

TEKNOLOGIFORSTÅELSE

8. KLASSE SOM FAG

2. forløb

FRA TILFÆLDIGHED TIL MØNSTRE

- ALGORITMER OG DATA SOM BAGGRUND FOR FORUDSIGELSER

Udarbejdet af Malte von Sehested, Brian Ravnborg Christensen, Mikkel Hjort og Malene Erkmann*

*Materialet er udviklet af Københavns Professionshøjskole, Professionshøjskolen UCN, VIA University College samt læremiddel.dk for Børne- og Undervisningsministeriet under rammerne for Forsøg med teknologiforståelse i folkeskolens obligatoriske undervisning. Læs mere om forsøget på www.tekforsøget.dk og www.emu.dk.



KØBENHAVNS
PROFESSIONS
HØJSKOLE



LÆRE
MIDDEL
DK



VIA University
College

ucn

RAMBOLL

INDHOLDSFORTEGNELSE

| | |
|---|-----------|
| 1. Forløbsbeskrivelse | 3 |
| 1.1 Overordnet beskrivelse – tre sammenhængende forløb..... | 3 |
| 1.2 Resume..... | 5 |
| 1.3 Rammer og praktiske forhold..... | 5 |
| 2. Mål og faglige begreber..... | 5 |
| 3. Forløbsnær del..... | 7 |
| 3.1 Introfase: Forforståelse og kompetencer..... | 7 |
| 3.2 Udfordrings- og konstruktionsfase..... | 9 |
| 3.3 Outrofase: Ny forståelse og nye kompetencer..... | 14 |
| 4. Perspektivering..... | 15 |
| 4.1 Evaluering..... | 15 |
| 4.2 Progression..... | 15 |
| 4.3 Differentieringsmuligheder..... | 16 |
| 4.4 Særlige opmærksomhedspunkter..... | 16 |

Version 2

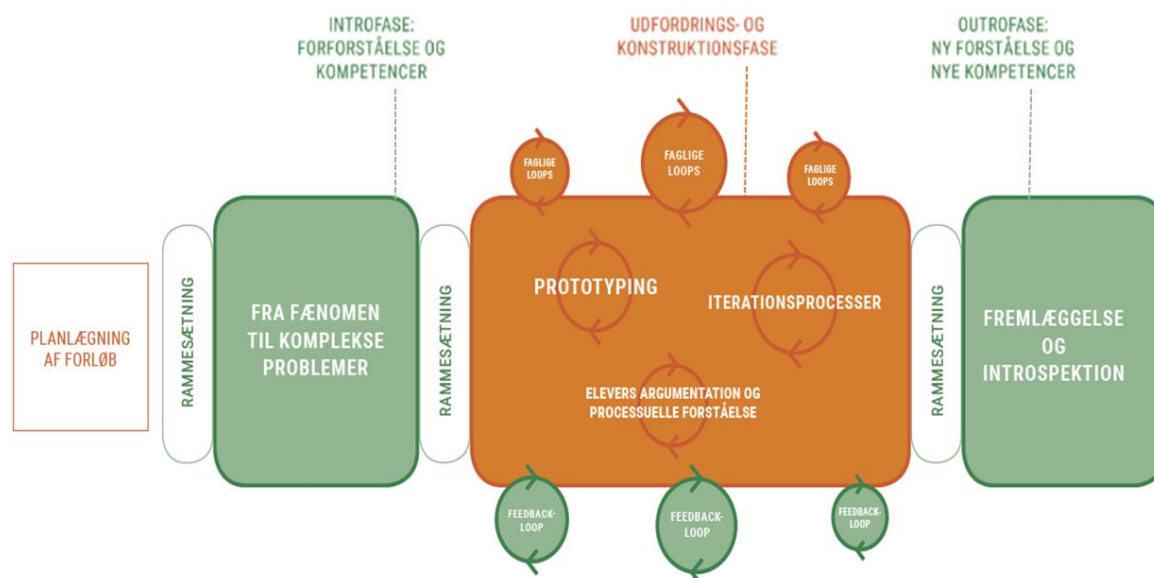
Dette er version 2 af forløbet. I revisionen af forløbene har vi arbejdet med at præcisere mål, rammer og aktiviteter. Der er ikke ændret fundamentalt ved forløbet, så materialer, som er udviklet til den konkrete undervisning på skolerne på baggrund af den første version af forløbet, vil stadig kunne anvendes.

1. Forløbsbeskrivelse

Dette forløb handler om de skjulte algoritmer, der styrer vores adfærd, og består af tre dele. Der tages udgangspunkt i simple spil som sten-saks-papir og gennem forløbet kommer eleverne til at arbejde med de data, der kan genereres i sådanne spil og med, hvordan dataene kan bruges til at modellere og forudsige fremtidige træk i spillet.

Forløbet er bygget op over det didaktiske format for prototyperne med en introducerende del, en mere undersøgende/eksperimenterende del og en outro-del med opsamlinger og evalueringer, se figur 1.

Figur 1: Forløbsmodel for prototyperne



1.1 Overordnet beskrivelse – tre sammenhængende forløb

Algoritmer og data benyttes i stigende grad af systemer der anvender forskellige grader af maskinlæring til, på baggrund af et stadig større datagrundlag, at forudsige sandsynligheder, muligheder og potentialer og i sidste ende til at foretage beslutninger. På et enkelt minut genereres uendelige mængder af data. Ifølge en opgørelse fra 2018 genereres fx 973.000 oplysninger om login på Facebook, der sendes 481.000 tweets og der bliver sendt 187 millioner e-mails, hvert eneste minut. Mange af disse data kan benyttes til at foretage forskellige forudsigelser og beslutninger.

En del af udfordringen i et digitaliseret samfund er at kunne forholde sig til hvordan smarte systemer og AI baseret på data og algoritmer foretager beregninger og beslutninger. Vi har brug for viden om, hvordan digitale systemer indsamler data om os, og hvordan vi selv kan indsamle og benytte data. Igennem tre

forløb om algoritmer og forudsigelser arbejder eleverne gennem aktiviteter der skal udvikle deres faglige kompetencer og give dem færdigheder og viden til at forstå muligheder og konsekvenser i teknologier omkring os, der på baggrund af data og statistik træffer beslutninger om og for os. Eleverne skal gennem forløbene arbejde med teknologifagets fagbegreber, og der lægges i arbejdet med forløbene vægt på, at der bevidst arbejdes med at udvikle et nyt fælles sprog i klassen.

Forløbene er planlagt i følgende rækkefølge:

Tabel 2: Forløbsoversigt

| TITEL OG VARIGHED | INDHOLD | KOMPETENCEMÅL |
|--|---|---|
| Tilfældigheder (8 lektioner) | I dette delforløb arbejder eleverne med at udvikle og spille sten-saks-papir. På baggrund af dataopsamling retter de herefter deres program, så det tager hensyn til fordelingen (sandsynligheden) for at der ikke er tale om et helt tilfældigt spil. Endelig perspektiveres i forhold til hvad det betyder, at man kender til sandsynligheden og hvordan det påvirker ens spil. | <p>Eleven kan reflektere over og anvende computationel tankegang på problemstillinger fra omverdenen.</p> <p>Eleven kan vurdere, vælge og på kvalificeret vis anvende digitale teknologier i autentiske situationer</p> <p>Eleven kan handle med dømmekraft i komplekse situationer, der vedrører digitale artefaktens betydning for individ, fællesskab og samfund</p> |
| Fra tilfældighed til mønstre (18 lektioner) | I dette delforløb arbejder eleverne videre med spillet og indsamler data om hvordan sandsynligheden ændrer sig, afhængig af det foregående spil. Med udgangspunkt i et færdigt program arbejder de med at forstå hvordan spillet forandrer og vejer sine sandsynligheder. Endelig perspektiveres i forhold til at tage udgangspunkt i større datasæt og brug af hele spillet som baggrund for vejningen af sandsynligheder. | <p>Eleven kan reflektere over og anvende computationel tankegang på problemstillinger fra omverdenen.</p> <p>Eleven kan vurdere, vælge og på kvalificeret vis anvende digitale teknologier i autentiske situationer</p> <p>Eleven kan handle med dømmekraft i komplekse situationer, der vedrører digitale artefaktens betydning for individ, fællesskab og samfund</p> |
| Udfordring (14 lektioner) | I tredje del skal eleverne bruge deres erfaringer med at indsamle data til at omforme et nyt spil til data, der kan danne baggrund for forudsigelser, som eleverne kan bruge til at vinde i spillet. | <p>Eleven kan reflektere over og anvende computationel tankegang på problemstillinger fra omverdenen.</p> <p>Eleven kan vurdere, vælge og på kvalificeret vis anvende digitale teknologier i autentiske situationer</p> |

Eleven kan handle med dømmekraft i komplekse situationer, der vedrører digitale artefaktens betydning for individ, fællesskab og samfund

1.2 Resume

I dette delforløb arbejder eleverne videre med spillet og indsamler data om, hvordan sandsynligheden ændrer sig, afhængig af det foregående spil. Med udgangspunkt i et færdigt program arbejder de med at forstå, hvordan spillet forandrer og vejer sine sandsynligheder. Endelig perspektiveres i forhold til at tage udgangspunkt i større datasæt og brug af hele spillet som baggrund for vejningen af sandsynligheder.

Produkt

Eleverne producerer en ny version af deres program der i stand til at spille ud fra det foregående spils resultat.

1.3 Rammer og praktiske forhold

Forløbet kræver ingen særlige lokaler eller andet udstyr end almindelige computere.

1.3.1 Samlet varighed

16 - 20 lektioner svarende til ca. 8 - 10 uger - afhængigt af brugen af faglige loops.

- Intro (4 lektioner)
- Dataindsamling og behandling samt opstilling af sandsynligheder (2 lektioner)
- Udvikling af program (6 lektioner)
- Perspektivering (4 lektioner)

1.3.2 Materialer

Der vil under forløbet være brug for følgende ressourcer:

- Computer med adgang til internettet
- Regneark til dataopsamling

2. Mål og faglige begreber

I denne del skal eleverne opnå en begyndende forståelse for, hvordan algoritmer kan bruges til at forudsige andres handlinger og til at vinde i spil. Eleverne skal gennem dataopsamling opnå en forståelse for, hvordan et spil kan omsættes til data for en computer, og de skal træne programmering af algoritmer. Endelig kan eleverne bruge deres algoritmer som udgangspunkt for at diskutere, hvordan sådanne algoritmer påvirker verden omkring dem - f.eks. ift. online gambling, aktiekurser osv.

KOMPETENCEOMRÅDER

DIGITAL
MYNDIGGØRELSE

COMPUTATIONEL
TANKEGANG

TEKNOLOGISK
HANDLEEVNE

| | | | |
|---|--|---|---|
| <p>Kompetencemål (efter 9. klassetrin)</p> | <p>Eleven kan handle med dømmekraft i komplekse situationer, der vedrører digitale artefakters betydning for individ, fællesskab og samfund</p> | <p>Eleven kan reflektere over og anvende computationel tankegang på problemstillinger fra omverdenen</p> | <p>Eleven kan vurdere, vælge og på kvalificeret vis anvende digitale teknologier i autentiske situationer</p> |
| <p>Færdigheds- og vidensmål (efter 9. klassetrin)</p> | <p>Eleven kan kritisk reflektere over digitale artefakters betydning for individ, fællesskaber og samfund. Eleven har viden om digitale artefakters betydning for individ, fællesskaber og samfund</p> | <p>Eleven kan behandle, vurdere og visualisere data reflekteret ved hjælp af digital teknologi. Eleven har viden om kriterier for datakvalitet</p> <p>Eleven kan vurdere forskellige algoritmers anvendelighed og kan benytte forskellige metoder til at afprøve algoritmer. Eleven har viden om forskellige parametre til vurdering af algoritmers anvendelighed</p> <p>Eleven kan konstruere digitale modeller af virkeligheden og ud fra dem lave forudsigelser og følgeslutninger og vurdere begrænsninger i modellen. Eleven har viden om, hvordan abstraktion af virkeligheden kan bruges til at beskrive og behandle denne i digitale modeller, og hvordan man kan afprøve en model ift. dens intentioner.</p> | <p>Eleven kan læse og forstå programmer skrevet i et tekstbaseret programmeringssprog samt anvende et sådant til systematisk modifikation og konstruktion af programmer ud fra en problem-specifikation. Eleven har viden om metoder til at analysere og forudsige programmets opførelse samt teknikker til systematisk og trinvis udvikling af programmer.</p> |

Konkretiserede læringsmål. Eleven kan...

- kan forstå, hvordan programmer indsamler data til machine learning
- indsamle og analysere data med henblik på at komme med forslag til redesign af et program
- kritisk reflektere over positive og negative konsekvenser ved anvendelse af "big data" og "machine learning" med udgangspunkt i et eller flere eksempler

3. Forløbsnær del

3.1 Introfase: Forforståelse og kompetencer

På baggrund af arbejdet med første delforløb har eleverne fået begyndende forståelser af sandsynligheder, og de har arbejdet med, hvordan disse sandsynligheder kunne bringes i spil. I dette delforløb kommer eleverne til at arbejde mere med betydningen af, hvad der lige er spillet i forhold til, hvad det næste træk bliver. Gennem dette arbejde opbygger eleverne en forståelse for, hvordan maskinlæring foregår i forhold til, at det handler om mønstre, data og sandsynligheder.

Der arbejdes med begreber som mønstre, betingelser og dataopsamling. Det anbefales at arbejdet begynder med en fælles snak i klassen om de centrale begreber og med at opbygge algoritmer til at brug for at spille sten-saks-papir baseret på de foregående træk.

3.1.1 Varighed

Estimeret til 4 lektioner á 45 minutter.

3.1.2 Problemfelt

Maskinlæring er en voksende del af vores samfund. Digitale robotter lærer sig at genkende ansigter ud fra fotos, at forstå menneskers tale og at vinde over os i komplekse spil som GO. I dette forløb undersøger eleverne måder, hvorpå systemer kan handle på baggrund af foregående handlinger ved at kigge efter mønstre.

3.1.3 Problemstilling

For at komme tættere på, hvordan maskiner lærer vores handlemønstre at kende, skal eleverne i dette forløb arbejde med følgende problemstilling: Hvordan kan vi fremstille et program, der i stand til at spille sten-saks-papir og foretage sine træk ud fra de foregående træk på baggrund af en databaseret undersøgelse?

3.1.4 Iscenesættelse/scenarie:

Da dette forløb bygger videre på det første forløb, så er følgende iscenesættelse fra det første forløb stadig relevant, så den kan med fordel genbesøges. I dette forløb fører iscenesættelse til at en opgave om at forstå programmer, som forsøger at gætte genstande på baggrund af simple input og en algoritme.

Computere er gode til at følge regler og rutiner. Ved at lave programmer kan man få computeren til at følge en række instruktioner. Så når man kan nedbryde en opgaveløsning til en algoritme, kan man også skrive et program der kan få en computer til at følge denne algoritme. De fleste af de programmer, vi benytter i dagligdagen, fungerer på denne måde, hvad enten det er til at skrive og sende e-mail, lave et opslag på Facebook eller se balancen på bankkontoen: Der er regler for, hvordan en e-mail sendes og

modtages, på Facebook er der et system til, hvordan input fra brugeren bliver til et opslag og udregningen af bankkontoens balance baserer sig på (relativt) simple beregninger.

Problemet er, at vi ikke altid ved, hvordan noget præcist skal gøres, og når vi ikke kender reglerne, er det meget sværere at udføre en opgave. Hvis vi ikke er helt sikre på, hvordan en række af regler fungerer, er det jo svært at fortælle en computer, hvordan den skal gøre det. Det kan også være at opgaveløsningen er baseret på usikkerhed. Når vi møder usikkerhed, prøver vi at lave det bedste gæt. Et gæt baseret på vores erfaringer og tanker om sandsynligheder for de forskellige udfald.

I forhold til et simpelt spil som sten-saks-papir kan vi naturligvis bruge den viden at fordelingen af træk ikke er lige. Der er flere der spiller sten end henholdsvis saks og papir. Umiddelbart betyder det, at den bedste strategi til at vinde er altid at spille papir. Problemet er bare, at hvis vores modstander har gennemskuet dette, er vi lette at slå. Derfor bliver vores bedste træk at basere det på det vi allerede ved om modstanderen, og på det eller de træk, der lige er lavet.

Når vi sætter computere til at lave samme type af opgaver, hvor vi ikke kan programmere dem til at følge en simpel instruktion, men i stedet skal have dem til at foretage et kvalificeret gæt, kan vi begynde at tale om maskinlæring som en løsning. Med maskinlæring forsøger vi at lære computeren selv at lære, hvordan den bedst løser opgaver baseret på usikkerhed. Når vi skal lære den at løse opgaver baseret på usikkerhed, tager vi fat i sandsynligheder.

Som en del af denne iscenesættelse kan man arbejde med enten Quickdraw eller Teachable Machine (eller begge dele) som eksempler på, at man først lærer computeren at skelne og genkende og ud fra sandsynligheder at vælge det bedste bud.

3.1.4.1 Quickdraw

Eleverne skal prøve at bruge Quickdraw til at tegne figurer. Quickdraw findes på <https://quickdraw.withgoogle.com/> Quickdraw er et program, der ved hjælp af maskinlæring er blevet god til at genkende, hvad det er, vi forsøger at tegne. I programmet skal man tegne en bestemt genstand og se om programmet kan genkende genstanden, som det man forsøger at tegne.

Lad eleverne udforske, hvad programmet kan gætte, og hvad det ikke kan gætte. Lad enten eleverne lave deres egne lister over, hvad programmet genkender, og især hvad den ikke genkender, eller lav disse lister i plenum.

Lad derefter eleverne forsøge at gennemskue, hvordan programmet indsamler data - måske hver gang, de bruger den?

- Hvordan kan Quickdraw vide, hvad jeg prøvede at tegne?

3.1.4.2 Teachable machine

I denne aktivitet skal eleverne arbejde med teachable machine, som er en gratis maskinlæringsapplikation fra Google. Med teachable machine kan eleverne selv prøve at hjælpe en maskine til at lære ud fra data. Eleverne skal bruge <https://teachablemachine.withgoogle.com/>.

Princippet er, at man fodrer maskinen med tre ting/personer, som den forsøger at finde mønstre og sammenhænge i, sådan at den efterfølgende kan genkende den samme ting/person i nye billeder. Ud fra de fænomener, man indlæser, genererer teachable machine altså data, som den lærer af og bruger til at lave forudsigelser/genkendelser. Lad f.eks. Eleverne arbejde med at lære maskinen at genkende tre ting fra klasseværelset ud fra lyd eller billeder.

3.1.5 Feedbackloop

Når eleverne har trænet maskinen, kan de reflektere over følgende spørgsmål vedrørende elevernes arbejde med sten-saks-papir samt maskinlæring som teachable machine og quickdraw:

- Hvad ved vi indtil nu om maskinlæring?
- Hvad adskiller et program, som spiller sten-saks-papir, fra de programmer eleverne lige har arbejdet med?

Her kunne man fx komme ind på forskellen i kompleksitet, når et program skal finde mønstre i data bestående af enkelte tal, og så andre typer af data såsom lyd og billeder. Et eksempel, hvor man udnytter denne kompleksitet, er, når man bruger identificering af symboler på billeder som en ekstra sikkerhed for, at det er en person, og ikke en computer, som skriver en adgangskode.

- Hvor kender du maskinlæring fra din hverdag?
- Hvor kunne du tænke dig, at det blev anvendt?
- Hvor kunne denne teknologi hjælpe/gøre noget bedre?

Dette kan gøres som skriftlige, individuelle opgaver, som gruppediskussioner og som dialog i hele klassen. Gennem elevernes evner til at pege på machine learning i deres hverdag, kan man evaluere deres forståelse, og gennem at se på elevernes evner til at finde på nye potentialer, kan man evaluere deres evner til at tænke kreativt med machine learning.

3.2 Udfordrings- og konstruktionsfase

I denne del arbejder eleverne med at programmere samt med at indsamle og behandle data. Det anbefales at eleverne arbejder i grupper.

3.2.1 Varighed

I alt estimeret til 8 lektioner a 45 minutter. Dertil kommer faglige loops, hvis behov afhænger meget af elevernes erfaringer.

3.2.2 Konkrete udfordringer

3.2.2.1 Dataindsamling og mønstre (2 lektioner)

Når vi ser på et simpelt spil som sten-saks-papir viser det sig hurtigt at måden det bliver spillet på slet ikke er så simpel endda. I foregående forløb arbejdede eleverne med at udvikle et program, der kunne spille alene på baggrund af statistikken og dermed sandsynlighederne overordnet set for, om der blev spillet mest sten, saks eller papir i deres datagrundlag. Denne gang vil vi lave en lidt større undersøgelse, hvor vi tager udgangspunkt i, hvad der bliver spillet på baggrund af, hvad der lige er blevet spillet.

Hvis der lige er blevet spillet sten mod saks, hvad er så sandsynligheden for, at der spilles sten i næste træk?

Til denne dataindsamling bliver vi nødt til at registrere i et større skema:

| | | | |
|---------------|--|--|--|
| Sten - Sten | | | |
| Sten - Saks | | | |
| Sten - Papir | | | |
| Saks - Sten | | | |
| Saks - Saks | | | |
| Saks - Papir | | | |
| Papir - Sten | | | |
| Papir - Saks | | | |
| Papir - Papir | | | |

Det anbefales at man arbejder 3 personer sammen (to spillere og en dataopsamler). Man kan eventuelt skifte lidt rundt undervejs. Har man allerede etableret grupper med for eksempel 4 kan man have en observatør der fortæller dataopsamleren hvad der blev spillet.

På baggrund af vores dataindsamling kan vi nu sætte fordelinger/sandsynligheder på den enkelte rækker:

| | | | |
|-------------|---|---|----|
| Sten - Sten | 9 | 5 | 11 |
|-------------|---|---|----|

giver for eksempel

| | | | |
|-------------|------------------|------------------|-------------------|
| Sten - Sten | 9 ud af 25 (36%) | 5 ud af 25 (20%) | 11 ud af 25 (44%) |
|-------------|------------------|------------------|-------------------|

Man kan overveje at lade gruppernes dataindsamling blive samlet til et større datasæt. Brug eventuelt muligheden for en klassediskussion. Hvad vil det betyde, hvis alle data bliver samlet til et datasæt i forhold til, at den enkelte gruppe har deres eget datasæt?

3.2.2.2 Redesign af program (6 lektioner)

I første forløbsdel arbejdede eleverne med at udvikle et program, der kunne spille sten-saks-papir baseret på ren tilfældighed og efterfølgende forbedre det. Nu skal de redesigne programmet, så det kan tage højde for den viden eleverne netop har skabt gennem deres dataindsamling.

De skal igennem to redesigns, hvor de i det første laver et design for et program, der alene tager udgangspunkt i den på forhånd indsamlede data og i det andet designer et program, der benyttes til at spille bedst af 10 og bruger data, der samles op undervejs til at basere sin gæt på.

Eleverne starter i begge opgaver med at diskutere opgaven og fremstille et flowchart over, hvordan programmet skal fungere. Dette fremlægger de for en opponentgruppe, og gennemgår det som et papirprogram, hvor de hele tiden følger med i flowchartet. På denne måde bliver deres flowchart udfordret, og de kan bruge denne viden til at lave en ny iteration.

Afhængig af elevernes programmeringsfærdigheder, kan man lade dem udvikle dele af eller hele programmet enten i blokkode eller i et tekstbaseret programmeringssprog, men det vigtigste er, at de får designet programmet som flowchart og diskuteret den computationelle logik, som programmet følger, i begge tilfælde.

Hvis eleverne kaster sig over programmeringsopgaven, kan det være en fordel at kende til datatypen Array til at gemme sandsynlighederne for de forskellige kombinationsmuligheder, da man ellers ender med at have ret mange variable at holde styr på.

3.2.2.2.1 Redesign ud fra på forhånd indsamlede data

I denne del af opgaven arbejdes med at designe et program, der benytter den data, de har indsamlet i den foregående aktivitet. Dette program vil som udgangspunkt ligne deres program fra foregående forløbsdel, men så snart der er foretaget et træk, vil programmet benytte dette træk sammen med den indsamlede data om spillede spil som grundlag for sit eget næste træk.

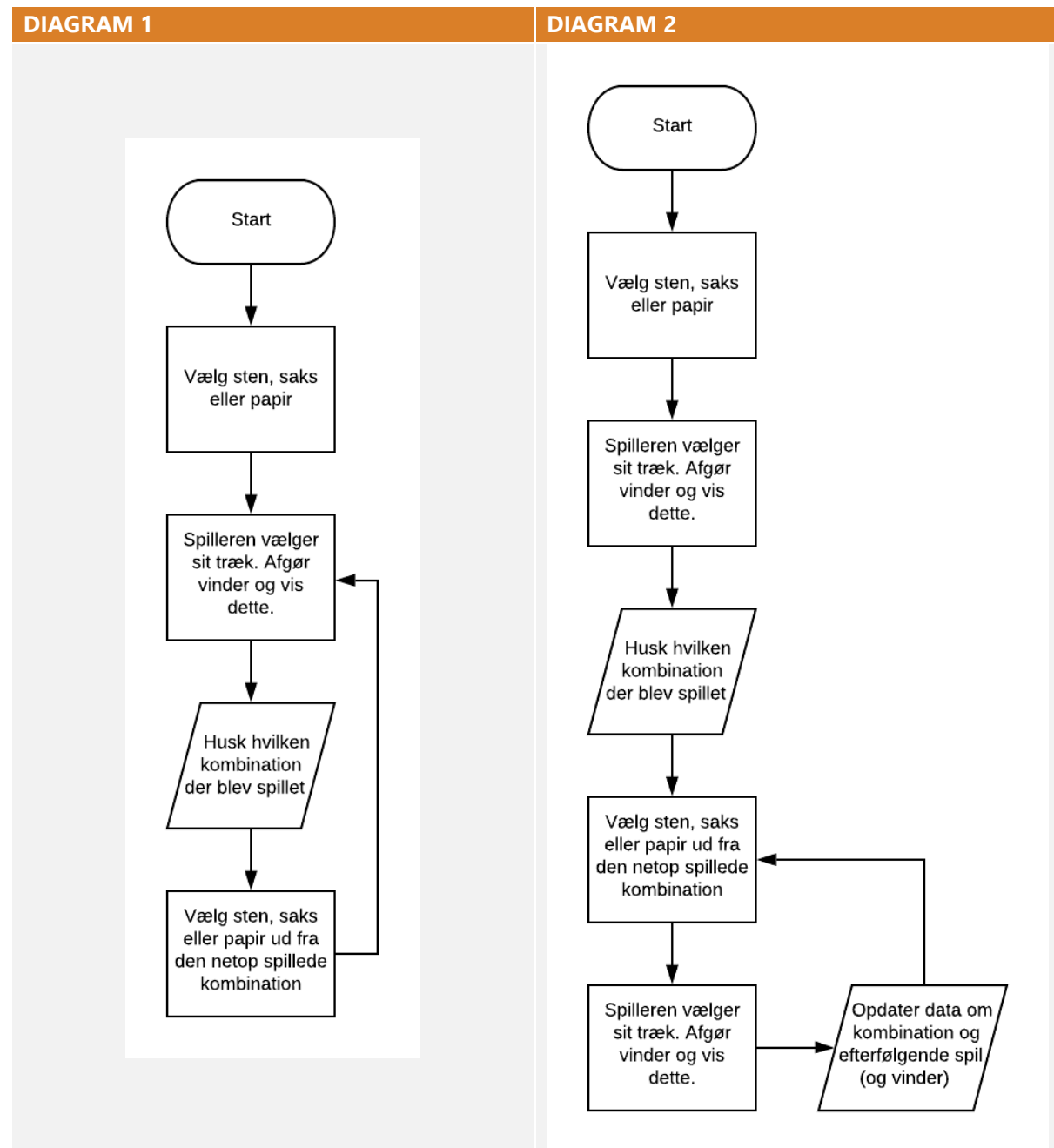
Et eksempel på et flowchart ses som diagram 1 (nedenfor).

3.2.2.2.2 Redesign ud fra data der skabes undervejs

I denne del af opgaven skal eleverne designe et program der i stedet for at benytte på forhånd indsamlet data selv skaber sin data efterhånden som der spilles. Det vil sige at det hele tiden samler data om de allerede spillede træk og bruger det som datagrundlag for sit næste træk.

Det vil altså sige at programmet hele tiden udbygger data om, hvilke kombinationer af et træk der giver hvilke sandsynligheder for det næste træk, på samme måde som eleverne gjorde manuelt i foregående opgave, og bruger denne data til at bestemme sit næste træk.

Et eksempel på et flowchart ses som diagram 2 (nedenfor).



3.2.3 Faglig loop

Elevernes arbejde med sten-saks-papir illustrerer dels dataindsamling og dels machine learning som et element i kunstig intelligens. Det nedenstående faglige loop giver eksempler på en app, der arbejder med hhv. store datamængder og machine learning. Dermed kan dette faglige loop anvendes til at perspektivere elevernes arbejde med sten-saks-papir. Det er oplagt at tale med eleverne om potentialerne i citizen's science for indsamling af big data og for træning af machine learning algoritmer. Pludselig har forskningen med hjælp af inddragelsen af nye digitale teknologier fået nogle muligheder, som var utænkelige for få år siden.

3.2.3.1 Brugen af data i AI

2 lektioner

iNaturalist: Kan hentes på App Store og Google Play. Se også <https://www.inaturalist.org/>.

I Danmark findes der mellem 40.000 og 50.000 forskellige arter af liv. Det er umuligt at lave en præcis optælling, da nogle arter lever i nogle meget begrænsede områder, og andre er ekstremt svære at bestemme. Med dataindsamling og AI har vi dog en chance for, at vi med tiden kan få et overblik, da helt almindelige borgere, uden nogen form for artskenndskab, kan bidrage med dataindsamlingen. iNaturalist er første skud på stammen. Med denne app kan man tage billeder af forskellige planter, dyr og svampe og derved hjælpe til med dataindsamlingen. Efterfølgende giver en AI et bud på, hvilken art der er på billedet.

App'en er endnu ikke fuldt oversat til dansk. Så ved nogle arter kan det være en fordel at oversætte artsnavnet ved at lave en Google søgning på det. Søgeresultatet vil ofte give det danske navn, når man søger fra Danmark. Google Translate er ikke brugbar til oversættelse af artsnavne.

Lige nu er app'en desuden inde i et machine learning forløb, hvor den dag for dag bliver bedre til at genkende de arter, som bliver fundet af brugerne. Man kan støde på arter som app'en endnu ikke kan genkende, og hvor et menneske skal ind og hjælpe. Her er der indbygget en forumfunktion i app'en, så eksperter kan hjælpe AI'en med artsgenkendelse.

Udover at app'en kan hjælpe brugeren med at bestemme forskellige dyr, planter og svampe, indsamler den også masser af data omkring livet på jorden, så man på sigt får en kæmpe database over udbredelsen af forskellige arter, samt deres hyppighed, næsten kun fantasien sætter grænser for, hvad man på sigt ville kunne søge frem i denne database.

Lad eleverne afprøve appen på forskellige dyr, planter og svampe omkring skolen. Man kan evt. udvide dette loop med en analyse og re-design af appen.

3.2.4 Feedbackloops

Der kan arbejdes med to korte feedbackloop i forbindelse med forløbet

3.2.4.1 Feedback efter databehandlingen

Når grupperne er færdige med databehandlingen, kan de diskutere deres resultater med en eller flere andre grupper eller diskussionen kan tages i klassen. En del af diskussionen kan omhandle muligheder og udfordringer ved at samle hele klassens data til et datasæt.

3.2.4.2 Feedback til algoritme og program

I løbet af arbejdet med algoritme og program kan der med fordel indlægges et par korte feedbackrunder, hvor eleverne giver og modtager feedback og på den måde fanger eventuelle uhensigtsmæssigheder eller bliver inspirerede af hinandens opgaveløsning.

3.3 Outrofase: Ny forståelse og nye kompetencer

3.3.1 Varighed

Estimeret 4 lektioner á 45 minutter.

3.3.2 Fremlæggelse og introspektion

Outrofase har fokus på at afprøve programmerne og reflektere over forskellige algoritmers anvendelighed og forbedringsmuligheder. Derudover bør de i fasen gives rum til at reflektere over, hvad det betyder at computeren selv løbende bliver klogere ved at lade nye data påvirke dens algoritme. Her kan man som afslutning på fasen bede eleverne tænke tilbage på deres første arbejde med sten-saks-papir, hvor deres algoritme var baseret på enkelte data.

Hvad betyder det for spillet, at algoritmen hele tiden udvikler sig?

Hvis algoritmen ikke udvikler sig, hvad betyder det så for måden man spiller på?

Nogle elever vil måske kunne trække på erfaringer fra computerspil, hvor du har gennemskuet algoritmen, og derfor nemt kan vinde spillene hver gang.

3.3.2.1 Den store sten-saks-papir turnering (2 lektioner)

Eleverne skal nu spille mod hinandens programmer i en turnering. Det anbefales, at man bruger en turneringsform hvor alle grupper (eller alle elever) prøver at spille mod alle programmer, og der ses både på antal sejre og på pointscore. Efterfølgende kan man diskutere, hvilket program der har klaret sig bedst og hvorfor.

Hvis man i 3.2.2.2 alene har designet programmerne og lavet flowcharts, kan man erstatte denne aktivitet med en fremlæggelse.

3.3.2.2 Mønstre og programmer der lærer (2 lektioner)

I det program, der er blevet arbejdet med, har man indsamlet data på forhånd og programmeret efter dette, men hvad med systemer der lærer undervejs?

Ved at have et program, der hele tiden bruger det aktuelle træk til at ændre på forståelsen af sandsynligheden af næste træk, er man på vej til at udvikle et program, der er sværere at lære at spille imod. Et eksempel på et program, der gør dette, kan man prøve her:

<http://www.rpscontest.com/human/5706404883070976>

Hvad sker der, hvis man spiller i faste mønstre? Hvilken strategi er god til at vinde over programmet?

I denne aktivitet kan det være en god ide at lade eleverne genbesøge deres flowchart fra 3.2.2.2 og vurdere, hvordan programmer, der hele tiden lærer og bruger både nuværende og tidligere spil som datagrundlag, kunne fungere.

Til denne del kan man også vælge at se programmet DR2 Tema: Hvad pokker er kunstig intelligens? (Kan for eksempel lånes via mitCFU: <http://hval.dk/mitcfu/materialeinfo.aspx?SearchID=29e24a11-36ef-4f8a-9c0f-9d6e74189a7f>)

4. Perspektivering

4.1 Evaluering

Hovedfokus i forløbet er elevernes viden og færdigheder ift. data og algoritmer. Læreren vurderer undervejs ifm. både skabelse og test af algoritmerne, i hvor høj grad eleverne er i stand til at udvælge og skabe relevante data for deres spil, samt at modellere disse spil vha. flowcharts og programmerede algoritmer. I forbindelse med outro-fasen vurderer læreren i særlig grad, i hvor høj grad eleverne er i stand til at forholde sig til algoritmens anvendelighed og forbedringsmuligheder, samt til at vurdere konsekvenser af at basere et spil på algoritmer fremfor menneskelige interaktion.

4.2 Progression

Eleverne bygger i dette forløb videre på det arbejde, de har lavet med programmering, data og algoritmer siden 1. klasse. I Del 2 bringes mange af elevernes opøvede kompetencer i spil. Del 2 bygger dermed videre på og forudsætter mange af de kompetencer og færdigheder, eleverne har erhvervet sig i hele arbejdet med teknologiforståelse.

4.3 Differentieringsmuligheder

Opgaven hvor eleverne skal udvikle deres eget program kan være vanskelig for nogle grupper. Her vil læreren kunne hjælpe ved i højere grad at støtte disse elevers udvikling samt ved at tilskynde eleverne til at vælge at lave simple programmer.

4.4 Særlige opmærksomhedspunkter

Dette forløb trækker på mange af de færdigheder og kompetencer, eleverne har opøvet igennem deres arbejde med teknologiforståelse, og derfor vil det for nogle elever være et vanskeligt forløb. Forløbet stiller store krav til lærerens evne til selv at se mulighederne for at anvende algoritmer til at vinde spil. Det er væsentligt at læreren lader eleverne prøve sig frem - også selv om resultatet ikke bliver lige godt hver gang.