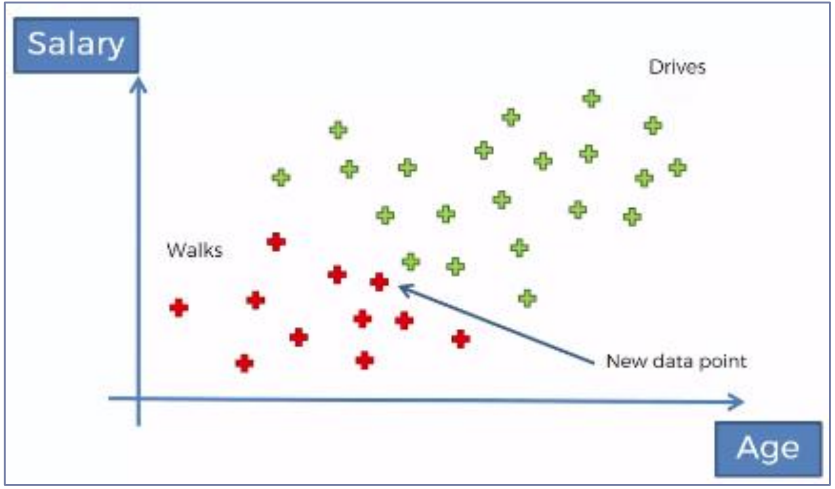
**Step 3: Compare**

**P (Walks | X) vs. P(Drives | X)**

0.75 > 0.25

P (Walks | X) > P(Drives | X)

Thus there is 25% chance that the new observation (person) drives to works, but there is 75% change that the person walks to work.
Hence we will classify the person as **'person walks to work'** as probability of walking to work is greater than driving to work.



# Naïve Bayes Classifier – Zero Frequency

If a categorical variable has a category in test data set which was not observed in training data set, then the model will assign a **zero probability**.

It will not be able to make a prediction. This is often known as "**Zero Frequency**".

- To solve this, we can use the smoothing technique.
- One of the simplest smoothing techniques is called **Laplace estimation**.
- Sklearn applies Laplace smoothing by default when you train a Naive Bayes classifier

# Different Sklearn NB classifiers

**1. Gaussian Naive Bayes :** (Features are continuous)
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. **These features are continuous**
We can’t represent features in terms of their occurrences.
**Hence we use Gaussian Naive Bayes here**.

# Different Sklearn NB classifiers

**2. Multinomial Naive Bayes :**

- Its is used when we have discrete data

**Example:**

Movie ratings ranging 1 and 5 as each rating will have **certain** frequency to represent.

In text learning we have the count of each word to predict the class or label.

**3. Bernoulli Naive Bayes :**

- It assumes that all our features are binary such that they take only two values.

**Example:**

0s - represent “word does not occur in the document”

1s - represent "word occurs in the document" .

# Advantages of Naïve Bayes

- They are extremely fast for both training and prediction
- They provide straightforward probabilistic prediction
- They are often very easily interpretable
- They have very few (if any) tunable parameters
- Works well with small datasets

# When to use Naïve Bayes ?

- Naïve Bayes is often a good choice as an initial baseline classification. If it performs suitably, then congratulations: you have a very fast, very interpretable classifier for your problem
- When the naive assumptions actually match the data (very rare in practice); e.g. features are totally independent of each other.
- It works well for less complex datasets i.e. when the categories are very well separated in dataset.