# Data Intelligence – ITU

## Undervisningsgang 1 – Introduktion

Set = unikke set og er ordnet. Dette kan være mere brugbart end en liste.

Dictionary = Gemme værdier, som kan kaldes

% 2 == 0 = kan bruges til at kalde/tjekke om ting fx i en liste kun er lige tal

Vi skal lære, om mutable og immutable objects og hvis disse bruges i funktioner, kan der være udfordringer.

**Beginning data analysis**

* Features/ dimensions 🡪 a column in the dataset
* Element/instance/example 🡪 row in the dataset
* Summary statistics: describing data using varioys aggregating methods, e.g. sums, means, variation
* Inferential statistics: learning how the data is generated, predicting a future \_\_

**Web scraping**

Python packages : request and Beautifulsoup

Additional package: scrapy

response = requests.get(“”)

soup = bs4.BeautifukSoup(response.text, “html.parser”)

soup.find =(“span”, “text”)

### Undervisningsgang 2 – web scraping part 2

With open file 🡪 “with” bruges til at åbne og lykke filen. Uden with så skal man selv lukke filen.

Danmarks statistik bruger den gammeldags latin1 encoded.

[I for i in range(?)] er det samme som et for loop

**Javascript scraping**

Selenium – the browser becomes scriptable

Kig på slide for kode til chrome

To forskellige måder til at selektere forskellige dele på en hjemmeside.

Xpath (ville Jens bruge, men kan ikke bruges med BeautifulSoup)

CSS selectors (kan bruges med beautifulsoup)

Load til selenium

Driver.get(”url”)

Stocks = driver.find

Når man bruger selenium

Fx pas på cookiebokse. De skal acceptere.

Def set\_up\_browser():

Options = selenium.webdriver.firefox.options.Options()

Options.add\_argument(--headless—’)

Options.add\_argument(‘—no-remote’)

Browser = selenium.webdriver.Firefox(executable\_path=”geckodriver/geckodriver”

Browser.save\_screenshot(“dr.png”)

Åbne delecveloperboks på hjemmesiden (hvor/hvad??)

Browser.get(”https://dr.dk”)

Btn = browser.find\_element

Btn.click()

**JSON**

Import json

Response.json() is the same as json.loads(response.text)

Json.dumps(d) 🡪 viser string som den ville være i json

Python Dict og json objects er næsten det samme fx {’key’: value}

Ved CSV kan du loade dele af en fil ved JSON skal man loade alt på en gang. Det kan være en ulempe hvis man ikke har nok hukommelse på computeren. En workaround spørg Jens, hvis relevant!

Rektangulær data fx dem der kan komme ind i excel har to dimensioner 🡪 csv

JSON kan have alle dimensioner af data. Man kan vise embedded expression i JSON det kan man ikke i CSV. Fx (”a”: 123, ”b”: {”fj”:5668, ”fg”:688})

**API’s**

Autenticitet

Requests.get(url, auth=(username, password))

Nogle skal også kun have en key til at vise autenticitet

Import tweepy

Auth = tweepy.OAuthHandler(key,secret)

Api = tweepy.API(auth,wait\_on\_rate\_limit = True)

Tweets = tweepy.Cursor(api.search\_tweets, “#dkpol”, lang=”da”, since =”yyyy-dd-mm”)

For batch in tweets.iterator:

For tweet in batch:

Print(tweet.user,screen\_name, tweet.created\_at, tweet.text)

Extract score kan man gennem REGEX

EXTRA: SCRAPY (se slide)

## Undervisningsgang 3

Uniform distribution – lige sandsynlighed for en fordeling

Exponential distribution – ulige sandsynlighed fx skewed

Normal distribution – klokken

## Undervisningsgang 4 – data analysis part 2

Summary/descriptive – siger noget om sample

Inferential – forudsiger hvordan populationen ser ud and the relations to the population. Ikke alle dog owners are people (måske noget med at man kan forudsige hvilke hunde der er?)

Distribution – how sample are distributed. Natural phenomena will often tend to be normally distributed, popularity tend to be exponential distributed and dice throw tends to be uniform distributed

*Pandas*

*Hele datasættet er et dataframe en kolonne er en series*

Type() = typen af data.

Df.columns = kollonnenavne

Når man loader filen skriv: skiprows=1, names = coloumn names efter stien

Df.info() = basisinformationerne. Object betyder alt der ikke er et nummer – ofte er det en string. Fx kan man se, hvor god pandas har været til at læse den korrekte type.

Df.describe()

Vælge en liste er strenge = df[[”fat-total”, ”calories”]] yderste brakets er index det inderste er en liste.

For at fikse at der er noget galt i hvad der bliver læst af pandas:

Vælg noget der er i overenstemmelse med en betingelse:

Df[”fat-saturated”] > 0.5 🡪 her vil der forekomme en boolean med true og false

Han bruger unique() for at finde, hvad listen indeholder

Df[”fat-saturated”] >= 0.1.unique()

Det kan bruges til at indeksere of finde fejlen.

Df[Df[”fat-saturated”] >= 0.1]

Nedenfor gemmer hele datasættet for dette subset: (det kan man se ved .coloumns)

­­\_\_df = Df[Df[”fat-saturated”] >= 0.1]

Den med fat-total

.map kan bruges til at transformere data

Df[”fat-saturated”].map(lambda x: x + 1)

Df[”fat-saturated”].map(lambda x: type(x))

Lambda adderer med 1 uden at definere en funktion

Definerer en funktion som finder floats

I pandas skal man ikke bruge for loops

Try:

Float(“3.1”)

Except valueerror as e:

Print(“not float”)

Try:

Float(“3 1”)

Except valueerror as e:

Print(“not float”)

Def is\_float(row):

Try:

Float(row)

Return True

Except ValueError:

Return False

Df[~Df[“fat-total”].map(is\_float)]

Df[~Df[“fat-total”].map(is\_float)][“fat-total”]

~ = vender en kolonne om

Logics

Deduction – eksisterende data bliver instrumentaliseret

Abduction – stiller nye spørgsmål på baggrund af data

Induction - generalisering

I eksamen skal man kunne forklare hvad man forventer af sin data og hvordan disse forventninger hænger sammen med de analyser du vil foretage.

*Scrape coop-data med pandas*

Hvis man søger på en \* kommer alt op

I didentificer kan man bruge network for at se hvilke processer der sker (requests browser has made) kan give os en url.

Efter denne url skal man komme =\* på for at få alt

Import requests.get(url)

Response.json 🡪 giver dog kun 30 varer

Load et json object

Df = pd.Datafram(products)

Df[”salesprice”].map( lambda x: x[“amount”])

Ny kolonne til datasæt:

Df[“sales-expected”] = Df[”salesprice”].map( lambda x: x[“amount”])

Man kan gemme til en csv:

Df.to\_csv(”coop.csv”)

## Undervisningsgang 5 – inferential statistics

Linear regression

Random forest

Eksempel:

Stil en hypotese

Step 2 hvad er der i vores datasæt?

Df.plot.scatter

Pandas groups kan hjælpe med at gøre det mere intelligent.

Species = df[”species”].unique()

For s in species:

S\_df = df[df[”species”] == s]

L =

Den smarte

Figure, ax = plt.subplots(figsize =( 12, 12))

Species = df[“species”].unique()

Plots = []

For s in species:

S\_df = df[df[”species”] == s]

L = ax.scatter(s\_df[“colmen\_lenght\_mm”], s\_df[“culmen\_deåpth\_mm”])

Plot.append(l)

Ax.legend(tuple(plots), tuple(species))

**Pandas!**

Se hans eksempel for hvordan man sletter alle problematiske kolonner hvor der er skrevet forskert.

Def is\_float8row):

If row is Nobe:

Return None

Try: float(roe)

Return(row)

Except ValueError:

Return None

Brug dem sådan her for at håndtere missing data:

Df[”fat-total”].map(is\_float).isna()

Så kommer true eller false

Hvis vi er ligeglade med hvilke det er så brug fillna(). Det er dog ikke den bedste løsning. Der er forskellige strategier, som er brugbare. Vi vil cirka om to lektioner se på, hvilke der er mest nyttige

Df[”fat-total”].map(is\_float).fillna(0)

Brug dropna() hvis man ikke vil bruge kolonnen. Det er okay, hvis man blot har enkelte problematiske rækker.

Df[”fat-total”].map(is\_float).dropna()

Rækken han arbejder med er stadig object nu kan astype(float) benyttes til at ændre typen. Nu kan man lave describtives på det.

Nu kigger vi på group vi ser særligt på de kategoriske.

Man bruger grouby() på hele datasættet

Fx df.groupby(”beverage-category”).mean()

Denne operation kan bruges på forskellige hierarkiske niveauer fx en ydre kategori og en indre. Jens mener dog at mennesker lidt bliver forvirret og så skal man kigge på enkelte features. At kigge på enkelte features er bedre til at præsentere data

df.groupby([”beverage-category”, “bevaerage”]).mean()

brug agg()

df.groupby(”beverage-category”).agg({“sugar”: “median”, “mean”})

vi kan bruge egne udregninger. Han fik det dog ikke til at virke, da der er tale om grupper af rææer og ikke enkelte rækker

def sum(row):

return row + 1

df.groupby(”beverage-category”).agg

man kan også bruge flere keys

df.groupby(”beverage-category”).agg({“sugar”: “median”, “mean”, “cholesterol”: “median”})

value\_counts() funktionen kan bruges med groupby. Han fnadt ud af de virjkede bedre med size()

df.groupby([”beverage-category”, beverage”). Size()

**Inferential statistics**

You aim for an arbritary number in linear regression

With classification you aim to infer a category

Vi kan halvere vores sample size, så vi kan teste mod halvdelen af datasættet

Test size ved mindre datasæt tag 10%

X\_train, X\_test, y\_train, y\_test =

Train\_test\_split(df[[“sugar”]]

Overfitting er et problem hvis vi har en trænet model, som ikke har noget at teste imod.

Kategorisk data (nominal og ordinal) 🡪 god til random forest

Quantitative data (diskret og kontinuer) 🡪 god til lineær regression

**Undervisningsgang 6: Visualization**

Klassifikation - learning to assign a class. Assign label we already know. Can be continues.

Lineær regression – works on continues data used to predict values. Assigning a single value that exist arbitrary on the range. Can only be continues.

I eksemplet med pingvinerne så er det mere interessant at se på populationen og hvad der gør den specielle.

R^2 er en analyse der siger noget om, hvor meget den uahængige variabel forklarer den afhængige variabel

Precision and recal: hvorfor er vi ofte mere interesseret i disse end i accuracy? Accuracy er nummeret af korrekte forudsigelser ud af alle forudsigelser. I de tilfælde hvor der er lige eksempler er det okat, men problemet opstår, hvis der er et usymetrisk antal af klasser. Recall portionen af true positive/true positives and false positive

F1 = en kombination af precision and recall. 2x precision x recall / precision + recall. Harmonic mean of proportion of positive predictions that is correct

.get\_dummies = man laver binære numre

Np.random.RandomState(0) = producerer den samme randomness hver gang

Support = antallet af pingviner i testdelen

Det er ikke et mål for prection model over pingvinerne. Vi ønsker at teste, hvorledes at de fysiske attributer, kan sige noget om hvilket pingviner , der allerede er.

Visualiseringer er en vigtig del af den eksplorative fase. Vigtig måde at formidle til de, der ikke har så meget erfaring med data.

Noter til visualisering. Mennesker er ikke gode til at overskue arealer derfor ikke piecharts og donut charts

Never plot in 3d (unless you’re a physicist)

## Undervisningsgang 7 – preprocessing

Håndtering af Danmarks statistik.

Der er ofte en bemærkning til sidst.

Hvordan man omorganisere finansielle kvartale bør man give et nyt index som python forstår.

Df.index.map(covert\_date) 🡪 kan konvertere alle rækker til timestamps ved hjælp af def, som jeg har taget billede af. Denne skal man så huske reassigne så man har det nye datasæt.

Nu skal man håndtere missing data. På det datasæt vi har med æggene, giver det mening at slette de sidste linjer, da der slet ikke er noget indhold i dem.

Df.dropna(how=”all”)

Nu vil vi gerne transformere Danmarks statistik (”..”)

Df.replace(”..”,np.nan)

Def covert\_to\_float(x):

If x is None:

Return np.nan

Try:

Return float(x)

Except

Df.applymap(convert\_to\_float)

En måde at håndtere den nu manglende data introducing gitter into the plot

Hvorfor giver det mening? Replacing missing values de blå ville ikke ligget klumpet sammen ude i siden. Means er derfor dårligt i den her sammenhæng.

I stedet df.fillna(method = ”bfill”)

**Eksamensworkshop**

Hold din opgave op mod ILO

Spørgsmål?

* Referencesystem?
* Which are accepted languages?

## Undervisningsgang 8 – text processing 1

Trin: Preprocessing, modelling (inferential statistics)

## Undervisningsgang 10 – data formats

Normaldict – der er keys ikke rangeret der garanterer ordereddict.

Tidy data:

Link structure with meaning

In this paper, the chosen provided data set “…..” will be investigated. Firstly, data from the data set will be explored and described to create an initial understanding and contextualization of the data. Secondly, a research question will be posed based on the findings from the exploration and contextualization phase. From this research question initial hypothesis about the data will be formed and will include expectations to the analysis’ outcomes. Thirdly, data will be prepared and cleaned according to taught concepts and technics from the Data Intelligence course. Fourthly, the data will undergo different analyses to validate or falsify the posed hypothesis. Finally, the findings from the analyses will be discussed together with eventual limitations and reflections to the data set and analyses. Then the hypothesis will be revised followed by a conclusion to this paper.

## Undervisningsgang 11 – Critical Data analysis 19/04

.map is used to apply to each element of a series.

Eksempel ændre så alle W og M’er bliver til 0 og 1

Det kan eksempelvis laves med en funktion som .map kan tage eller lambda eller smartere endnu dictionaries.

Nedenfor eksempel på funktion

Def transform(value):

If value == ’m’:

Return 1

Elif value == ’w’:

Return 0

Dictionary

D= {’m’ :1, ’w’:0}

.map(D)

Hvordan bruger man .loc?

Mask = df[’owner-gender’] == ’m’

Hvis kun bruges på en kolonne:

Df.loc[mask, ’owner-age’]

Bruges på flere kolonner

Df.loc[mask, [’owner-age’, ‘owner-gender’]] = ‘daw’

Biases

Over or underestimates the targets

“raw-data”

## Session 13

Comments for trial exam

.unique() 🡪 den giver ikke I sig selv et godt overview. Prøv med value\_counts()

Lesseon 5 for evaluation methods

Precesion and accuracy, recall F1

Accuracy quickly gets misleading

F1 – kombination af præcision og recall

Recall – hvor mange af dem der ikke har kræft, klassificerer vi korrekt (vores egne klassificeringer)

Precision - hvor mange af dem der har kræft, klassificerer vi korrekt (klassificeringer som modellen har lavet)

.groupby 🡪 en finere deskriptiv analyse

Husk litteraturliste

Splitte i test og train sets (test test er en simulation af vores populaation) vi forstår at det kan bruges til noget andet. Hvis r^2 er for småt, så kan r^2 svinge meget.

.reset\_index 🡪 fra serie til dataframe

Df[”rating-round”] = round(df[“ratings”])

S = df.groupby(“rating-round”).mean()[“price”]