

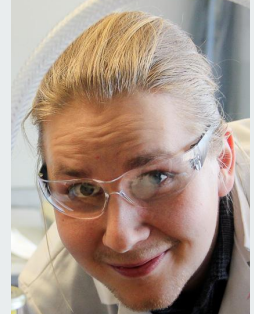


Databearbetning

Steget innan datavetenskap

Lektion 1 - Kursinnehåll & -upplägg, Intro till Python

Dennis Biström
bistromd@arcada.fi



Kursinnehåll



Kompetensmål

Målsättningen med kursen är att bekanta er med centrala begrepp inom datavetenskap och lära sig behandla olika typer av data på praktisk nivå.

Kurslitteratur

Största delen av materialet är i elektronisk form
Ni klarar er långt på pandas och pythons dokumentation, förutsatt att ni *orkar läsa...*

<http://pandas.pydata.org/>

<https://docs.python.org/3/reference/index.html>

“Some of us just like to read” - Lady Gaga
Python for Data Analysis, Wes McKinney (O'Reilly Media)

Jag ska givetvis försöka ge er effektivare och mer intressanta resurser t.ex i videotutorialformat.

Kolla gärna **systerkursen** vid Harvard: [CS109 Data Science](#)

Föreläsningssinnehåll

Datahantering med Python och Pandas
Web scraping och data APIs
Visualisering, Tidsserier och signalbehandling
Bilder som data

Läranderesultat

Efter avklarad kurs förväntas den studerande vara förmögen att

- Hämta, formatera, och visualisera data med hjälp av Python, Pandas och matplotlib
- Behandla olika sorters data (ljud, bild, csv, json...)
- Göra web scraping och använda data APIs (GET & Parsing)
- Känna till signalbehandling (t.ex. ljudfil med excel, audacity)
- Analysera bilddata och grafer (info i alpha, visualisering)

Praktiskt om kursen - Kursplan och förväntningar



Elevens ansvar

Kursen består av 7 lektioner (8). Jag använder inte lektionerna bara på att stå och föreläsa, utan vi använder en stor del av lektionstiden på uppgifter.

Föreläsarna på Lynda är världsbäst på vad de gör, och varje kurs har ett planerings & analysteam bakom sig, något jag inte har. Använd alltså Lynda!

Utöver lynda så ska kursen träna er i informationssökning, ert viktigaste verktyg som programmerare.

Jag menar inte att ni går kursen på egen hand, jag ger resurser enligt förmåga när jag märker att ni "kör fast".

Kursplan

Ja presenterar en deluppgift (grovt baserad på teorin)

Vi gör en del av uppgiften i klassen

Jag hjälper er komma vidare med uppgiften

Resten av uppgiften slutför ni till nästa gång

Tidsplanering

Från eleven väntas en insats á 133h. De här timmarna är fördelade på följande sätt:

Föreläsningar	3/20 h
Praktiska övningar	1/13 h
Projekt- och produktionsarbete	0/60 h
Självstudier	0/40 h

Praktiskt om kursen - Bedömning och närvaro



Bedömning

Vitsordet bestäms på basis av era lösningar på kursuppgifterna.

Maxpoäng 10 per uppg

Uppgifterna ska lämnas in inom utsatt tid för att man skall kunna få fulla poäng för dem. 25% avdras för sen inlämning.

Varje uppg bygger på kunskap från den föregående, så försök att inte falla av kälken.

Närvaro

Jag använder mig av en närvarolista.

De som inte har deltagit på nån av de två första föreläsningarna blir borttagna från ASTA

För mig är det viktigt att ni deltar på lektionerna så ni inte faller efter med uppgifterna.

Ifall ni närvarar på mindre än 50 % av kursen förlorar ni mycket av er "förhandlings- och klagomålsrätt" då det kommer till er kursprestation och ert vitsord.

Intensiv 4 veckors kurs! - 8 lektioner, 4 uppg



Upplägg

Föreläsningar med exempel - Var på plats, följ med!

Videoföreläsningar (60–120 min) att se på hemma

Veckouppgifter med deadline **varje** vecka.

Inget kodtilfälle! Använd F369 och fråga kaveri?

Kursverktyg

Python

Pandas (Python Data Analysis Library)

Jupyter Notebook

Installering: Anaconda (Linux / Mac / Windows)

<https://www.continuum.io/anaconda-overview>

Bedömning

Vitsordet bestäms på basis av era lösningar på kursuppgifterna. Maxpoäng 110p

Varje uppg är värd 20p. 3 förhör 10p, 3 läxor 10p

Bonus upp till 10p för smarta lösningar elr tilläggsfunktioner

5p avdrag per förseningsvecka

Närvaro

Jag använder mig av en närvarolista.

De som inte har deltagit på nån av de två första föreläsningarna blir borttagna från ASTA

<70% närvaro => begränsad klagomålsrätt

Upplägg - 10 lektioner, 4 inlämningsuppg



- | | |
|--|------------------------|
| Lektion 1 - Kursinfo, verktyg & resurser, Intro till Databearbetning. My first python app | Läxa 1 ut |
| Lektion 2 - Python teori & quirks, My second and third app. Läxa 1 hjälp? | Läxa 1 in, Förhör 1 ut |
| Lektion 3 - Pandas basics, Uppg 1 start, Python in class övningar | Förhör 1 in, Läxa 2 ut |
| Lektion 4 - Datasets och webscraping, Hjälp med Inlämningsuppg 1, Övning 3 (Lab 2), | Läxa 2 in |
| Lektion 5 - Inlämningsuppg 2, Läxa: Numpy, Matplotlib, Pandas på DataCamp | |
| Lektion 6 - Visualisering och Numpy och Matplotlib Övning, | |
| Lektion 7 - Förhör 2 (Numpy, MatPlotLib & Pandas), Ljud och Bilder som data, Inlämningsuppg 3 start | |
| Lektion 8 - Inlämningsuppg 3 fortsättning, Övning med Bilder och Signaler. | |
| Lektion 9 - Deadline för Förhör 2 och Inlämningsuppg 3. Kodande & Feedback, Julglögg? | |

Idag - Getting started, lite teori och Läxa 1



Getting started - Installera Anaconda och lägg upp utvecklingsmiljön

Vad är databearbetning *och datavetenskap?*

Resurser

Läxa 1

- Sten sax påse
- Tankeläsaren
- Optimus Prime

Om Verktygen



Anaconda - En python platform med verktyg för datavetenskap och maskininlärning

[Getting started](#)

Anaconda har 6milj användare och är i sig självt skriven i python ;)

Conda - OS, Xplatform paket OCH omgivningshanterare. Går alltså att köra app1 i ena env och app2 i andra env

```
conda install NumPy, pandas, matplotlib, Seaborn, Jupyter
```

Jupyter - Spinoff till [Fernando Pérez](#) tidigare IPython (-2014)

Koda, kör, visa output och kommentera i ett och samma dokument.

.ipynb filen är egentligen data i JSON format, men jupyter filer går att ladda ner som bl.a. HTML, LaTeX, PDF, Markdown och såklart Python filer. (Genom att använda [nbconvert](#))

Man kan såklart utveckla python med en IDE som t.ex JetBrains [Pycharm](#), men själv tycker jag bra om .ipynb

[Skriv kod, skicka till kompis](#) kör med shift+enter, inkludera kommentarer med [LATEX](#) syntax.

[DOCS](#)

Idag - Getting started, lite teori och Läxa 1



Getting started - Installera Anaconda och lägg upp utvecklingsmiljön

Vad är databearbetning *och datavetenskap?*

Resurser

Läxa 1

- Sten sax påse
- Tankeläsaren
- Optimus Prime

Vad är datavetenskap? - o vadå databearbetning?

Datavetenskap handlar om att hitta ny kunskap från data via beräkningsmodeller, statistik och visualisering

Varför datavetenskap? [Slides](#)

[Nate Silver](#)

Det fanns tidigare en sida som hette isnatesilverawitch.com

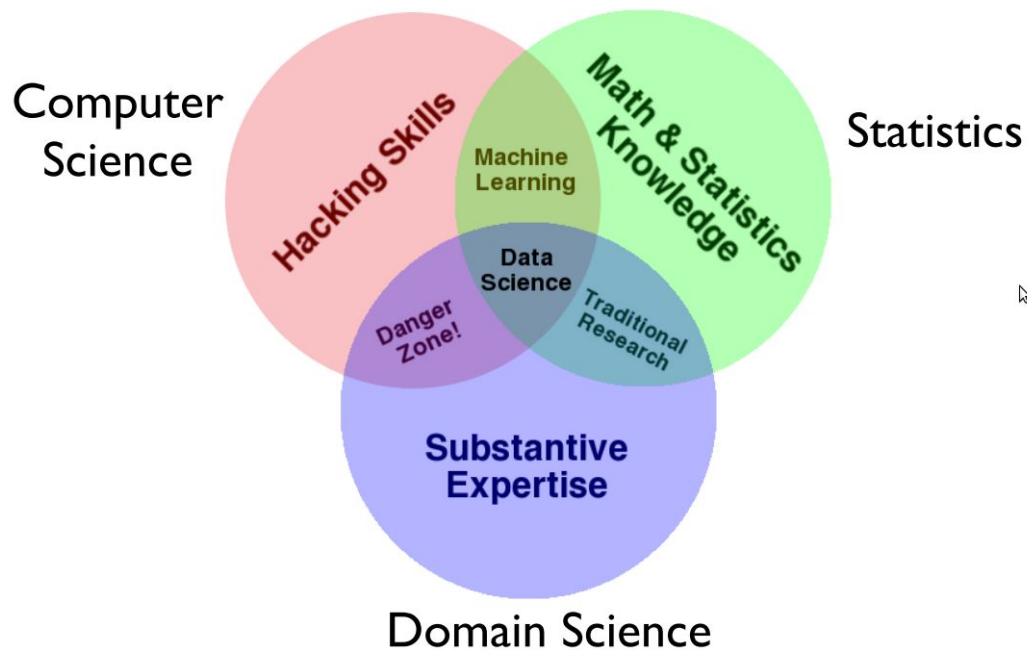
2008 lyckades han förutspå röstresultatet i 49 av 50 stater. På sajten stod det "no" på med förklaringar över hur Nate hade kommit till sin prediction mm.

2012 lyckades han förutspå röstresultatet i alla 50 stater. Efter det stod det "While we cannot say yes or no with any certainty, Nate Silver might, in fact, be a witch."




Vad är datavetenskap? - o vadå databearbetning?

A data scientist is someone who knows more statistics than a computer scientist and more computer science than a statistician



Människa + information = sanning? Movie time!



Illusionen om sanning

Depression i vetenskap

Vad är kändisar

Cola® är inte godare än Rainbow Cola

The illusion of thruth - Veritasium

<https://youtu.be/cebFWOlX848>

PS. om ni sett den här.. Vinn en miljon på [kaggle.com](https://www.kaggle.com)

Idag - Getting started, lite teori och Läxa 1



Getting started - Installera Anaconda och lägg upp utvecklingsmiljön

Vad är databearbetning och datavetenskap?

Resurser Klicka genom resurser och skriv ner vilka ni tror passar er.


Nästa gång har vi en poll och lite in class diskussion.

Hitta er väg till learnpython.org och gå genom basics fram till functions (9 onde sidan)
20 min?

Läxa 1

- Sten sax påse
- Tankeläsaren
- Optimus Prime

Resurser 1 - learnpython.org och Derek B?

- 
- Cheat sheet: [Anaconda Cheat Sheet - Getting Started - PDF](#)
[Pandas Cheat Sheet - PDF](#)
- Manual/Docs [Conda package manger - Docs](#)
[Pandas - QuickStart & Cookbook](#)
- Tutorials (text) [Anaconda Getting Started - User Guide](#)
[Python - Intro till avancerat - Övningar och förklaringar !!!](#)
[Pandas tutorial - PythonSpot](#)
[Intro to data science Numpy, MatPlot & Panda](#)
[\(Pandas - How do pivot tables work - ExcelCampus\)](#)
- Tutorials (video) [Socratica python tutorial - Youtube](#)
[Derek Banas - "Learn Python in one video"](#)

Resurser 2 - Interaktivitet och Linkedin learning



- Interaktiva: [Intro to **python** for data science - Gratiskurs - DataCamp](#)
[Intro to python for data science - Ch4 - **Numpy** \(DataCamp\)](#)
[Intermediate python for data science - Ch1 - **MatPlotLib** \(DataCamp\)](#)
- Lynda: [6h nybörjarkurs **Python** för datavetenskap med Lillian Pierson - Lynda](#)
[2h intermediate - **Numpy** Data Science Essentials - Charles Kelly](#)
[Intermediate - Ch3: **Numpy**, Ch4: **Pandas**, Ch 9: **matplotlib** - Miki Tebaka](#)
[2h intermediate kurs i **Pandas** med Jonathan Fernandes - Lynda](#)
[2h intermediate **Pandas** för Datavetenskap med Charles Kelly - Lynda](#)
[Big Data Analysis in python using **Numpy** and **Pandas** - Michele Vallisneri](#)

Resurser 3 - Ingen motivation? Kanske nåt “roligt”?



Interaktiva roligheter

[The Python Challenge](#)

[How to think like a Computer Scientist](#)

[CodeSignal - Interaktiva utmaningar, badges, points etc.](#)

[Reddit daily programmer challenges](#)

Vill du vinna 1 miljon \$

[7h gratis tutorial på Kaggle - Känner ni till kaggle?](#)

Python 2 vs Python 3 trubbel, kolla [här](#)

Python basics - Hur kändes resurserna?



Python - GPP, bygga webbsidor, analysera data, koda verktyg

Kommentar, även `""" Multiline comment """`

Variables - behöver endast ett namn, tolken känner igen typen

Strings - "Text" eller 'strängar'

Använd `\` för att skriva t.ex citattecken bland strängar.

Numror - int float complex - Märk decimaltecken . inte , . Märk även j för komplexa tal, inte i

Tolken hanterar användningen av minne, även vid typkonversioner

int -> float -> complex. Python konverterar till bredare format innan aritmetik.

Aritmetik - % modulo returnerar resten, // returnerar kvoten, ** fungerar som exponent. *Se upp! $2^{1/2} = ?$*

Booleans - = för tilldelning (assignment), == för utvärdering. Efter `str(True)` går variabeln inte att använda i logik!

If elif else - `raw_input("Mata in en sträng")`. se upp för felaktig inmatning (Error handling).

Python quirks - Indentation styr koden, försiktigt med mellanslag! Kolontecken efter if och else.

Python basics - Interaktiv hjälp



dir() - Se vilka moduler, objekt, klasser och metoder ni har. T.ex efter *import math* visar dir() även math modulen

help(someObject.someMethod()) - Få tilläggsinformation om objekt, metod eller funktion

someObject.someMethod? - Visa docstring

jupyter-notebook - tryck tab 1-4 gånger för att utöka information om det ni håller på att skriva just nu

Python basics - Funktioner och loopar



Funktioner utför uträkningar, formatterar data, inkapslad funktionalitet. Definiera funktion med *def*.

Funktioner tar ofta emot data, **parametrar**. Funktioner **returnerar** ofta värden

`def funcname(params1[, param2])` Parametrar inom parenteserna. `[]` i docstring betyder valfri parameter

`def g(y, x=0)` Obligatoriska parametrar först, valfria måste nämnas vid namn

`g(3, x=5)` Kalla på funktionen genom att ge obligatoriska värden först, valfria parametrar med namn!

While loopen - "Programloop", alltid om ni tänker copypastea funktionalitet, tänk, kan jag loopa?

<code>while 1 == 1:</code>	<code># Alltid sant</code>
<code>if input = "exit":</code>	<code># Undantaget</code>
<code>break</code>	<code># Stoppar en while loop</code>
<code>elif input = "restart":</code>	<code># undantaget som skippar funktionalitet</code>
<code>Continue</code>	<code># börja en ny iteration</code>
<code>else:</code>	<code># I alla andra fall</code>
<code>#programlogik</code>	<code># Gör "som vanligt"</code>

Idag - Getting started, lite teori och Läxa 1



Getting started - Installera Anaconda och lägg upp utvecklingsmiljön

Vad är databearbetning och datavetenskap?

Resurser Klicka genom resurser och skriv ner vilka ni tror passar er.

Nästa gång har vi en poll och lite in class diskussion.

Hitta er väg till learnpython.org och gå genom basics fram till functions (9 onde sidan)
20 min?

Läxa 1

- Sten sax påse
- Tankeläsaren
- Optimus Prime

Läxa 1 (en av tre) - Sten Sax påse



Skapa ett sten sax påse spel - Googla inte svaret! Det här är en övning i programlogik, if och else

1. Låt spelarna skapa två användare (Mata in sina namn)
 2. Be spelarna turvis välja en av de tre alternativen
 3. Bygg logik som returnerar t.ex "Dennis vann" eller "Ingen vinner!"
- Bonus: Fråga ifall de vill spela igen?

Reglerna: Sten vinner sax, sax vinner papper, papper vinner sten

Utmaningar:

1. Hur får ni inputs utan att kompisen brevid ser?
2. Hur skapar ni programlogiken?
3. Bonus: Hur implementerar ni spela igen?

Läxa till nästa gång! - Det blir förhör!



Kolla igenom resurserna på itslearning & slides

Lär er om datatypes i python med sokratica

Kolla på båda lynda tutorials upplägg.

Slutför Sten Sax Påse

Lek med exercise.ipynb filen på IL (nästa gång)

Berätta för mig nästa gång hur nivån känns

