Naïve Bayes

What happens under the hood

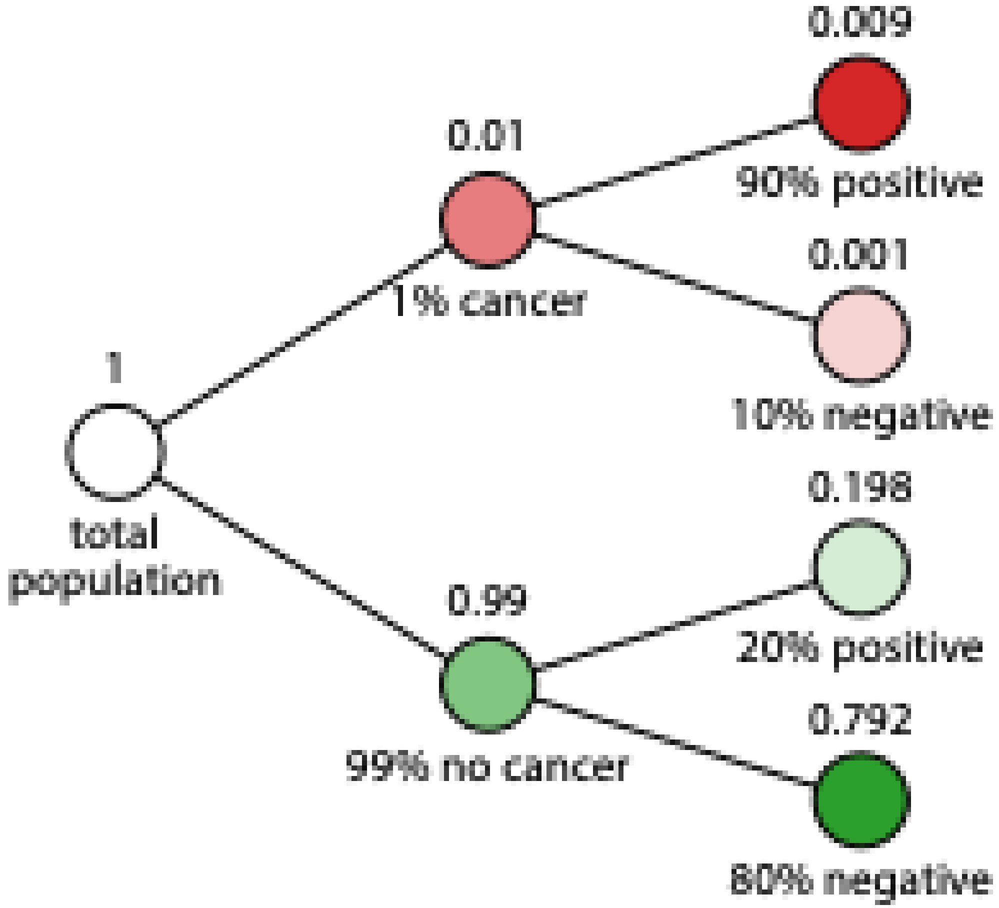
In the class…

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

- “We can normalize the probability” ← What does that mean?

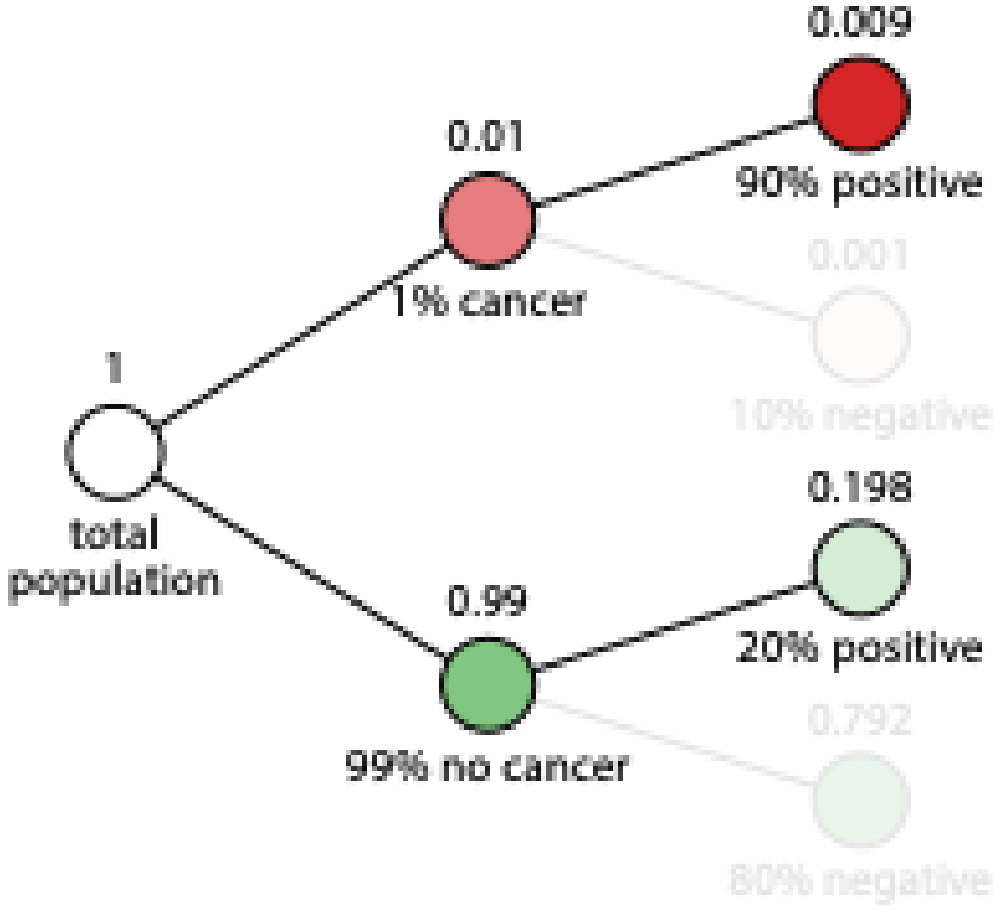
Cancer test

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$



Cancer test

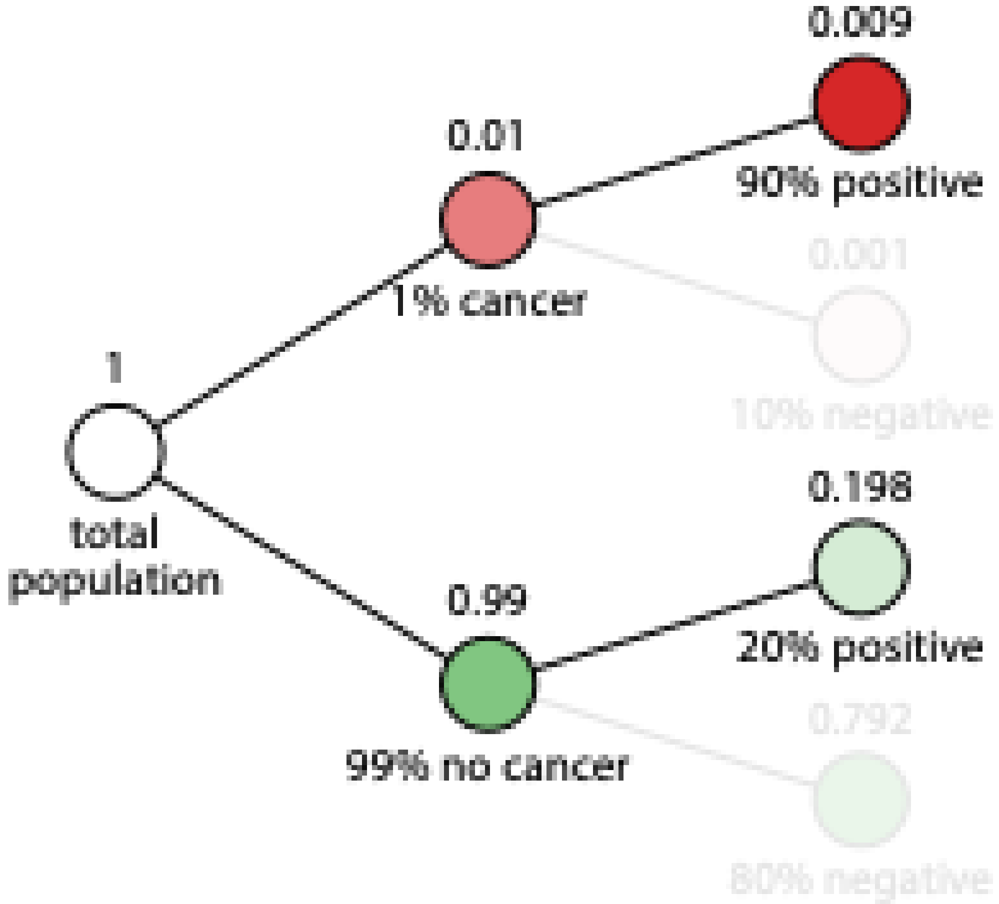
$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$



- How worried you should be if you're tested positive?

Cancer test

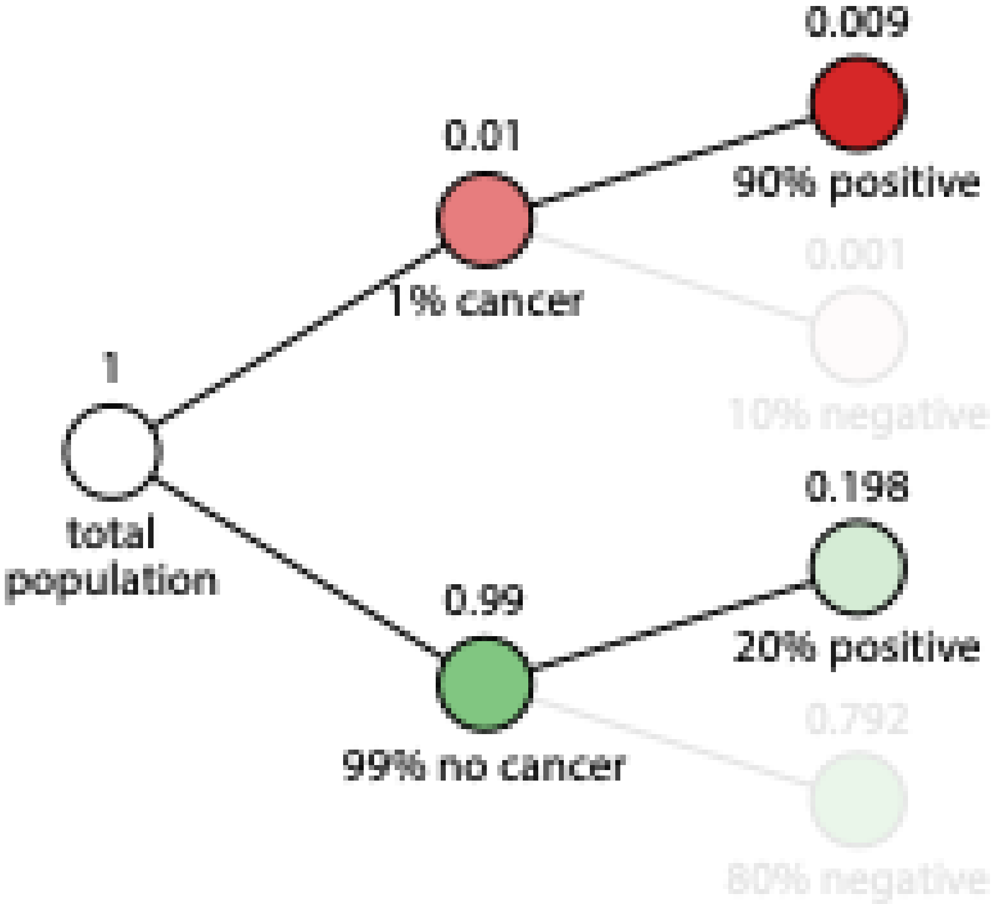
$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$



- How worried you should be if you're tested positive?
- Method 1:
  Normalize probability

Cancer test

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$



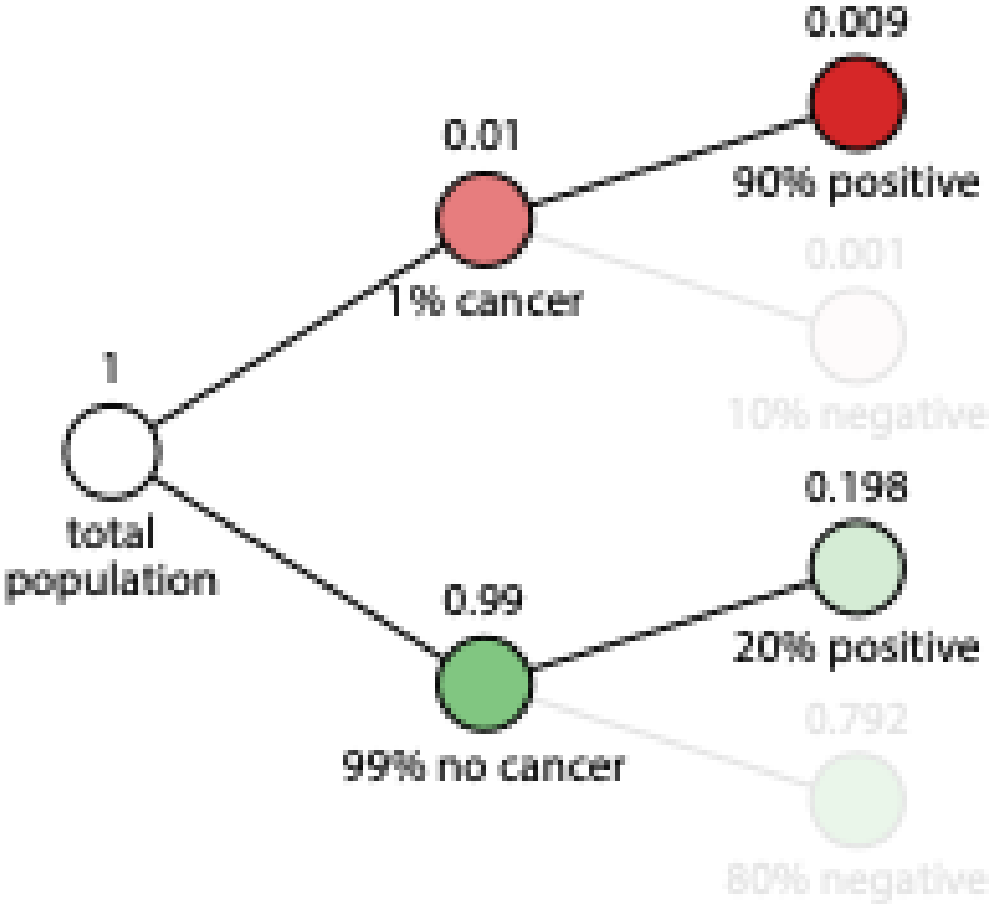
- How worried you should be if you're tested positive?
- Method 1:
  Normalize probability

$$P(+|cancer)P(cancer)$$
$$= 0.90 * 0.01 = 0.009$$
$$P(+|no\ cancer)P(no\ cancer)$$
$$= 0.20 * 0.99 = 0.198$$

$$P(cancer|+) = \frac{0.009}{0.009 + 0.198} = 0.043$$

Cancer test

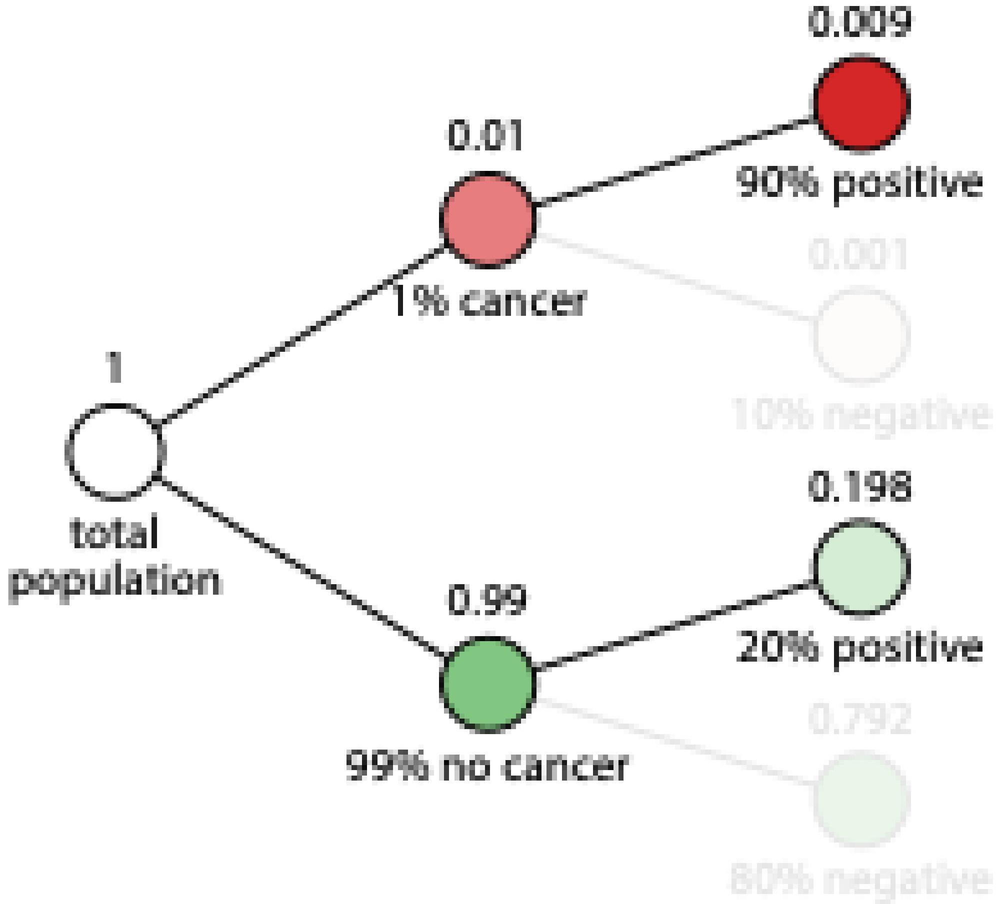
$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$



- How worried you should be if you're tested positive?
- Method 2:
  Calculate total probability

Cancer test

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$



- How worried you should be if you're tested positive?
- Method 2:
  Calculate total probability

$$P(+|cancer)P(cancer)$$
$$= 0.90 * 0.01 = 0.009$$
$$P(+|no\ cancer)P(no\ cancer)$$
$$= 0.20 * 0.99 = 0.198$$
$$P(+) = 0.009 + 0.198$$

$$P(cancer|+) = \frac{0.009}{0.009 + 0.198} = 0.043$$

Mathematically speaking

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

- “We can normalize the probability”

$$P(x) = \sum_{C} P(x|C)P(C)$$

In the class…

```
model = GaussianNB()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

- ↑ What does that actually do?

Cancer and aging

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

- Ten patients

Age	10	20	30	40	50	60	70	80	90	100
Cancer	no	no	no	yes	no	yes	yes	yes	yes	yes

Cancer and aging

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

- Ten patients

Age	10	20	30	40	50	60	70	80	90	100
Cancer	no	no	no	yes	no	yes	yes	yes	yes	yes

- $P(cancer) = 0.6, P(no\ cancer) = 0.4$

Cancer and aging

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

- Ten patients

Age	10	20	30	40	50	60	70	80	90	100
Cancer	no	no	no	yes	no	yes	yes	yes	yes	yes

- $P(cancer) = 0.6, P(no\ cancer) = 0.4$
- $\mu(age|cancer) = 73$
- $\sigma(age|cancer) = 20$
- $\mu(age|no\ cancer) = 28$
- $\sigma(age|no\ cancer) = 15$

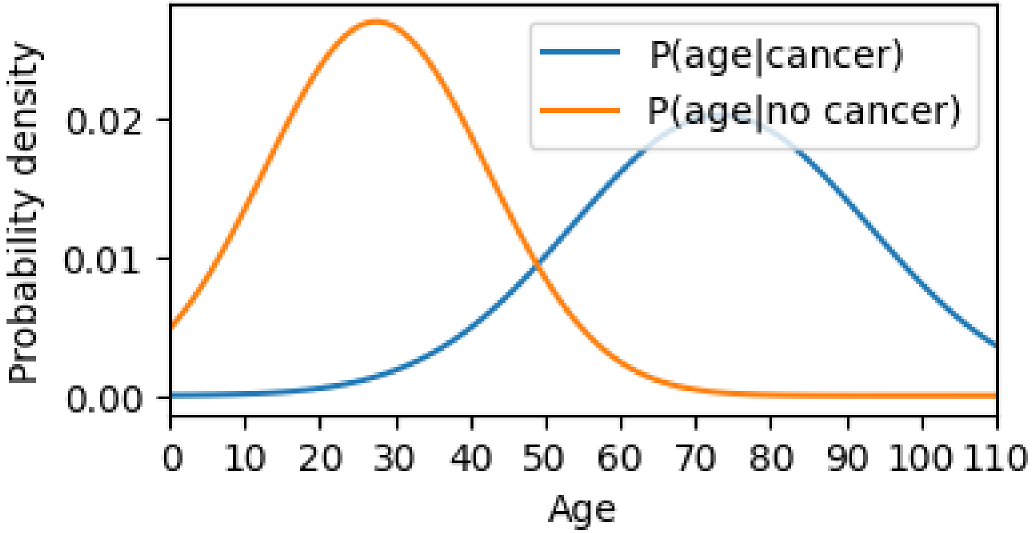
Cancer and aging

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

- Ten patients

Age	10	20	30	40	50	60	70	80	90	100
Cancer	no	no	no	yes	no	yes	yes	yes	yes	yes

- $P(cancer) = 0.6, P(no\ cancer) = 0.4$
- $\mu(age|cancer) = 73$
- $\sigma(age|cancer) = 20$
- $\mu(age|no\ cancer) = 28$
- $\sigma(age|no\ cancer) = 15$



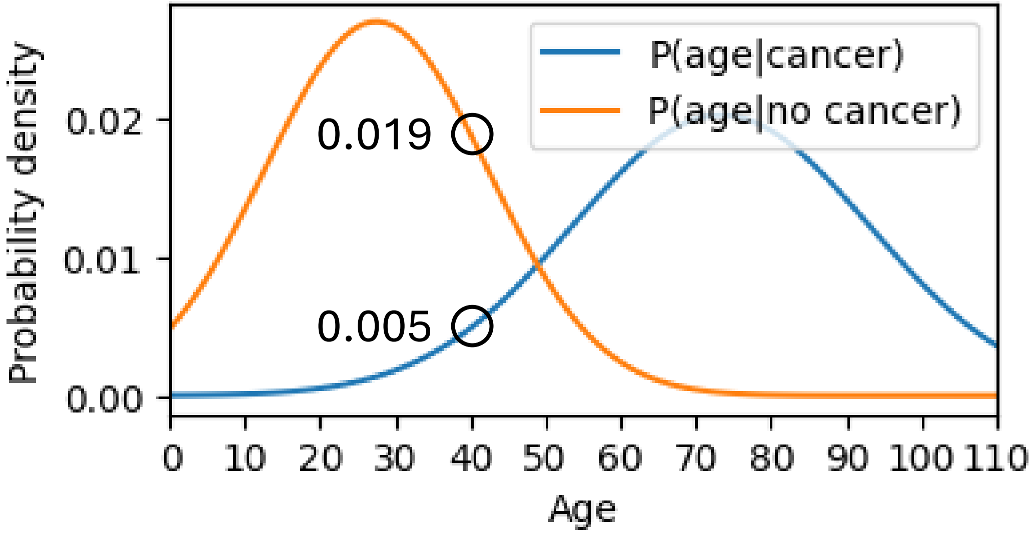
Cancer and aging

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

- Ten patients

Age	10	20	30	40	50	60	70	80	90	100
Cancer	no	no	no	yes	no	yes	yes	yes	yes	yes

- $P(cancer) = 0.6, P(no\ cancer) = 0.4$
- $P(cancer|age = 40) = ?$



Cancer and aging

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

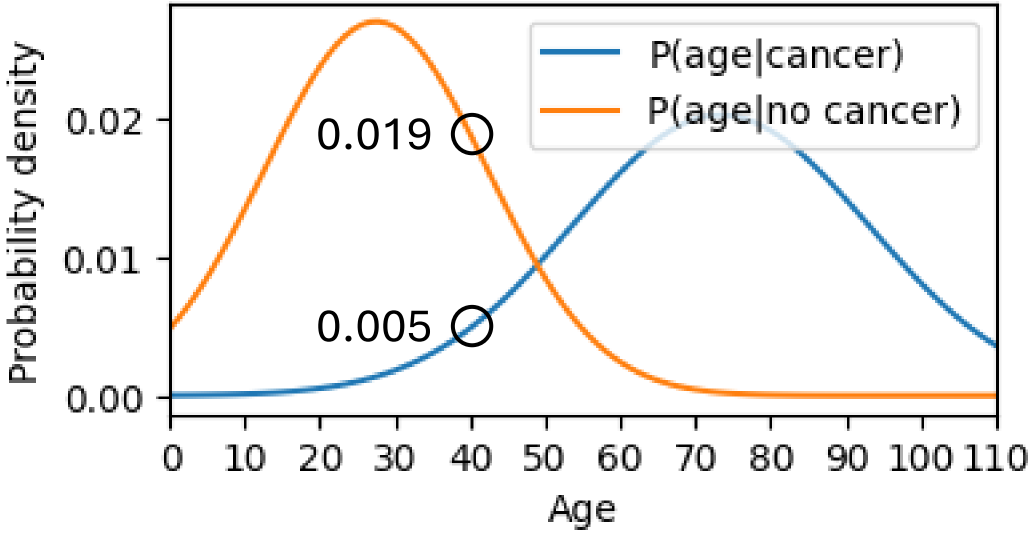
- Ten patients

Age	10	20	30	40	50	60	70	80	90	100
Cancer	no	no	no	yes	no	yes	yes	yes	yes	yes

- $P(cancer) = 0.6, P(no\ cancer) = 0.4$
- $P(cancer|age = 40)$

$$= \frac{0.005*0.6}{0.005*0.6+0.019*0.4}$$

$$= 28\%$$



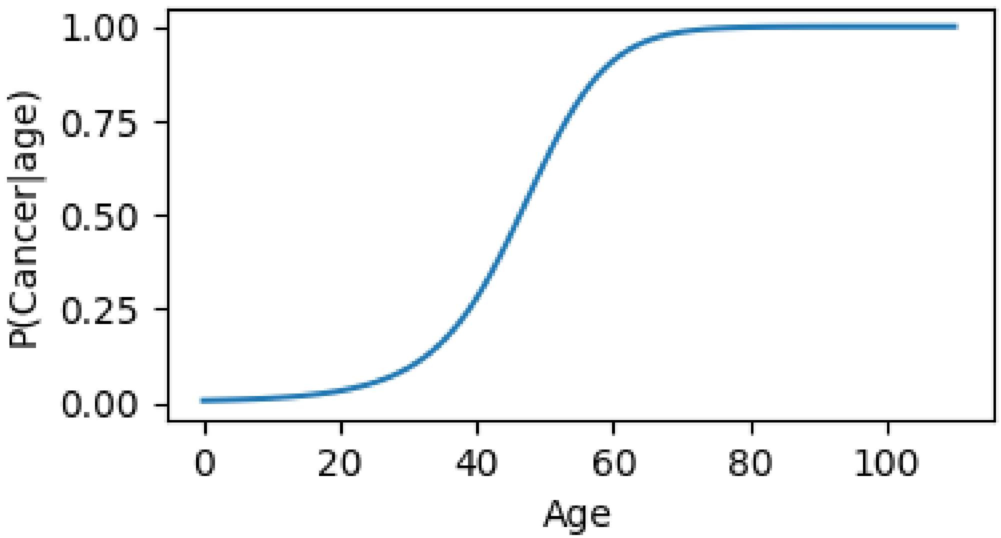
Cancer and aging

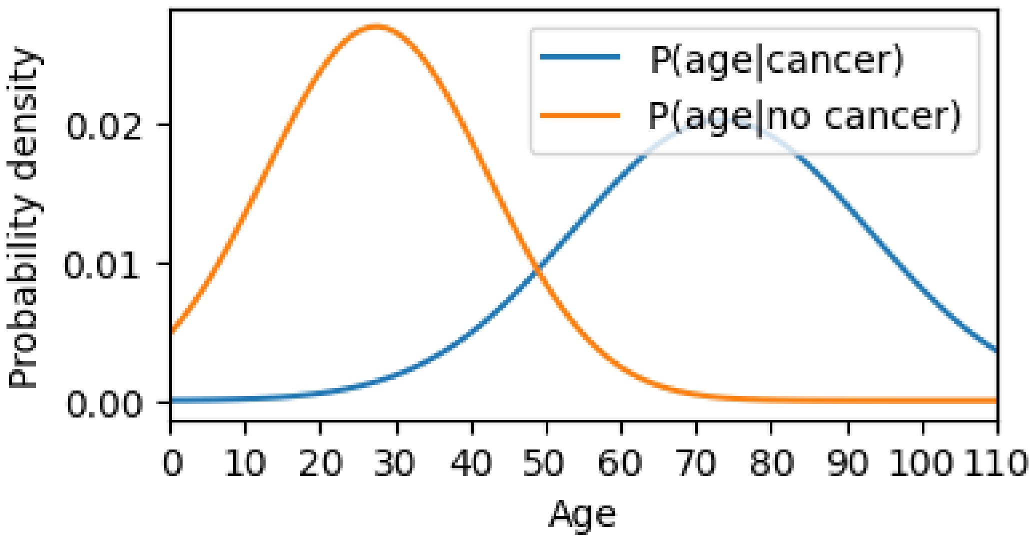
$$P(C|x) = \frac{P(x|C)P(C)}{P(x)} \sim P(x|C)P(C)$$

- Ten patients

Age	10	20	30	40	50	60	70	80	90	100
Cancer	no	no	no	yes	no	yes	yes	yes	yes	yes

- $P(cancer) = 0.6, P(no\ cancer) = 0.4$





Exercise

- Implement your own Gaussian Naïve Bayes!
  - Try it on the toxicity data in the lecture
  - Check if you get the same results (probabilities and predictions)

```python
class MyGaussianNaiveBayes:
    def fit(self, X, y):
        self.mean0 = np.mean(X[y == 0], axis=0)
        self.mean1 = np.mean(X[y == 1], axis=0)
        self.std0 = np.std(X[y == 0], axis=0)
        self.std1 = np.std(X[y == 1], axis=0)
        self.prior0 = np.mean(y == 0)
        self.prior1 = np.mean(y == 1)

    def _gaussian_pdf(self, X, mean, std):
        ...

    def predict_proba(self, X):
        gaussian0 = self._gaussian_pdf(X, self.mean0, self.std0)
        gaussian1 = self._gaussian_pdf(X, self.mean1, self.std1)

        likelihood0 = np.prod(gaussian0, axis=1)
        likelihood1 = np.prod(gaussian1, axis=1)

        posterior0 = likelihood0 * self.prior0
        posterior1 = likelihood1 * self.prior1

        total = posterior0 + posterior1
        return np.vstack((posterior0 / total, posterior1 / total)).T

    def predict(self, X):
        proba = self.predict_proba(X)
        return np.argmax(proba, axis=1)

gnb = MyGaussianNaiveBayes()
gnb.fit(X_train.values, y_train.values)
y_proba = gnb.predict_proba(X_test.values)
y_pred = gnb.predict(X_test.values)
```

Change these if you need

Write the Gaussian distribution

Confirm your predictions