



NAÏVE BAYES ALGORITHMS

Instructor: Aidos Sarsembayev, Alexey Leshev



Outline:

- What are Naïve Bayes Algorithms?
- How NB works?
- Advantages and disadvantages of using NB
- Gaussian, Multinomial, and Bernoulli NB models.

What are Naïve Bayes Algorithms?

- Naïve Bayes algorithms are among the most famous supervised learning algorithms.
- They are used in classification.
- NB is inherently multiclass – i.e. you can easily apply it to cases where you have more than one type of output.
- They are extremely easy to build and very useful for very large data sets.

Bayes Theorem

3. Likelihood

1. Prior Probability

4. Posterior probability

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

2. Marginal Likelihood

Why naïve?

- Because Bayes theorem requires some independent assumptions
- Bayes theorem is the base for Naïve Bayes machine learning algorithm
- Hence Naïve Bayes also relies on these assumptions which are often not correct, and therefore – NAÏVE.

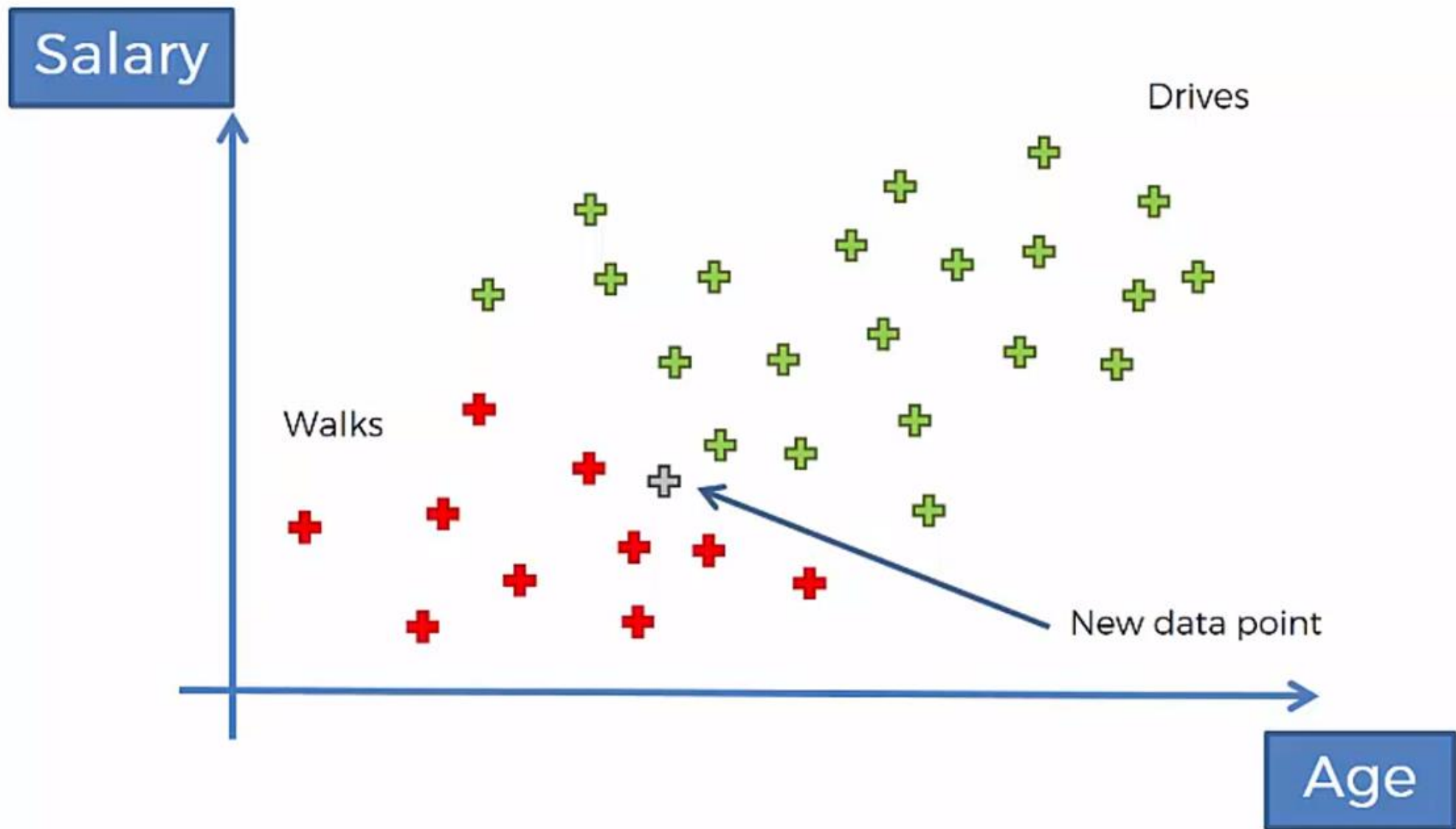
Naïve Bayes

- Naïve Bayes Assumption:

- **Features are independent** given class (This is a strong assumption):
- $P(X_1, X_2 | c_j) = P(X_1 | c_j)P(X_2 | c_j) = P(X_1 | c_j) * P(X_2 | c_j)$

$$P(x_1, x_2, \dots, x_n | c_j) = \prod_i P(x_i | c_j)$$
$$c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_i P(x_i | c_j)$$

‘Walking Dead’ example =)



- According to Naïve Bayes algorithm the 'Age' and 'Salary' parameters should be independent from each other
- However this is not always case. Also here...

Bayes Theorem applied to the current example

3. Likelihood

1. Prior Probability

4. Posterior probability

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

2. Marginal Likelihood

Bayes Theorem applied to the current example (1/3)

3. Likelihood

1. Prior Probability

4. Posterior probability

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

2. Marginal Likelihood

X – are the features of a some particular datapoint

Bayes Theorem applied to the current example (2/3)

3. Likelihood

1. Prior Probability

4. Posterior probability

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

2. Marginal Likelihood

X – are the features of a some particular datapoint

Bayes Theorem applied to the current example (3/3)

$$P(Walks|X) = P(Drives|X)$$

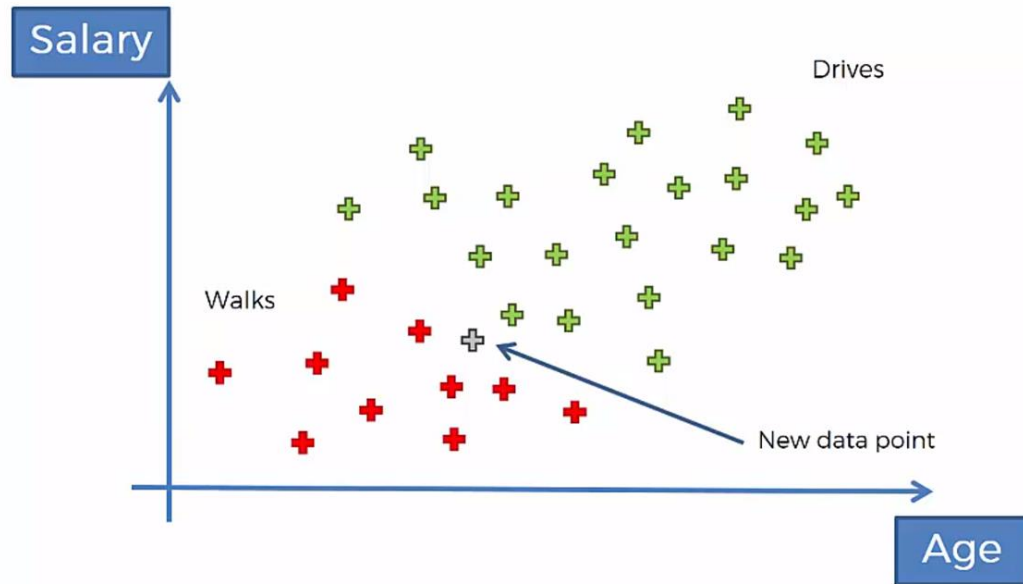
Step #1

#1. $P(\text{Walks})$

Q: what is the probability that the person we add to the dataset walks to work?

$$P(\text{Walks}) = \frac{\text{Number of Walkers}}{\text{Total Observations}}$$

$$P(\text{Walks}) = \frac{10}{30}$$



Bayes Theorem applied to the current example (2/3)

3. Likelihood

1. Prior Probability

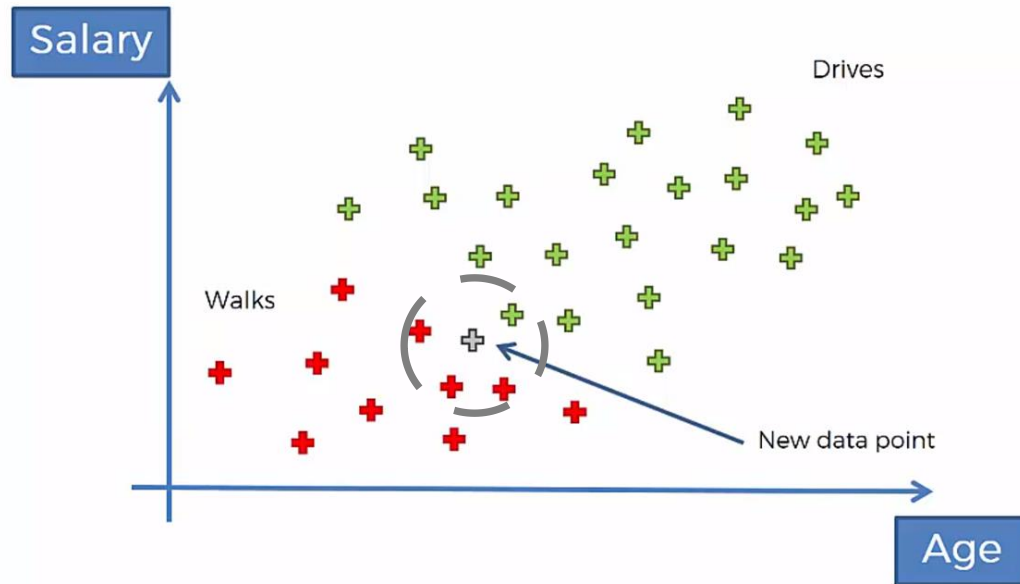
4. Posterior probability

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

2. Marginal Likelihood

X – are the features of a some particular datapoint

Step #2



#2. $P(X)$

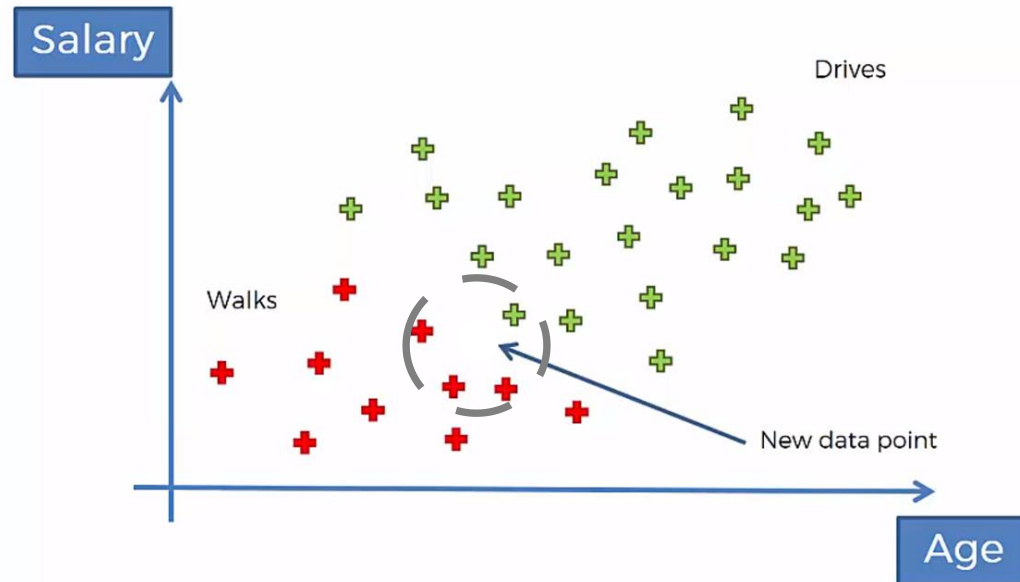
Q: what is the probability of a new point we add to the dataset being similar in features to the point that we are actually adding to the dataset?

OR simply – what is P for a new point to fall into this circle?

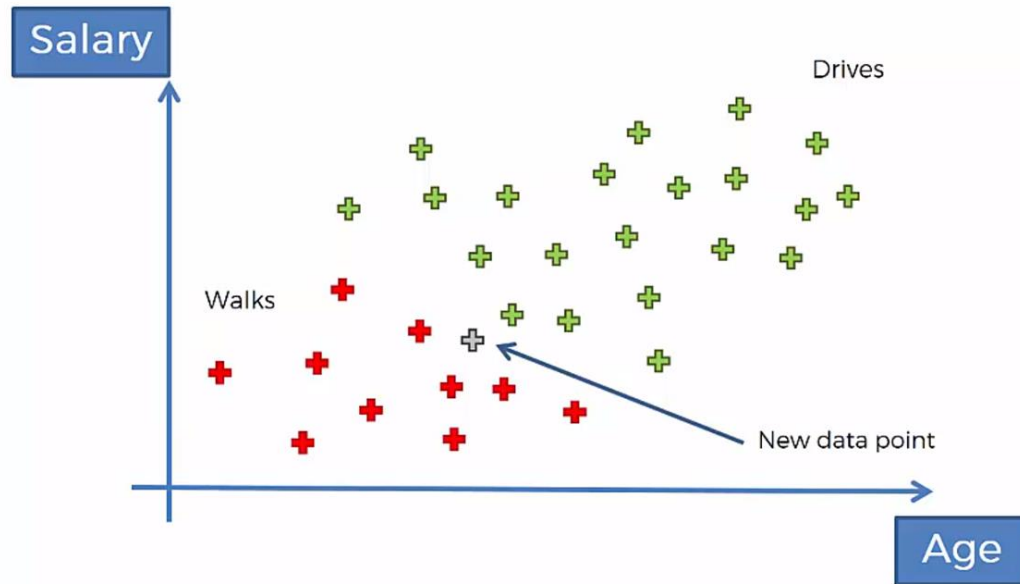
Step #2

#2. $P(X)$

Let's get rid of the new observation for a moment and assume that all of the four observation have similar features



Step #2



#2. $P(X)$

Let's get rid of the new observation for a moment and assume that all of the four observations have similar features

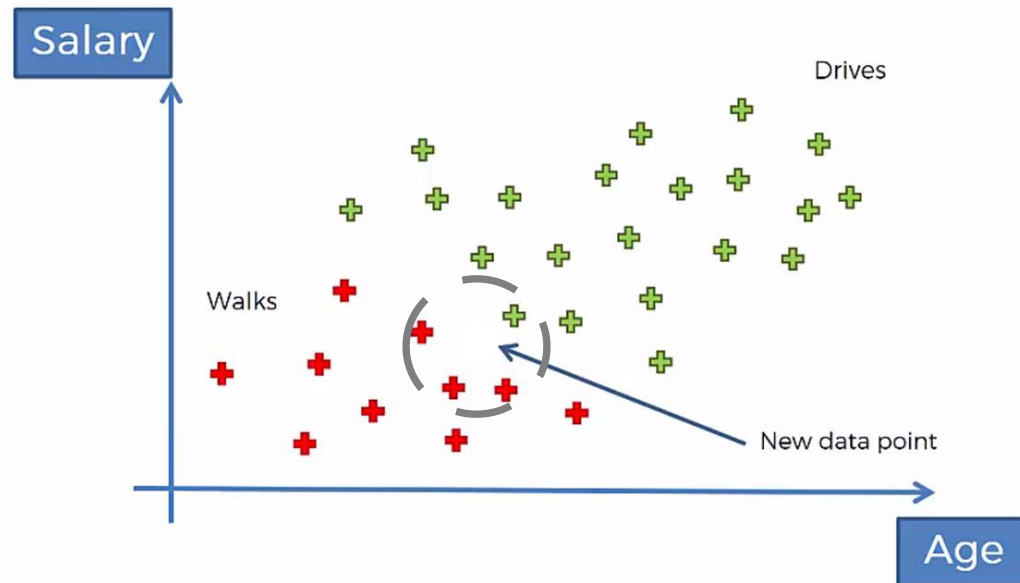
$$P(X) = \frac{\text{Number of similar observations}}{\text{Total observations}}$$

$$P(X) = \frac{4}{30}$$

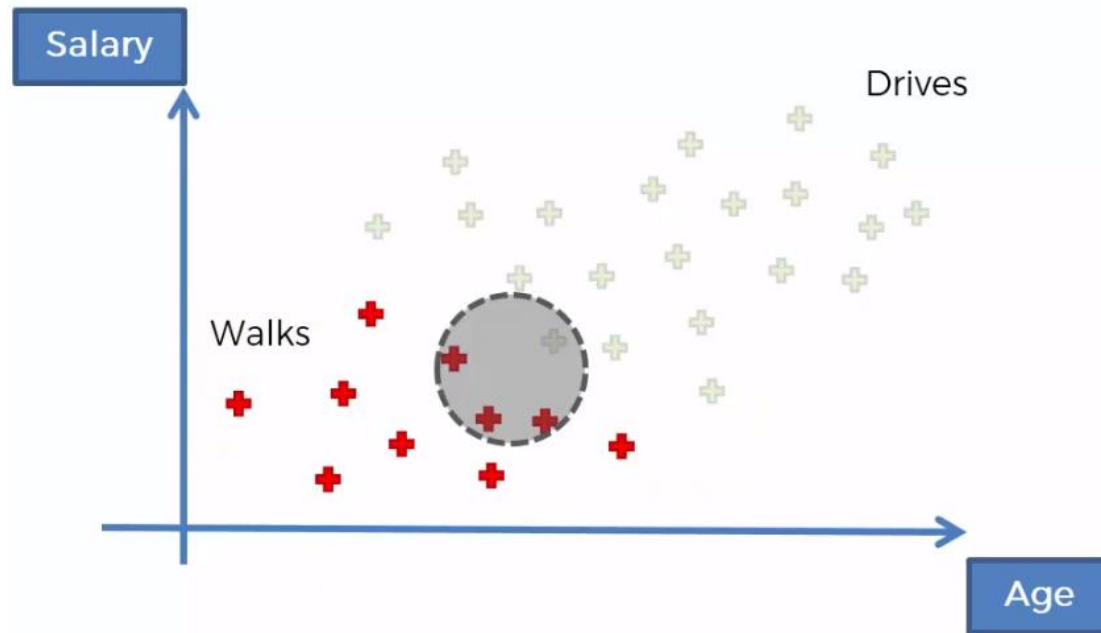
Step #3

#2. $P(X | \text{Walks})$

Q: What is the likelihood that somebody who walks exhibits features X?



Step #3



#2. $P(X | \text{Walks})$

Q: What is the likelihood that somebody who walks exhibits features X?

$$P(X) = \frac{\text{Number of similar observations among those who Walk}}{\text{Total number of Walkers}}$$

$$P(X) = \frac{3}{10}$$

Plug all in

$$P(Walks|X) = \frac{\frac{3}{10} * \frac{10}{30}}{\frac{4}{30}} = 0,75$$

Try it yourself ...

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

Plug all in

$$P(Drives|X) = \frac{\frac{1}{20} * \frac{20}{30}}{\frac{4}{30}} = 0,25$$

Compare

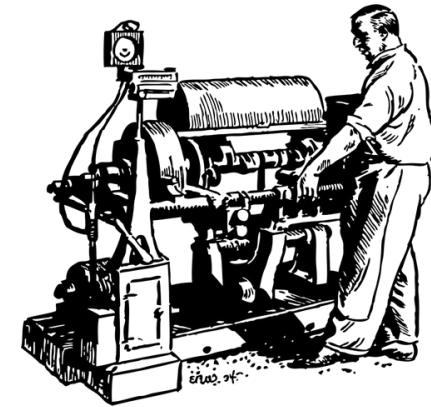
$$P(Walks|X) = P(Drives|X)$$

|

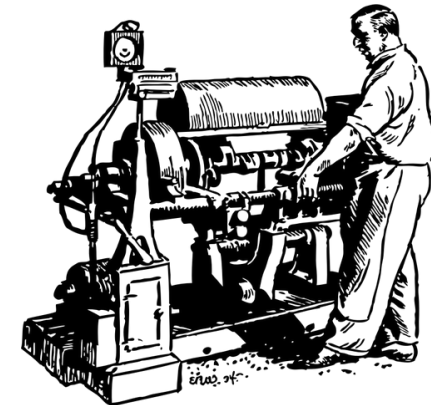
v

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

Another example



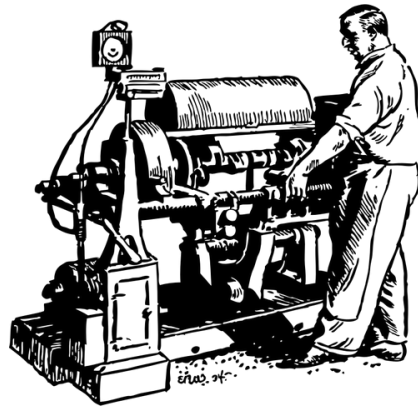
M1



M2

Spanners example

What's the P that M2 produces defective spanner?



What is given?

- M1: 30 spanners/hr
- M2: 20 spanners/hr
- Out of all produced parts:
- We can SEE that 1% are defective
- Out of all defective parts:
- We can SEE that 50% came from M1
- And 50% from M2
- Q: what is the probability that a part produced by M2 is defective?

What is given?

- M1: 30 spanners/hr
- M2: 20 spanners/hr
- Out of all produced parts:
 - We can SEE that 1% are defective
- Out of all defective parts:
 - We can SEE that 50% came from M1
 - And 50% from M2
- Q: what is the probability that a part produced by M2 is defective?

$$P(M1) = 30/50 = 0,6$$

$$P(M2) = 20/50 = 0,4$$

What is given?

- M1: 30 spanners/hr
- M2: 20 spanners/hr
- Out of all produced parts:
- We can SEE that 1% are defective

$$P(M1) = 30/50 = 0,6$$

$$P(M2) = 20/50 = 0,4$$

$$P(\text{Defect}) = 1\% \text{ OR } 0,01$$

- Out of all defective parts:
- We can SEE that 50% came from M1
- And 50% from M2
- Q: what is the probability that a part produced by M2 is defective?

What is given?

- M1: 30 spanners/hr
 - M2: 20 spanners/hr
- $P(M1) = 30/50 = 0,6$
 $P(M2) = 20/50 = 0,4$
- Out of all produced parts:
 - We can SEE that 1% are defective
- $P(\text{Defect}) = 1\% \text{ OR } 0,01$
- Out of all defective parts:
 - We can SEE that 50% came from M1
 - And 50% from M2
- $P(M1 | \text{Defect}) = 50\%$
 $P(M2 | \text{Defect}) = 50\%$
- Q: what is the probability that a part produced by M2 is defective?
- $P(\text{Defect} | M2) - ???$

We consider only M2

- M2: 20 spanners/hr
- We can SEE that 1% are defective
- And 50% from M2
- Q: what is the probability that a part produced by M2 is defective?

$$\begin{aligned}P(M2) &= 20/50 = 0,4 \\P(\text{Defect}) &= 1\% \text{ OR } 0,01 \\P(M2 | \text{Defect}) &= 50\% \\P(\text{Defect} | M2) &= ???\end{aligned}$$

$$P(\text{Defect}|M2) = \frac{P(M2|\text{Defect}) * P(\text{Defect})}{P(M2)}$$

Plug in

- M2: 20 spanners/hr
- We can SEE that 1% are defective
- And 50% from M2
- Q: what is the probability that a part produced by M2 is defective?

$$\begin{aligned}P(M2) &= 20/50 = 0,4 \\P(\text{Defect}) &= 1\% \text{ OR } 0,01 \\P(M2 | \text{Defect}) &= 50\% \\P(\text{Defect} | M2) &= ???\end{aligned}$$

$$P(\text{Defect} | M2) = \frac{0,5 * 0,01}{0,4} = 0,0125 = 1.25\%$$

The solution

$$P(Defect|M2) = \frac{0,5*0,01}{0,4} = 0,0125 = 1.25\%$$

For example:

- 1000 spanners
- 400 from M2
- 1% have a defect = 10
- 50% of them from M2 = 5
- Number of parts with defects from M2 -> $5/400 = 1,25\%$

Example. 'Play Tennis' data

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

Question: For the day <sunny, cool, high, strong>, what's the play prediction?

Example. 'Play Tennis' solution

$$P(\text{PlayTennis} = \text{yes}) = 9 / 14 = 0.64$$

$$P(\text{PlayTennis} = \text{no}) = 5 / 14 = 0.36$$

$$P(\text{Wind} = \text{strong} | \text{PlayTennis} = \text{yes}) = 3 / 9 = 0.33$$

$$P(\text{Wind} = \text{strong} | \text{PlayTennis} = \text{no}) = 3 / 5 = 0.60$$

etc.

$$P(\text{yes})P(\text{sunny} | \text{yes})P(\text{cool} | \text{yes})P(\text{high} | \text{yes})P(\text{strong} | \text{yes}) = 0.0053$$

$$P(\text{no})P(\text{sunny} | \text{no})P(\text{cool} | \text{no})P(\text{high} | \text{no})P(\text{strong} | \text{no}) = \mathbf{0.0206}$$

⊢ answer : $\text{PlayTennis}(x) = \text{no}$

Underflow Prevention

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since $\log(xy) = \log(x) + \log(y)$, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized log probability score is still the most probable.

$$c_{NB} = \operatorname{argmax}_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j)$$

A few issues with Bayes Algorithm

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability to it and will be unable 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.
- This is parameter “alpha” in BernoulliNB and MultinomialNB in python. Default is “alpha = 1”
- Naive Bayes a bad estimator – don’t take probability estimations seriously!

Different Types of NB models in Python

- What does dictate the type of NB model you can use? The answer is your feature types. Depending on features, you can use 3 different NB models: Gaussian, Bernoulli, and Multinomial
 - *If feature space is quantitative then you shall use GaussianNB*
 - *If feature space is Binary, then you better use BernoulliNB*
 - *If feature space is discrete counts, then you can use MultinomialNB.*
 - *Can you come up with few examples for each class of NB?*

Gaussian NB

- Your assumption for Gaussian NB is, your feature variables are independent and Normally distributed.
- You should make sure, your input features either look normal at their raw format, or look normal after transformation. For instance you can use log transform to make most of positively skewed distributions, symmetric.
- For Gaussian NB, we first need to estimate the mean and Standard deviation of each feature.
- Once you successfully calibrate your model, you can use probability density function of Normal Distribution to calculate likelihood functions.

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp \left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2} \right)$$

Advantages of using NB algorithm

- Very easy to compute
- Great for multi-class cases
- One of the fastest algorithms
- There is no parameter needed to be tuned. (an exception might be alpha – which by definition is not a tuning parameter).
- The algorithms can be used for **real time** prediction.
- Used often in Text Classification/Spam Filtering/Sentiment Analysis.

Issues with NB

- It works under the strong assumption that your feature inputs are independent.
- If you have highly dependent variables, you must drop one.
- GaussianNB works under the assumption that your inputs are normally distributed. If that is not the case, you either cannot use it or need to transform your variables.
- You cannot take probability predictions seriously!

Summary

- Definition of NB algorithm
- Learned how NB algorithm works
- Gaussian NB, Multinomial NB and Bernoulli NB
- Limitations and advantages of NB

FIN.