

Introduktion til grundlæggende sandsynlighedsregning i Python

Af Henrik Sterner (henrik.sterner@gmail.com)

Sandsynlighedsregning er en måde at beskrive usikkerhed på, som er matematisk veldefineret. Sandsynlighedsregning spiller en central rolle i mange aspekter af maskinlæring, og derfor er det vigtigt at have en grundlæggende forståelse af sandsynlighedsregning.

Vi viser hvorledes sandsynlighedsregning kan implementeres i Python ved hjælp af biblioteket `numpy`.

Om brugen af disse slides

- ▶ Disse slides forsøger at eksemplifere en lang række af de vigtige begreber i sandsynlighedsregning
- ▶ De må på ingen måde kopieres uden tilladelse fra Henrik Sterner
- ▶ De er lavet i markdown og kan derfor nemt konverteres til andre formater
- ▶ Ved brug af Visual Studio Code kan de konverteres til HTML, PDF, PowerPoint, Word, LaTeX og mange andre formater
- ▶ Slides er tilgængelige på github.com/henriksterner/IntelligenteSystemer/

Konvertere slides til andre formater

- Fra markdown til pdf ved brug af pandoc:

```
pandoc -t beamer -V theme:metropolis -o sandsynlighedsteori.pdf sandsynlighedsteori.md
```

- Fra markdown til word ved brug af pandoc:

```
pandoc -o sandsynlighedsteori.docx sandsynlighedsteori.md
```

Sandsynlighedsregning - centrale begreber

- ▶ Udfaldsrum
- ▶ Hændelser
- ▶ Sandsynlighedsfunktioner
- ▶ Sandsynlighedsmål
- ▶ Stokastiske variable
- ▶ Betinget sandsynlighed
- ▶ Uafhængighed
- ▶ Bayes regel
- ▶ Naive Bayes algoritme

Udfaldsrum og hændelser

Et udfaldsrum er en mængde af mulige udfald. Et udfald er en mulig hændelse. Eksempler på udfaldsrum: * Hvis vi kaster en mønt, så er udfaldsrummet $\{P, K\}$. P står for plat, og K står for krone. * Hvis vi kaster to mønter, så er udfaldsrummet $\{(P, P), (P, K), (K, P), (K, K)\}$. * Et udfaldsrum kan være uendeligt stort, f.eks. hvis vi kaster en terning, så er udfaldsrummet $\{1, 2, 3, 4, 5, 6\}$. * Hvis vi kaster to terninger, så er udfaldsrummet $\{(1, 1), (1, 2), (1, 3), \dots, (6, 6)\}$. * Hvis vi kaster to terninger, og vi er interesserede i summen af øjnene, så er udfaldsrummet $\{2, 3, 4, \dots, 12\}$.

Hændelser

En hændelse er en delmængde af udfaldsrummet. Eksempler på hændelser:

- * Hvis vi kaster en mønt, så er hændelsen $\{P\}$.
- * Hvis vi kaster to mønter, så er hændelsen $\{(P, P), (P, K)\}$.
- * Hvis vi kaster to terninger, så er hændelsen $\{(1, 1), (1, 2), (1, 3), \dots, (6, 6)\}$.
- * Hvis vi kaster to terninger, og vi er interesserede i summen af øjnene, så er hændelsen $\{2, 3, 4, \dots, 12\}$.
- * Hvis vi kaster to terninger, og vi er interesserede i summen af øjnene, og summen er større end 10, så er hændelsen $\{11, 12\}$.
- * Hvis vi kaster to terninger, og vi er interesserede i summen af øjnene, og summen er større end 10, og den første terning viser 6, så er hændelsen $\{11\}$.

Hændelser og udfaldsrum i Python

Vi kan repræsentere et udfaldsrum som en liste:

```
U = [1,2,3,4,5,6]
```

Vi kan repræsentere en hændelse som en liste:

```
A = [1,2,3]
```

Vi kan teste om et element er i en liste ved hjælp af `in`:

```
1 in A
```

Vi kan teste om en liste er en delmængde af en anden liste ved hjælp af `issubset`:

```
A.issubset(U)
```

Matematisk definition af udfaldsrum og hændelser

Et udfaldsrum er en mængde U . En hændelse er en delmængde af U . Hvis A er en hændelse, så er $A \subseteq U$. Det læses som at A er en delmængde af U . Men A kan også være lig med U . I så fald siger vi at A er en sikker hændelse. Hvis A er den tomme mængde, så siger vi at A er en umulig hændelse.

Sandsynlighedsfunktioner

Lad U være et udfaldsrum bestående af udfald u_1, u_2, \dots, u_n . En sandsynlighedsfunktion er en funktion P som tager en hændelse A som input og returnerer et tal $P(A)$ som er sandsynligheden for at hændelsen A indtræffer. Sandsynligheden $P(A)$ skal opfylde følgende betingelser: * $P(A) \geq 0$ for alle hændelser A . * $P(U) = 1$.

Udfaldsrummet sammen med sandsynlighedsfunktionen kaldes et sandsynlighedsrum eller et sandsynlighedsfelt

Disjunkte hændelser

Sætning: To hændelser A og B er disjunkte hvis $A \cap B = \emptyset$. Det betyder at A og B ikke har nogen fælles udfald. Hvis A og B er disjunkte, så er sandsynligheden for at enten A eller B indtræffer givet ved:

$$P(A \cup B) = P(A) + P(B)$$

Bevis: Lad A og B være disjunkte. Så er $A \cap B = \emptyset$. Derfor er $A \cup B = A \cup B \cup \emptyset = A \cup B \cup (A \cap B) = (A \cup B) \cap (A \cup B) = A \cup B$. Derfor er $P(A \cup B) = P(A \cup B) = P(A \cup B) + P(\emptyset) = P(A \cup B) + P(A \cap B) = P(A) + P(B)$.

Disjunkte hændelser i Python

Vi kan teste om to lister er disjunkte i numpy ved hjælp af `intersection`:

```
A = [1,2,3]
```

```
B = [4,5,6]
```

```
print(A.intersection(B) == set())
```

Komplementære hændelser

Definition: Lad A være en hændelse. Så er den komplementære hændelse \bar{A} givet ved:

$$\bar{A} = U \setminus A$$

Det betyder at \bar{A} er alle de udfald som ikke er i A .

Komplementære hændelser i Python

Vi kan finde den komplementære hændelse til en liste ved hjælp af `difference`:

```
A = [1,2,3]
```

```
U = [1,2,3,4,5,6]
```

```
print(U.difference(A))
```

Komplementære hændelser og sandsynlighedsfunktioner

Sætning: Lad A være en hændelse. Så er sandsynligheden for at A ikke indtræffer givet ved:

$$P(\bar{A}) = 1 - P(A)$$

Vi kan bevise sætningen ved at bruge at A og \bar{A} er disjunkte og at $A \cup \bar{A} = U$.

Da gælder at $P(U) = P(A \cup \bar{A}) = P(A) + P(\bar{A})$. Derfor er $P(\bar{A}) = P(U) - P(A) = 1 - P(A)$.

Sandsynlighedsfunktioner i Python

Vi kan repræsentere en sandsynlighedsfunktion som en dictionary:

```
P = {1: 1/6, 2: 1/6, 3: 1/6, 4: 1/6, 5: 1/6, 6: 1/6}
```

Vi kan finde sandsynligheden for en hændelse ved at summere sandsynlighederne for de udfald som er i hændelsen:

```
A = [1,2,3]  
print(sum([P[u] for u in A]))
```

Uafhængige hændelser

Definition: Lad A og B være hændelser. Så er A og B uafhængige hvis:

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

Det betyder at sandsynligheden for at A og B indtræffer er lig med sandsynligheden for at A indtræffer gange sandsynligheden for at B indtræffer.

Eksempler på uafhængige og ikke uafhængige hændelser

- ▶ Hvis vi kaster to mønter A og B, så er hændelsen at A viser plat og hændelsen at B viser krone uafhængige. De er uafhængige fordi ene mønt ikke påvirker den anden mønt.
- ▶ Hvis vi kaster to mønter A og B, så er hændelsen at A viser plat og hændelsen at summen af øjnene er 2 ikke uafhængige. De er ikke uafhængige fordi hvis A viser plat, så er summen af øjnene mindst 2.
- ▶ Hvis vi kaster to terninger A og B, så er hændelsen at A viser 1 og hændelsen at B viser 2 uafhængige. De er uafhængige fordi ene terning ikke påvirker den anden terning.

Eksempel på uafhængige og ikke uafhængige hændelser

- ▶ Hvis vi kaster to terninger A og B, så er hændelsen at A viser 1 og hændelsen at summen af øjnene er 2 ikke uafhængige. De er ikke uafhængige fordi hvis A viser 1, så er summen af øjnene mindst 2.
- ▶ Hvis vi kaster to terninger A og B, så er hændelsen at A viser 1 og hændelsen at summen af øjnene er 7 uafhængige. De er uafhængige fordi ene terning ikke påvirker den anden terning.
- ▶ Hvis vi kaster to terninger A og B, så er hændelsen at A viser 1 og hændelsen at summen af øjnene er 12 ikke uafhængige. De er ikke uafhængige fordi hvis A viser 1, så er summen af øjnene mindst 2.

Uafhængige hændelser i Python

Vi kan teste om to hændelser er uafhængige ved at teste om $P(A \cap B) = P(A) \cdot P(B)$:

```
A = [1,2,3]
B = [4,5,6]
for u in A.intersection(B):
    print(P[u] == P[u] * P[u])
```

Sandsynligheden for uafhængige hændelser

Sætning: Lad A og B være uafhængige hændelser. Så er sandsynligheden for at begge hændelser indtræffer givet ved:

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

Det kan vi bevise ved at bruge at A og \bar{A} er disjunkte og at $A \cup \bar{A} = U$. Da gælder at $P(U) = P(A \cup \bar{A}) = P(A) + P(\bar{A})$ idet, at A og \bar{A} er disjunkte

Derfor er $P(\bar{A}) = P(U) - P(A) = 1 - P(A)$. Dvs. at $P(A) = 1 - P(\bar{A})$. Vi kan bruge det til at vise at $P(A \cap B) = P(A) \cdot P(B)$:

$$P(A \cap B) = P(A) \cdot P(B) = (1 - P(\bar{A})) \cdot P(B) = P(B) - P(\bar{A}) \cdot P(B) = P(B) -$$

Derfor er $P(A \cap B) = P(A) \cdot P(B)$.

Betinget sandsynlighed

Definition: Lad A og B være hændelser. Så er den betingede sandsynlighed for at A indtræffer givet at B indtræffer givet ved:

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

Bemærk at det antages, at $P(B)$ ikke er lig med 0.

Det læses som at sandsynligheden for at A indtræffer givet at B indtræffer er lig med sandsynligheden for at A og B indtræffer delt med sandsynligheden for at B indtræffer.

Eksempler på betinget sandsynlighed

- ▶ Hvis vi kaster to mønter A og B, så er sandsynligheden for at A viser plat givet at B viser krone givet ved $P(A|B) = \frac{1}{2}$.
- ▶ Hvis vi kaster to mønter A og B, så er sandsynligheden for at A viser plat givet at summen af øjnene er 2 givet ved $P(A|B) = \frac{1}{2}$.
- ▶ Hvis vi kaster to terninger A og B, så er sandsynligheden for at A viser 1 givet at B viser 2 givet ved $P(A|B) = \frac{1}{6}$.
- ▶ Hvis vi kaster to terninger A og B, så er sandsynligheden for at A viser 1 givet at summen af øjnene er 2 givet ved $P(A|B) = \frac{1}{1} = 1$.
- ▶ Hvis vi kaster to terninger A og B, så er sandsynligheden for at A viser 1 givet at summen af øjnene er 7 givet ved $P(A|B) = \frac{1}{6}$.

Flere eksempler på betinget sandsynlighed

- ▶ Hvis vi kaster to terninger A og B, så er sandsynligheden for at A viser 1 givet at summen af øjnene er 12 givet ved $P(A|B) = \frac{1}{1} = 1$.
- ▶ Hvis vi kaster to terninger A og B, så er sandsynligheden for at A viser 1 givet at summen af øjnene er 13 givet ved $P(A|B) = \frac{0}{1} = 0$.

Betinget sandsynlighed i Python

Vi kan udregne den betingede sandsynlighed for at A indtræffer givet at B indtræffer ved at bruge at $P(A|B) = \frac{P(A \cap B)}{P(B)}$:

```
A = random.sample(range(1, 100), 10)
B = random.sample(range(1, 100), 10)
```

Ved brug af numpy kan vi skrive det mere kompakt:

```
for u in A.intersection(B):
    for b in B:
        if (P[b] != 0):
            print(P[u] / P[b])
```


Betinget sandsynlighed og uafhængige hændelser

Sætning: Lad A og B være uafhængige hændelser. Så er den betingede sandsynlighed for at A indtræffer givet at B indtræffer givet ved:

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

Det kan tolkes som at hvis A og B er uafhængige, så påvirker B ikke sandsynligheden for at A indtræffer.

Vi kan bevise sætningen ved på følgende måde:

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

Første lighed følger af definitionen af betinget sandsynlighed. Anden lighed følger af at A og B er uafhængige. Dvs. at $P(A \cap B) = P(A) \cdot P(B)$. Tredje lighed følger af at $P(B) \neq 0$.

Bayes regel

Sætning: Lad A og B være hændelser. Så er:

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

Bemærk: Vi antager ikke at A og B er uafhængige! Denne sætning er meget vigtig i maskinlæring og kaldes Bayes regel. Opkaldt efter Thomas Bayes.

Tolking af Bayes regel

Vi kan tolke Bayes regel på følgende måde:

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

Hvis A og B er uafhængige, så er $P(A|B) = P(A)$. Hvis A og B ikke er uafhængige, så er $P(A|B) \neq P(A)$. Bayes regel fortæller os hvor meget sandsynligheden for at A indtræffer ændres hvis B indtræffer:

$$\frac{P(B|A) \cdot P(A)}{P(B)}$$

Bevis for Bayes regel

Vi kan bevise sætningen ved at bruge at $P(A|B) = \frac{P(A \cap B)}{P(B)}$ og at $P(B|A) = \frac{P(A \cap B)}{P(A)}$:

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

Bayes regel i numpy

Vi kan udregne den betingede sandsynlighed for at A indtræffer givet at B indtræffer ved at bruge Bayes regel:

```
A = [1,2,3]
B = [4,5,6]
for u in A.intersection(B):
    for b in B:
        if(P[b] != 0):
            print(P[u] * P[b] / P[b])
```

Bayes regel med flere betingelser

Vi kan bruge Bayes regel til at udregne betingede sandsynligheder med flere betingelser. Lad A og B være hændelser. Så er:

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

Betragt nu en hændelse C . Så er:

$$P(A|B, C) = \frac{P(B, C|A) \cdot P(A)}{P(B, C)}$$

Vi skriver komma mellem B og C hvilket betyder at B og C begge skal indtræffe. Det svarer til at vi skriver $B \cap C$.

Lighedstegnet følger af Bayes regel. Vi kan bevise det ved at bruge at $P(B, C|A) = \frac{P(A \cap B \cap C)}{P(A)}$ og at $P(B, C) = \frac{P(B \cap C)}{P(A)}$:

$$P(A|B, C) = \frac{P(B, C|A) \cdot P(A)}{P(B, C)} = \frac{\frac{P(A \cap B \cap C)}{P(A)} \cdot P(A)}{\frac{P(B \cap C)}{P(A)}} = \frac{P(A \cap B \cap C)}{P(B \cap C)}$$

Bayes regel med n betingelser

Antag at vi har n hændelser B_1, B_2, \dots, B_n . Så er:

$$P(A|B_1, B_2, \dots, B_n) = \frac{P(B_1, B_2, \dots, B_n|A) \cdot P(A)}{P(B_1, B_2, \dots, B_n)}$$

Vi skriver komma mellem B_1, B_2, \dots, B_n hvilket betyder at B_1, B_2, \dots, B_n alle skal indtræffe. Det svarer til at vi skriver $B_1 \cap B_2 \cap \dots \cap B_n$.

Lighedstegnet følger af Bayes regel. Vi kan bevise det ved at bruge at $P(B_1, B_2, \dots, B_n|A) = \frac{P(A \cap B_1 \cap B_2 \cap \dots \cap B_n)}{P(A)}$ og at

$$P(B_1, B_2, \dots, B_n) = \frac{P(B_1 \cap B_2 \cap \dots \cap B_n)}{P(A)}.$$

$$P(A|B_1, B_2, \dots, B_n) = \frac{P(B_1, B_2, \dots, B_n|A) \cdot P(A)}{P(B_1, B_2, \dots, B_n)} =$$

$$\frac{\frac{P(A \cap B_1 \cap B_2 \cap \dots \cap B_n)}{P(A)} \cdot P(A)}{\frac{P(B_1 \cap B_2 \cap \dots \cap B_n)}{P(A)}} = \frac{P(A \cap B_1 \cap B_2 \cap \dots \cap B_n)}{P(B_1 \cap B_2 \cap \dots \cap B_n)}$$

Bayes regel med n uafhængige betingelser

Antag at vi har n uafhængige hændelser B_1, B_2, \dots, B_n . Så er:

$$P(A|B_1, B_2, \dots, B_n) = \frac{P(B_1, B_2, \dots, B_n|A) \cdot P(A)}{P(B_1, B_2, \dots, B_n)}$$

Vi kan bruge at

$P(B_1, B_2, \dots, B_n|A) = P(B_1|A) \cdot P(B_2|A) \cdot \dots \cdot P(B_n|A)$ og at
 $P(B_1, B_2, \dots, B_n) = P(B_1) \cdot P(B_2) \cdot \dots \cdot P(B_n)$:

$$P(A|B_1, B_2, \dots, B_n) = \frac{P(B_1, B_2, \dots, B_n|A) \cdot P(A)}{P(B_1, B_2, \dots, B_n)} =$$
$$\frac{P(B_1|A) \cdot P(B_2|A) \cdot \dots \cdot P(B_n|A) \cdot P(A)}{P(B_1) \cdot P(B_2) \cdot \dots \cdot P(B_n)}$$

Bayes regel i maskinelæring

Bayes regel er meget vigtig i maskinelæring. En enkelt og ret elegant læringsalgoritme kaldet, Naive Bayes, er baseret på Bayes regel. Vi skal se på Naive Bayes i det følgende.

Grundlæggende er Naive Bayes en klassifikationsalgoritme. Det betyder at den kan bruges til at klassificere data i klasser. Vi skal se på et eksempel.

Naive Bayes - et indledende eksempel

Naive Bayes kan bruges til at klassificere data i klasser ud fra data. Vi skal se på et eksempel.

Betragt følgende tænkte eksempel: Vi har en database med 10 personer. - For hver person kender vi kønnet, højden og vægten. - Vi vil gerne klassificere personerne i to klasser: mænd og kvinder. - Vi har 5 mænd og 5 kvinder.

Naive Bayes - et indledende eksempel - fortsat

Data kan opsummeres i følgende tabel:

Køn	Højde	Vægt
M	180	80
K	160	60
M	170	70
K	150	50
M	190	90
K	170	60
M	180	70
K	160	50
M	170	80
K	150	60

Naive Bayes - et indledende eksempel - fortsat

Vi kan nu bruge Bayes regel til at finde sandsynligheden for at en person er en mand givet højden og vægten. Vi kan bruge følgende notation: * M er hændelsen at en person er en mand. * K er hændelsen at en person er en kvinde. * H er hændelsen at en person har en bestemt højde (f.eks. 180). * V er hændelsen at en person har en bestemt vægt (f.eks. 80).

Naive Bayes - et indledende eksempel - fortsat

Da har vi ifølge Bayes regel:

$$P(M|H, V) = \frac{P(H, V|M) \cdot P(M)}{P(H, V)}$$

Kommentarer til ligningen: * $P(M|H, V)$ er sandsynligheden for at en person er en mand givet højden og vægten. * $P(H, V|M)$ er sandsynligheden for at en person har højden og vægten givet at personen er en mand. * $P(M)$ er sandsynligheden for at en person er en mand. * $P(H, V)$ er sandsynligheden for at en person har højden og vægten. * $P(H, V|M) \cdot P(M)$ er sandsynligheden for at en person har højden og vægten og er en mand.

Naive Bayes - et indledende eksempel - fortsat

Lad os antage at højden og vægten er uafhængige. Så er:

$$P(H, V|M) = P(H|M) \cdot P(V|M)$$

Dette følger af at H og V er uafhængige givet M . Ligningen bliver nu:

$$P(M|H, V) = \frac{P(H|M) \cdot P(V|M) \cdot P(M)}{P(H, V)}$$

Naive Bayes - et indledende eksempel - fortsat

Lad os benytte Naive Bayes til at klassificere en person med højde 180 og vægt 80. Vi skal finde sandsynligheden for at personen er en mand og sandsynligheden for at personen er en kvinde. Vi skal bruge følgende notation: * M er hændelsen at en person er en mand. * K er hændelsen at en person er en kvinde. * H er hændelsen at en person har højden 180. * V er hændelsen at en person har vægten 80. * $H \cap V$ er hændelsen at en person har højden 180 og vægten 80. * $H \cup V$ er hændelsen at en person har højden 180 eller vægten 80. * $H \cap V \cap M$ er hændelsen at en person har højden 180 og vægten 80 og er en mand.

Naive Bayes - et indledende eksempel - fortsat

Ved brug af Naive Bayes får vi:

$$P(M|H, V) = \frac{P(H|M) \cdot P(V|M) \cdot P(M)}{P(H, V)}$$

I de følgende vil vi finde de forskellige sandsynligheder i ligningen.

Naive Bayes - et indledende eksempel - fortsat

Vi kan finde sandsynligheden for at en person har højden 180 givet at personen er en mand ved at tælle antallet af mænd med højden 180 og dividere med antallet af mænd:

$$P(H|M) = \frac{\text{antal } m_nd \text{ med } h\emptyset jden \ 180}{\text{antal } m_nd}$$

Vi kan finde sandsynligheden for at en person har vægten 80 givet at personen er en mand ved at tælle antallet af mænd med vægten 80 og dividere med antallet af mænd:

$$P(V|M) = \frac{\text{antal } m_nd \text{ med } v_gten \ 80}{\text{antal } m_nd}$$

Naive Bayes - et indledende eksempel - fortsat

Vi kan finde sandsynligheden for at en person er en mand ved at tælle antallet af mænd og dividere med antallet af personer:

$$P(M) = \frac{\text{antal } m_nd}{\text{antal personer}}$$

Vi kan finde sandsynligheden for at en person har højden 180 og vægten 80 ved at tælle antallet af personer med højden 180 og vægten 80 og dividere med antallet af personer:

$$P(H, V) = \frac{\text{antal personer med } h\emptyset jden\ 180\ \text{og } v_gten\ 80}{\text{antal personer}}$$

Naive Bayes - et indledende eksempel - fortsat

Ved indsættelse i Naive Bayes får vi:

$$P(M|H, V) = \frac{P(H|M) \cdot P(V|M) \cdot P(M)}{P(H, V)}$$

Vi kan nu finde sandsynligheden for at personen er en mand givet højden og vægten:

$$P(M|H, V) = \frac{\frac{\text{antal m_nd med h\o jden 180}}{\text{antal m_nd}} \cdot \frac{\text{antal m_nd med v_gten 80}}{\text{antal m_nd}} \cdot \frac{\text{antal m_nd}}{\text{antal personer}}}{\frac{\text{antal personer med h\o jden 180 og v_gten 80}}{\text{antal personer}}}$$

Naive Bayes - et indledende eksempel - fortsat

Dvs. sandsynligheden for at personen er en mand givet højden og vægten er i følgende tilfælde:

$$P(M|H, V) = \frac{\frac{2}{5} \cdot \frac{3}{5} \cdot \frac{5}{10}}{\frac{2}{10}} = \frac{\frac{6}{50}}{\frac{2}{10}} = \frac{6}{50} \cdot \frac{10}{2} = \frac{60}{100} = 0.6$$

Naive Bayes - et indledende eksempel - fortsat

Herunder tabel over vægtene, højderne samt sandsynlighederne for hver række i datasættet:

Køn	Højde	Vægt	$P(H M)$	$P(V M)$	$P(M)$	$P(H, V)$	$P(M H, V)$
M	180	80	0.6	0.6	0.5	0.2	0.75
K	160	60	0.4	0.4	0.5	0.2	0.25
M	170	70	0.5	0.5	0.5	0.2	0.5
K	150	50	0.3	0.3	0.5	0.2	0.5
M	190	90	0.7	0.7	0.5	0.1	0.875
K	170	60	0.5	0.4	0.5	0.1	0.625
M	180	70	0.6	0.5	0.5	0.1	0.75
K	160	50	0.4	0.3	0.5	0.1	0.375
M	170	80	0.5	0.6	0.5	0.1	0.625
K	150	60	0.3	0.4	0.5	0.1	0.375

Naive Bayes i det generelle tilfælde

Betragt et datasæt med n rækker, hvor hver række indeholder m attributter og k klasser. På matrixform kan datasættet skrives som en $n \times m$ matrix X og en $n \times 1$ vektor y :

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Her kaldes X for attributmatricen og y for klasserne.

Naive Bayes i det generelle tilfælde - fortsat

Vi kan nu finde sandsynligheden for at en række x har klassen c givet ved:

$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)}$$

Udskrevet får vi:

$$P(c|x) = \frac{P(x_1, x_2, \dots, x_m|c) \cdot P(c)}{P(x_1, x_2, \dots, x_m)}$$

Det giver os følgende:

$$P(c|x) = \frac{P(x_1|c) \cdot P(x_2|c) \cdot \dots \cdot P(x_m|c) \cdot P(c)}{P(x_1, x_2, \dots, x_m)}$$

Her er $P(x_1|c)$ sandsynligheden for at attribut x_1 har værdien x_{11} givet at klassen er c . Vi har udnyttet at attributterne er uafhængige givet klassen.

En algoritme til Naive Bayes

Vi kan nu skrive en algoritme til Naive Bayes. Algoritmen er baseret på Bayes regel. Algoritmen er som følger:

1. Find sandsynligheden for hver klasse.
2. For hver klasse c :
 1. Find sandsynligheden for hver attribut givet klassen.
 2. Find sandsynligheden for hver attribut givet rækken.
 3. Find sandsynligheden for rækken givet klassen.
 4. Find sandsynligheden for klassen givet rækken.
 5. Vælg den klasse med den største sandsynlighed.
 6. Tilføj klassen til listen over klasser.
 7. Tilføj sandsynligheden for klassen til listen over sandsynligheder.
 8. Tilføj klassen og sandsynligheden til listen over klasser og sandsynligheder.
 9. Gentag for hver klasse.
3. Returner klassen med den største sandsynlighed.
4. Eventuelt: Returner listen over klasser og sandsynligheder.

Naive Bayes i numpy

Herunder en simpel implementering af Naive Bayes i numpy:

```
def naive_bayes(X, y, x):  
    classes = set(y)  
    probabilities = []  
    for c in classes:  
        probabilities.append((c, probability_class(y, c)))  
    probabilities = sorted(probabilities, key=lambda x: x[1])  
    return probabilities[0][0]
```

Her er X attributmatricen, y klasserne og x rækken som skal klassificeres. Funktionen returnerer den klasse som x tilhører.

Eksempel på Naive Bayes i numpy

Herunder et eksempel på Naive Bayes i numpy:

```
X = np.array([[180, 80], [160, 60], [170, 70], [150, 50],  
y = np.array(['M', 'K', 'M', 'K', 'M', 'K', 'M', 'K', 'M',  
x = np.array([180, 80])  
print(naive_bayes(X, y, x))
```

Output:

M

Hvor er Naive Bayes god og hvor er den dårlig?

- ▶ Naive Bayes er en simpel algoritme.
- ▶ Den er god til at klassificere data i klasser, hvis attributterne er uafhængige.
- ▶ Den er dårlig til at finde sammenhænge i data, hvis attributterne er afhængige.

Gode cases for Naive Bayes

- ▶ Klassificering af tekster i kategorier. Det kan f.eks. være at klassificere en tekst i kategorierne: sport, politik, underholdning, etc. Det skyldes, at ordene i teksten er uafhængige.
- ▶ Klassificering af billeder i kategorier. Det kan f.eks. være at klassificere et billede i kategorierne: kat, hund, hest, etc. Det skyldes, at pixels i billedet er uafhængige.
- ▶ Klassificering af personer i kategorier. Det kan f.eks. være at klassificere en person i kategorierne: mand, kvinde, barn, etc. Det skyldes, at attributterne er uafhængige.