|  |
| --- |
| Projekt 1  Bestemmelse af alder  https://lh4.googleusercontent.com/3cRIjyg56_nFKY2h4I_wcCGQIetifSzHM2AmL0JzeF3pIWgiZyO6oP55vmit1i-SMpw3mt2zEC3kTYztJFB6SawrlE3NzjGIoQaUKUsrtCvtGbJPS-g0jW8Y4U_UbCNkXGr-A0lr |

Indholdsfortegnelse

[Teori om maskinlæring 3](#_Toc528082757)

[Hvad er maskinlæring? 3](#_Toc528082758)

[Hvad kan maskinlæring bruges til? 3](#_Toc528082759)

[Hvad er klassifikation? 3](#_Toc528082760)

[Decision stomp 4](#_Toc528082761)

[Nearest neighbour 5](#_Toc528082762)

[Hvad er regression? 6](#_Toc528082763)

[Lineær regression 6](#_Toc528082764)

[Polynomial regression 6](#_Toc528082765)

[Neurale netværk 7](#_Toc528082766)

[Projekt 8](#_Toc528082767)

[Idebeskrivelse 8](#_Toc528082768)

[Teknisk problemstilling 9](#_Toc528082769)

[Eventuelle problemer 9](#_Toc528082770)

[Valg af algoritme 9](#_Toc528082771)

[Hvordan output skal opføre sig 9](#_Toc528082772)

[Blokdiagram 10](#_Toc528082773)

[Konklusion 10](#_Toc528082774)

# Teori om maskinlæring

## Hvad er maskinlæring?

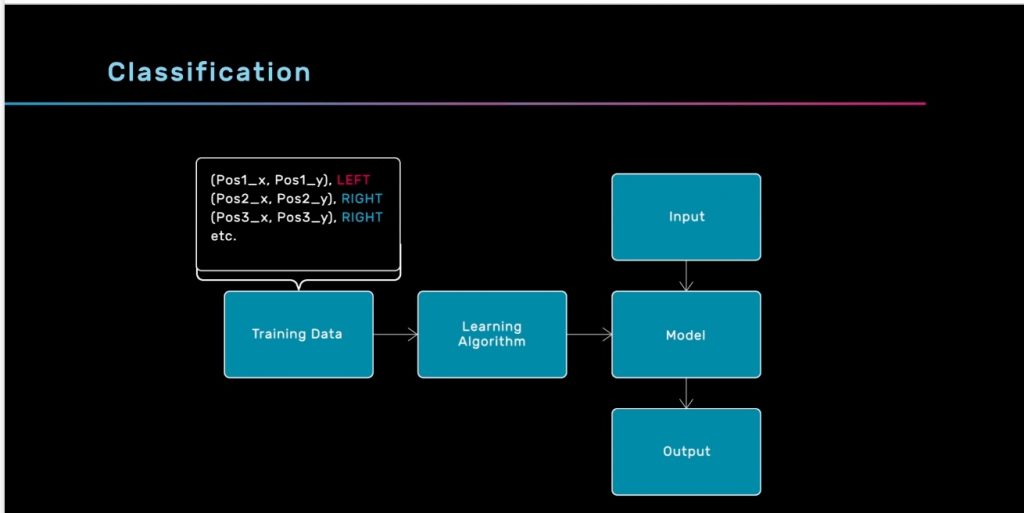
Maskinlæring er en type af digital modellering der i de seneste år har gjort indtog i mange forskellige produkter og services. Man kan groft sige at maskinlæring går ud på at bruge computere til at ”gætte” sig frem til reaktioner på baggrund af nogle kvalificerede træningsdata. I virkeligheden gætter computeren sig ikke frem, men regner derimod sandsynligheder ud for et specifikt spørgsmål eller scenarie, som derefter skal være i stand til at blive fortolket med et bestemt svar.

## Hvad kan maskinlæring bruges til?

I den moderne verden med en stigende datamængde er maskinlæring et nødvendigt til at forstå de interessante data i forbindelse med salg, markedsføring kommunikation, policy og sundhed. Maskinlæring er også et relevant værktøj at bruge, da os menneskers evne til at fortolke data på en logisk måde er begrænset.

## Hvad er klassifikation?

En klassifikation refererer til at computeren deler ting op i forskellige muligheder eller kategorier. Et eksempel kan være et webcam input der har tre forskellige ”kategorier/konfigurationer”, som bliver sammenlignet med hinanden. Man kan visuelt forestille sig hvordan en klassifikation virker på således.

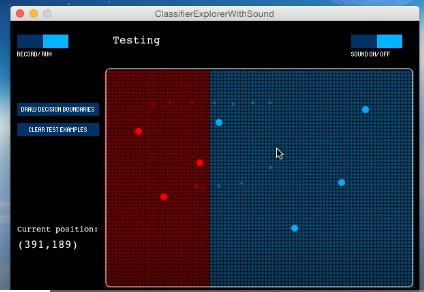


### Decision stomp

Decision stomp klassifikationen bygger på at computeren forsøger at lave en opdelingslinje mellem to forskellige klasser af input. Denne linje kan forekomme både lodret og vandret. Denne type klassifikation kan også beskrives med en form for kodesprog der siger: hvis x er mindre end et tal er det klasse 1, og ellers er det klasse 2. Man kan visuelt forestille sig måden hvorpå decision stomp klassifikationen fungere på således.

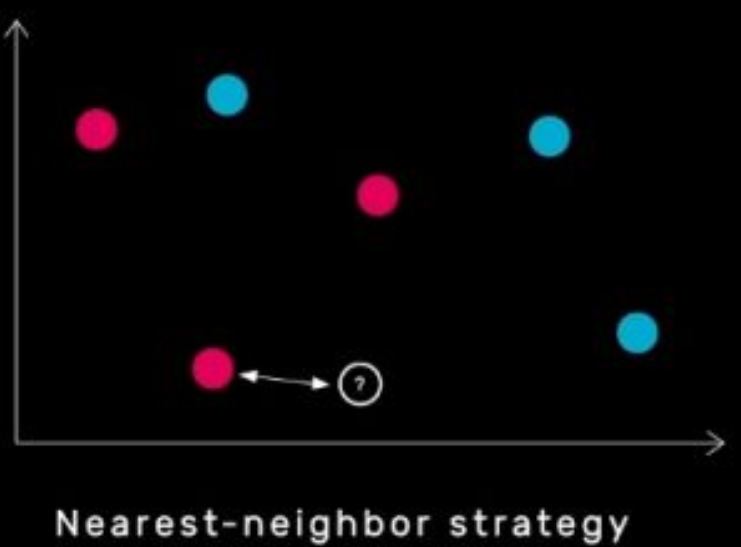


Hvis man træner på nogle punkter med Wekinator med brug af decision stomp klassifikationen, og derefter bruger et af de programmer som Rebecca Fiebrink har kodet, kan man se hvordan denne klassifikation virker i praksis. Det ser således ud.

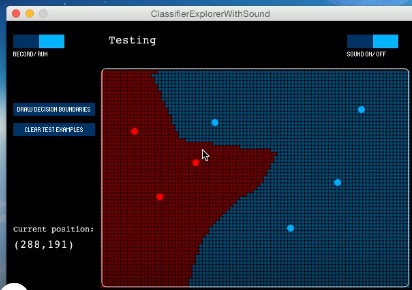


### Nearest neighbour

Nearest neighbour klassifikationen virker på den måde at den modtager nogle forskellige input, fx nogle x-y-værdier, som den får at vide, er klasse 1. Derefter modtager den nogle flere værdier, som den denne gang får at vide er klasse 2. Når der så skal klassificeres et vilkårligt punkt vil nearest neighbour klassifikationen, som det ligger i navnet, kigge på det nærmeste punkt, eller den nærmeste nabo, og derefter klassificere punktet baseret på det tætteste allerede eksisterende punkt. Man kan visuelt forestille sig måden hvorpå nearest neighbour klassifikationen fungere på således.



Hvis man træner på nogle punkter med Wekinator med brug af nearest neighbour klassifikationen, og derefter bruger et af de programmer som Rebecca Fiebrink har kodet, kan man se hvordan denne klassifikation virker i praksis. Det ser således ud.



## Hvad er regression?

Regression beregner et numerisk output, altså et faktisk tal, fra et input i modsætning til en klassifikation som beregner en klasse. Outputtet vil forandre sig når inputtet forandrer sig. Dette bliver gjort jævnt. Regression bliver brugt til at kontrollere forskellige parametre, såsom volumen, farve og position.

### Lineær regression

Når computeren laver en lineær regression, bygger det på den matematiske metode for at lave en lineær linje. En matematisk formel for en lineær linje lyder .

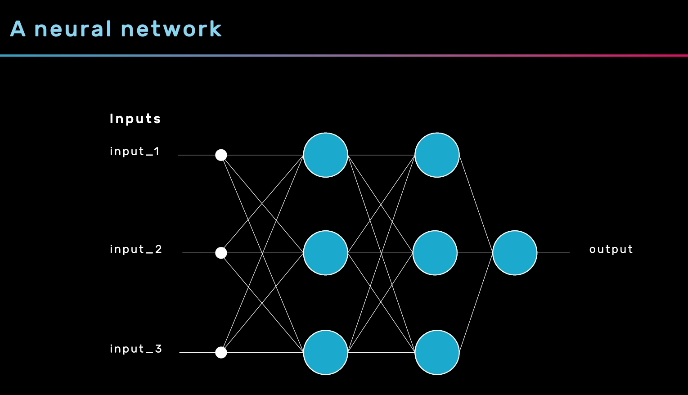
En lineær regression laver en lige linje mellem de punkter som man har trænet på. Dette kan være smart at bruge hvis man fx har med volumen at gøre, eller andre ting hvor der foregår en stabil og glidende overgang mellem to punkter.

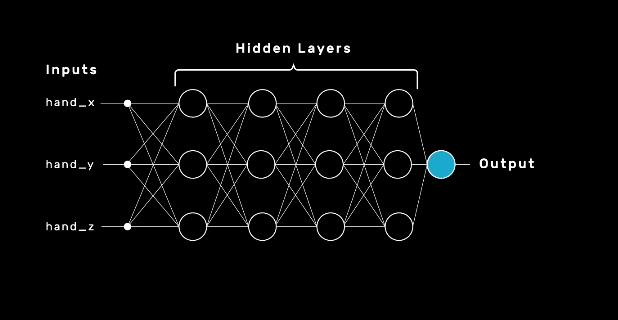
### Polynomial regression

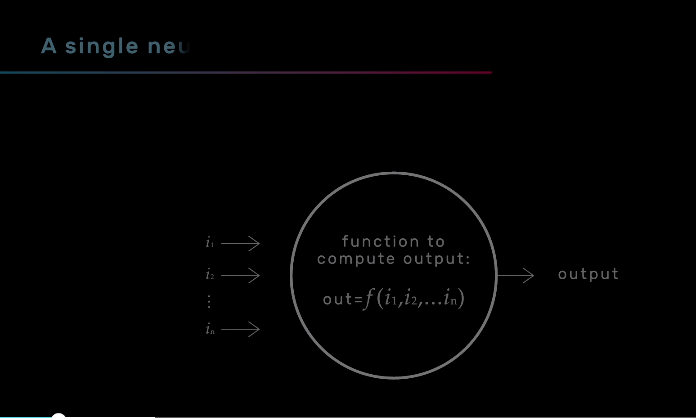
En polynomial regression vil skabe en kurve ud fra den matematiske formel for et polynomium, som lyder eller i en højere faktor. Denne regressionsmetode er god til at fange kurvede linjer, men derimod ikke til at skabe lige linjer.

### Neurale netværk

Et neuralt netværk virker på en meget anderledes måde en de forskellige andre regressionsmetoder. Et neuralt netværk danner ikke en enkelt funktion til at beskrive et output, men bruger i stedet mange forskellige formler i et netværk til at beregne et output. De følgende tre billeder beskriver visuelt processen der foregår i et neuralt netværk.



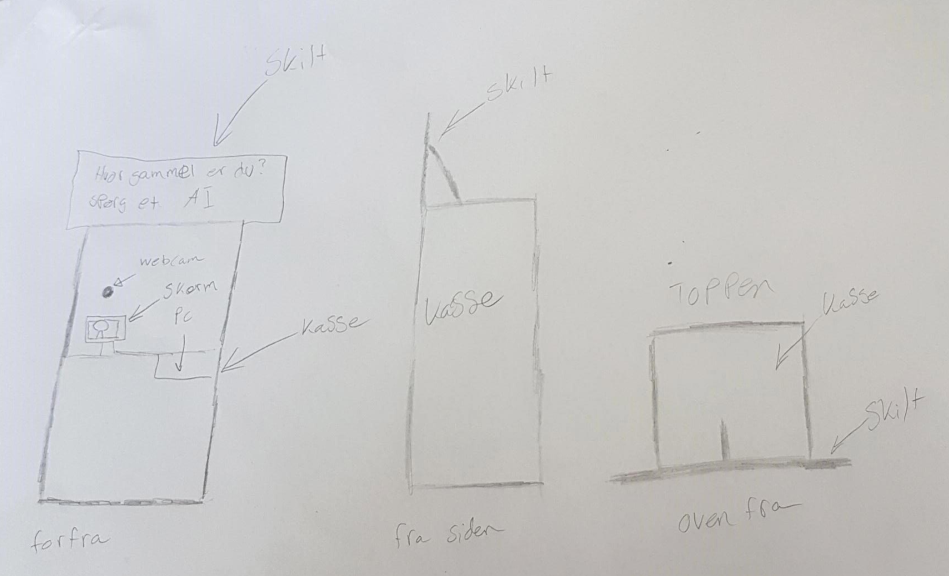




# Projekt

## Idebeskrivelse

Vores ide går ud på at lave et program der skal være i stand til at analysere en person, og give et godt gæt på den pågældende persons alder. Da vi i starten udarbejdede vores ide, lavede vi også en skitse for at illustrere vores ide. Denne skitse kan ses her.



Her kan det ses at vi forestillede os at vores program skulle sættes op ved en form for bod, hvor personer kunne gå op og teste vores program og finde ud af hvor gammel computeren tror at man er. Boden skulle bestå af en plade hvor computerskærmen og webcammet står, og et skilt ovenover hvor der står “Hvor gammel er du? Spørg et AI”.

## Teknisk problemstilling

Vores input skal komme fra et webcam. Dette input skal gennem en Processing skitse der laver dataene om til OSC-pakker, som derefter kan sendes til Wekinator. I Wekinator trænes der i forvejen på nogle billeder af folk der har forskellige aldre. Vi har valgt at dele alder op i 7 forskellige aldersgrupper, som lyder.

18-25

26-35

36-45

46-60

61-75

75+

Når inputtet fra webcammet når Wekinator vil det blive analyseret ved brug af nearest neighbour algoritmen, for derefter at blive inddelt i en af aldersgrupperne. Når Wekinator er færdig med at analysere det data der er blevet sendt ind, bliver resultatet sendt videre til vores output.

Vores output består af en Processing skitse der har en boks hvor resultatet af den test som vi foretager, vil være på skærmen. Der vil altså stå den alder som programmet har skudt en til at være.

### Eventuelle problemer

I forbindelse med udviklingen af vores program blev vi stillet et spørgsmål omkring vores ide. Det lød - *Kan software identificere alderen på en person ud fra et visuelt input?*

Det mener vi godt at det kan, hvis det vel og mærke bliver lavet rigtigt. Der er meget der kan gå galt, og der er også en stor chance for at det ikke virker som det skal eller overhovedet ikke virker.

## Valg af algoritme

Vi har valgt at bruge nearest neighbour algoritmen, da der skal gættes på en alder ud fra to ansigtet der skal sammenlignes. Vi mener derfor at det ville give mest mening at bruge netop denne algoritme, da Wekinator vil være i stand til at sortere testpersonerne i forhold til hvilken aldersgruppe de ligner mest.

## Hvordan output skal opføre sig

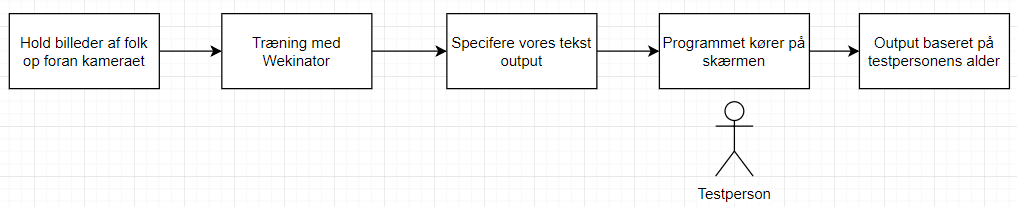
Vores output skal opføre sig på den måde at der skal vises en tekstboks hvor der står hvor gammel computeren tror at du er. Vores output skal være i konstant opdatering, da der hurtigt kunne komme en ny testperson og prøve vores program. Det er derfor vigtigt at det ikke er et statisk output i bruger, men snarere et dynamisk.

## Blokdiagram

Her vises et blokdiagram over vores ide. Man starter med at holde nogle billeder op foran kameraet for at træne på ved brug af Wekinator.

Grunden til at dette er nødvendigt er at vi var i tvivl omkring hvordan man fik billeder ind i Wekinator for at blive trænet på. Hvis vi havde haft viden omkring hvordan man gjorde det, ville vi have været i stand til at lave et bedre program.

Derefter skal vores output specificeres. Vores output består af en Processing skitse der indeholder noget tekst der beskriver hvor gammel computeren tror at testpersonen er. Den situation hvor testpersonen står foran computeren der kører vores program er også illustreret på blokdiagrammet.



# Konklusion

Vores projekt endte op med at fejle. Efter vores mening var det ikke fordi vi ikke tog os nok sammen eller ikke prøvede nok, det var fordi vi sigtede for højt, og hvad vi mente var muligt og relativt simpelt endte med at være et meget svært projekt som vi ikke kunne gennemføre med vores begrænsede viden om emnet. Det er derfor med stor ærgrelse at sige at vi ikke var i stand til at fremstille et reelt produkt, men blot en ide om et produkt.

Da vi startede vores brainstorm for at finde på en ide, havde vi store tanker om den ide vi kom op med. Vi tænkte at det ikke kunne være så svært, og vi havde endda tænkt over hvordan vi ville formode at lave produktet. Det endte dog desværre med at være et langt større projekt end hvad vi først havde regnet med, og da vi indså det var det allerede for sent at begynde med et nyt projekt. Vi valgte derfor at beskrive vores ide så godt som vi kunne, og prøve så godt vi kunne at så lavet noget der mindede om det vi originalt havde i tankerne.

Vi må dog indse at vi ikke fik formået at lave noget produkt der kunne bruge maskinlæring til at gætte folks alder. Vi fik ikke engang formået at lave et program der kunne tage billeder med et webcam. En del af grunden til dette er at vi begge havde langsomme computere, som gjorde at vores program frøs hver gang vi åbnede det. Så hvis vi havde haft nogle computere med lidt mere processorkraft, kunne vi måske havde fået et produkt færdigt. Vi ville i hvert fald have været i stand til at komme videre end det vi nåede.

Men hvis man skal se lidt positivt på det så har vi lært en masse om maskinlæring og om hvad der er muligt med den mængde viden og ekspertise vi har indenfor feltet. Vi vil derfor, i fremtiden, overveje kraftigt at skalere ned på sværhedsgraden af det vi sætter os for at lave, og starte småt. Vi har overestimeret vores evner, men den fejl laver vi ikke igen.