

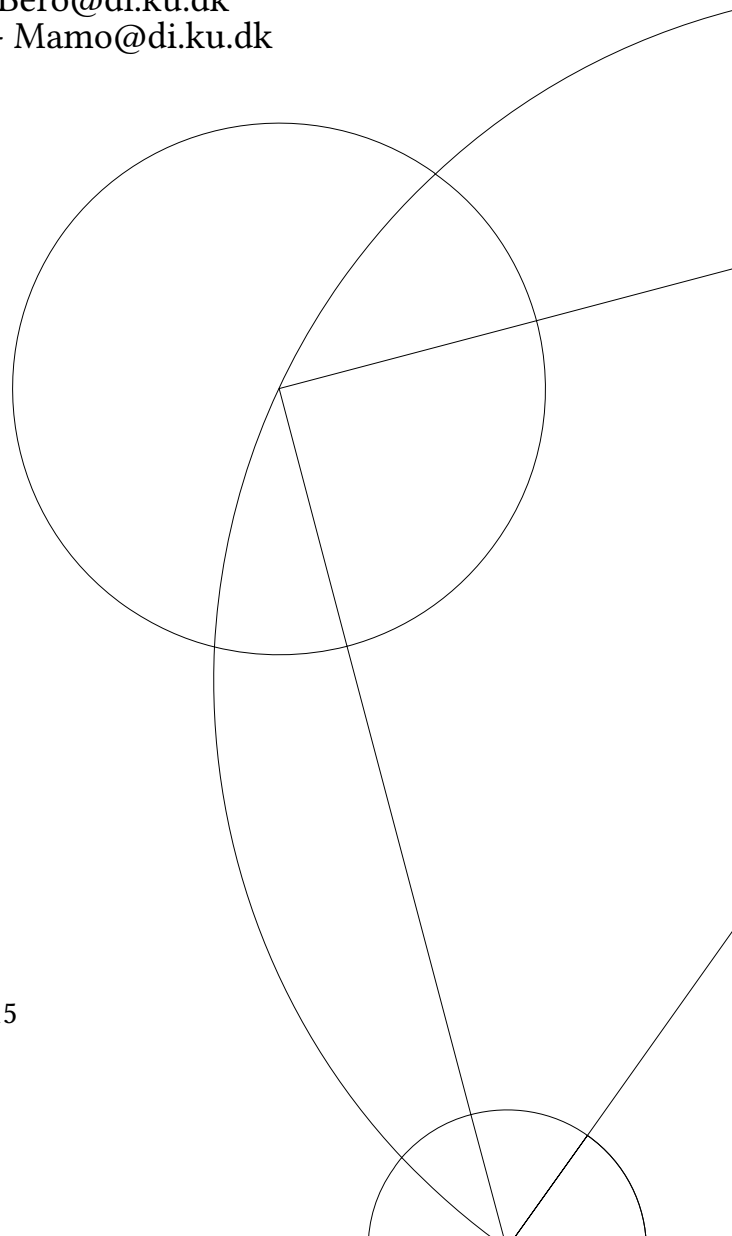


Københavns Universitet

3-Ugers forløb i Data analyse

Arinbjörn Brandsson - Arbr@di.ku.dk
Benjamin Rotendahl - Bero@di.ku.dk
Mathias Fleig Mortensen - Mamo@di.ku.dk

11. maj 2015



1 Introduktion

Vi repræsenterer gymnasietjenesten fra Datalogisk Institut ved København Universitet (DIKU)

Gymnasietjenestens formål er at få flere studerende til at vælge en IT-uddannelse, eller en kombinationsuddannelse af IT sammen med et andet fag og skabe en interesse for datalogi.

Vores plan for at nå dette mål er at udvikle undervisningsforløb til gymnasieklasser der giver en introduktion til datalogi. Disse forløb vil gymnasietjenesten tilbyde at rejse rundt med og overtage undervisningen i et par timer.

Vi estimerer at sådan et forløb ville tage cirka 6 skoletimer, der afvikles over 3 uger, med to timers undervisning per uge. Denne tidsgrænse er dog meget åben og kan tilpasses til hvert forløb.

Forløbet eksisterer i to versioner: et til klasser med en form for IT-fag, og et til klasser uden.

Vi er stadig i udviklingsfasen af disse forløb, og vil derfor sætte stor pris på din feedback.

2 Forløb for IT-klasser

Vi har haft svært ved at finde en undervisningsbeskrivelse af diverse programmerings fag i gymnasiet. Dette betyder at vi er stærkt interesseret i at finde niveauet for f.eks en HTX klasse med programmering som fag så vi kan tilrettelægge vores forløb efter deres evner.

1. Uge - Introduktion til data analyse
Fremvisning af de forskellige teknikker, 'sjov' demonstration og forklaring af den algoritme der passer bedst til deres forsøgsdata.
2. Uge - Introduktion til python og algorithmer
Vi laver nogle python opgaver der skal løses for at gøre eleverne komfortable med sproget python, samtidigt med at vi giver en introduktion til algorithmer.
3. Uge - Kombination af data analyse og python via Machine Learning
I denne uge skal eleverne kombinere deres viden fra de forgående uger og lave et program der benytter sig af machine learning til at forudsige nye data.

Uge 1

I den første uge vil vi introducere eleverne til forskellige teknikker indenfor naturvidenskab til at analysere data. Til dette formål kan der enten blive tilsendt data til os som de studerende har arbejdet med forinden (for eksempel i forbindelse med et eksperiment) som vi kan undervise de studerende i hvordan man kan få nyt information ud af, eller vi kan selv komme med noget data som eleverne kan arbejde med. Her vil vi gå i dybden med én teknik (for eksempel K-means neighbor) som har speciel relevans for den pågældende data. Her vil vi også forklare hvad en algoritme er, samt give en forklaring på hvordan den pågældende algoritme ser ud og fungerer. Til slut vil vi give de studerende nogle små øvelser i "python notebooks" som demonstrerer hvordan teknikkerne fungerer.

Uge 2

I uge 2 vil vi lave python øvelser med eleverne, vi forstiller os at de allerede har evner inden for basal programmering (variabler, data-typer, løkker, if/else betingelser, funktioner). Disse øvelser giver en intuitiv forståelse af algoritmik og køretid, f.eks forskellen på lineær søgning og binær søgning, implementering af sorterings algorithmer.

2.1 Uge 3

I den sidste uge vil vi lade de studerende demonstrere deres nye viden ved at implementere den data analyse teknik som vi snakkede om i den første uge via machine learning. Vi vil give dem en

skabelon som viser hvad strukturen af deres endelige kode skal se ud, med blanke linjer som de skal udfylde. Her vil vi tilbyde nogle forskellige sværhedsgrader (nem, middel og svær) hvor den nemme er for det meste fyldt ud i vores skabelon, mens den sværeste sværhedsgrad har mange manglende linjer kode som eleven skal udfylde. De studerende der hurtigt klarer sig igennem dette kan derefter arbejde på nogle mindre opgaver som vi har forberedt.

3 Forløb for ikke IT-klasser

Vi vil give eleverne en introduktion til data analyse som datalogisk disciplin, og vise hvordan programmering kan bruges til at behandle data.

Forløbet er

1. Uge - Introduktion til data analyse
Fremvisning af de forskellige teknikker, 'sjov' demonstration og forklaring af den algoritme der passer bedst til det valgte data.
2. Uge - Introduktion til python og programmering
Der gennemgås en introduktion til python hvor eleverne lærer basal programmering (variabler, data-typer, løkker, if/else betingelser, funktioner)
3. Uge - Kombination af data analyse og python
Eleverne skal nu sammensætte deres nye viden inden for data analyse og programmering og selv implementere en algoritme hvor vi har givet en skabelon der hjælper eleven på vej.

Uge 1

I den første uge vil vi introducere eleverne til forskellige teknikker indenfor naturvidenskab til at analysere data. Til dette formål kan der enten blive tilsendt data til os som de studerende har arbejdet med forinden (for eksempel i forbindelse med et eksperiment) som vi kan undervise de studerende i hvordan man kan få nyt information ud af, eller vi kan selv komme med noget data som eleverne kan arbejde med. Her vil vi gå i dybden med én teknik (for eksempel K-means neighbor) som har speciel relevans for den pågældende data. Her vil vi også forklare hvad en algoritme er, samt give en forklaring på hvordan den pågældende algoritme ser ud og fungerer. Til slut vil vi give de studerende nogle små øvelser i "python notebooks" som demonstrerer hvordan teknikkerne fungerer.

Uge 2

I den anden uge vil vi introducere eleverne til basal programmering. Da vi regner med at de studerende ikke har prøvet at programmere før vil vi undervise dem i python, som er et nemt og tilgængeligt programmeringssprog for nybegyndere. Vi vil introducere dem til hvad variabler, data-typer, løkker, betingelser og funktioner er, og give dem små opgaver designet til at få dem langsomt introduceret til dette. Målet ved denne uge er ikke at gøre dem til programmører, men blot til at give dem en idé om hvad programmering er, samt hvad det kan bruges til.

Uge 3

I den sidste uge vil vi lade de studerende demonstrere deres nye viden ved at implementere den data analyse teknik som vi snakkede om i den første uge. Vi vil give dem en skabelon som viser hvad strukturen af deres endelige kode skal se ud, med blanke linjer som de skal udfylde. Her vil vi tilbyde nogle forskellige sværhedsgrader (nem, middel og svær) hvor den nemme er for det meste fyldt ud i vores skabelon, mens den sværeste sværhedsgrad har mange manglende linjer kode som eleven skal udfylde. De studerende der hurtigt klarer sig igennem dette kan derefter arbejde på nogle mindre opgaver som vi har tilberedt.