

# Datavisualisering med Streamlit

Af Henrik Sterner ([hst@nextkbh.dk](mailto:hst@nextkbh.dk))

Datavisualisering er en vigtig del af data science og maskinelæring. Det er vigtigt at kunne visualisere data for at kunne forstå dem og for at kunne kommunikere til andre hvad dataene viser.

Til dette formål findes der mange forskellige værktøjer. I denne workshop vil vi kigge på et af dem: Streamlit.

# Om brugen af disse slides

- ▶ Disse slides forsøger at eksemplifere en lang række af de vigtige begreber i streamlit
- ▶ De må på INGEN MÅDE KOPIERES ELLER BENYTTES uden tilladelse fra Henrik Sterner
- ▶ De er lavet i markdown og kan derfor nemt konverteres til andre formater
- ▶ Ved brug af Visual Studio Code kan de konverteres til HTML, PDF, PowerPoint, Word, LaTeX og mange andre formater
- ▶ Slides er tilgængelige på [github.com/henriksterner/IntelligenteSystemer/slides](https://github.com/henriksterner/IntelligenteSystemer/slides)

# Hvad er Streamlit?

Streamlit er et værktøj til at lave interaktive webapplikationer til visualisering af data. Det er et open source projekt, der er udviklet af et startup i Silicon Valley. Streamlit er skrevet i Python og er derfor nemt at bruge sammen med andre Python-baserede værktøjer til data science og maskinelæring.

Streamlit er et relativt nyt værktøj, der først blev udgivet i 2019. Det er derfor stadig under udvikling og der kommer hele tiden nye funktioner til. Det er dog allerede nu et meget brugbart værktøj, der er nemt at bruge og som giver flotte resultater.

# Installation

Streamlit kan installeres med pip:

```
pip install streamlit
```

Hvis du bruger Anaconda kan du også installere det med conda:

```
conda install -c conda-forge streamlit
```

Hvis du bruger Google Colab er Streamlit allerede installeret.  
Streamlit og jupyter notebook er dog ikke compatible, så du kan ikke bruge dem sammen.

## Teste om streamlit virker

For at teste om Streamlit er installeret korrekt kan du køre følgende kommando i en terminal:

```
streamlit hello
```

Dette vil starte en lokal webserver og åbne en browser med en simpel Streamlit applikation. Hvis det virker er Streamlit installeret korrekt.

# Første Streamlit applikation

For at lave en Streamlit applikation skal du oprette en Python fil med følgende indhold:

```
import streamlit as st
st.title("Hello World!")
```

Dette er en meget simpel Streamlit applikation, der viser en titel.

# Afvikling af Streamlit applikationer

For at køre applikationen skal du køre følgende kommando i en terminal:

```
streamlit run hello.py
```

Dette vil starte en lokal webserver og åbne en browser med din Streamlit applikation. Hvis du ændrer noget i din Python fil og gemmer den, vil Streamlit automatisk genindlæse applikationen og vise ændringerne.

## Streamlit applikationer i Google Colab

Hvis du bruger Google Colab skal du ikke køre denne kommando. I stedet skal du køre følgende kommando i en kodecelle:

```
!streamlit run hello.py &>/dev/null&
```

Dette vil starte en lokal webserver og åbne en browser med din Streamlit applikation. Hvis du ændrer noget i din Python fil og gemmer den, vil Streamlit automatisk genindlæse applikationen og vise ændringerne.



## Streamlit applikationer i Jupyter Notebook

Hvis du bruger Jupyter Notebook skal du ikke køre denne kommando. I stedet skal du køre følgende kommando i en kodecelle:

```
!streamlit run hello.py &>/dev/null&
```

Dette vil starte en lokal webserver og åbne en browser med din Streamlit applikation. Hvis du ændrer noget i din Python fil og gemmer den, vil Streamlit automatisk genindlæse applikationen og vise ændringerne.

# Streamlit applikationer i Visual Studio Code

Hvis du bruger Visual Studio Code skal du ikke køre denne kommando. I stedet skal du trykke på “Run” knappen i øverste højre hjørne. Dette vil starte en lokal webserver og åbne en browser med din Streamlit applikation. Hvis du ændrer noget i din Python fil og gemmer den, vil Streamlit automatisk genindlæse applikationen og vise ændringerne.

# Widgets i Streamlit

Streamlit har mange forskellige indbyggede widgets, der kan bruges til at lave interaktive applikationer. Eksempler på widgets er: \*

- Knapper
- Checkbokse
- Dropdowns
- Sliders
- Text inputs
- Date inputs
- File uploads
- Plot widgets
- Map widgets

# Knapper

Et af de mest brugte widgets er knapper. Knapper kan bruges til at lave interaktive applikationer, hvor brugeren kan trykke på knapperne for at ændre på applikationen.

For at lave en knap kan du bruge følgende kode:

```
import streamlit as st
st.button("Klik her")
```

Dette vil lave en knap med teksten "Klik her". Hvis du trykker på knappen vil den skifte farve.

# Checkbokse

En anden meget brugt widget er checkbokse. Checkbokse kan bruges til at lave interaktive applikationer, hvor brugeren kan sætte flueben i checkboksene for at ændre på applikationen.

For at lave en checkboks kan du bruge følgende kode:

```
import streamlit as st
st.checkbox("Vis tekst")
```

Dette vil lave en checkboks med teksten "Vis tekst". Hvis du sætter flueben i checkboksen vil den skifte farve.

# Dropdowns

En anden meget brugt widget er dropdowns. Dropdowns kan bruges til at lave interaktive applikationer, hvor brugeren kan vælge en værdi fra en liste for at ændre på applikationen.

For at lave en dropdown kan du bruge følgende kode:

```
import streamlit as st
st.selectbox("Vælg en værdi", ["Værdi 1", "Værdi 2", "Værdi 3"])
```

Dette vil lave en dropdown med teksten "Vælg en værdi" og med tre værdier: "Værdi 1", "Værdi 2" og "Værdi 3". Hvis du vælger en af værdierne vil den skifte farve.

# Sliders

En anden meget brugt widget er sliders. Sliders kan bruges til at lave interaktive applikationer, hvor brugeren kan vælge en værdi fra en slider for at ændre på applikationen.

For at lave en slider kan du bruge følgende kode:

```
import streamlit as st
st.slider("Vælg en værdi", 0, 100)
```

Dette vil lave en slider med teksten "Vælg en værdi" og med værdier fra 0 til 100. Hvis du vælger en af værdierne vil den skifte farve.

## Text inputs

En anden meget brugt widget er text inputs. Text inputs kan bruges til at lave interaktive applikationer, hvor brugeren kan skrive en tekst for at ændre på applikationen.

For at lave en text input kan du bruge følgende kode:

```
import streamlit as st
st.text_input("Skriv en tekst")
```

Dette vil lave en text input med teksten "Skriv en tekst". Hvis du skriver en tekst i text input feltet vil den skifte farve.



## Date inputs

En anden meget brugt widget er date inputs. Date inputs kan bruges til at lave interaktive applikationer, hvor brugeren kan vælge en dato for at ændre på applikationen.

For at lave en date input kan du bruge følgende kode:

```
import streamlit as st
st.date_input("Vælg en dato")
```

Dette vil lave en date input med teksten "Vælg en dato". Hvis du vælger en dato i date input feltet vil den skifte farve.

## File uploads

En anden meget brugt widget er file uploads. File uploads kan bruges til at lave interaktive applikationer, hvor brugeren kan uploade en fil for at ændre på applikationen.

For at lave en file upload kan du bruge følgende kode:

```
import streamlit as st
st.file_uploader("Upload en fil")
```

Dette vil lave en file upload med teksten "Upload en fil". Hvis du uploader en fil vil den skifte farve.

## Plot widgets

Streamlit har indbygget støtte til mange forskellige plot biblioteker, herunder Matplotlib, Plotly og Altair. Hvis du bruger et af disse biblioteker kan du bruge følgende kode til at vise et plot:

```
import streamlit as st
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 10, 100)
y = np.sin(x)
plt.plot(x, y)
st.pyplot()
```

Dette vil lave et plot med Matplotlib. Hvis du bruger et andet plot bibliotek skal du bruge den tilsvarende funktion til at vise et plot.

## Map widgets

Streamlit har indbygget støtte til mange forskellige kort biblioteker, herunder Mapbox og Folium. Hvis du bruger et af disse biblioteker kan du bruge følgende kode til at vise et kort:

```
import streamlit as st
import folium
m = folium.Map(location=[55.676098, 12.568337])
st.map(m)
```

# Layout i Streamlit

Streamlit har indbygget støtte til layout. Du kan bruge layout til at ændre på udseendet af din applikation.

I det følgende vil vi se på nogle af de mest brugte layout funktioner:

- \* Kolonner
- \* Sidebars
- \* Ekspanderbare sektioner
- \* Titel
- \* Tekst
- \* Markdown

# Kolonner

En kolonne er en måde at opdele din applikation i flere kolonner. Du kan bruge kolonner til at vise flere widgets ved siden af hinanden. For at lave kolonner kan du bruge følgende kode:

```
import streamlit as st
col1, col2 = st.beta_columns(2)
col1.button("Klik her")
col2.button("Klik her")
```

Dette vil lave to kolonner med en knap i hver kolonne. Hvis du trykker på en af knapperne vil den skifte farve.

# Sidebars

En sidebar er en måde at opdele din applikation i en sidebar og en hoveddel. Du kan bruge sidebars til at vise widgets i en sidebar. For at lave en sidebar kan du bruge følgende kode:

```
import streamlit as st
st.sidebar.button("Klik her")
```

Dette vil lave en sidebar med en knap. Hvis du trykker på knappen vil den skifte farve.

## Ekspanderbare sektioner

En ekspanderbar sektion er en måde at opdele din applikation i en ekspanderbar sektion og en hoveddel. Du kan bruge ekspanderbare sektioner til at vise widgets i en ekspanderbar sektion. For at lave en ekspanderbar sektion kan du bruge følgende kode:

```
import streamlit as st
with st.beta_expander("Klik her"):
    st.button("Klik her")
```

Dette vil lave en ekspanderbar sektion med en knap. Hvis du trykker på knappen vil den skifte farve.



## Radio knapper i Streamlit

Streamlit har indbygget støtte til radio knapper. Du kan bruge dette til at lave radio knapper i din applikation. For at lave radio knapper kan du bruge følgende kode:

```
import streamlit as st
st.write("## Radio knapper")
option = st.radio("Vælg en værdi", ["Værdi 1", "Værdi 2", "Værdi 3"])
st.write(f"Du valgte {option}")
```

Dette vil lave radio knapper med teksten "Vælg en værdi" og med tre værdier: "Værdi 1", "Værdi 2" og "Værdi 3". Hvis du vælger en af værdierne vil den skifte farve.

# Markdown

Markdown er en måde at formatere tekst i din applikation. Du kan bruge markdown til at vise tekst i din applikation. For at lave markdown kan du bruge følgende kode:

```
import streamlit as st
st.markdown("## Overskrift")
```

Dette vil lave en overskrift med teksten "Overskrift".

## En bmi-beregner i Streamlit

I det følgende vil vi se på et eksempel på en Streamlit applikation. Denne applikation viser en kort bmi-beregner. Denne bmi-beregner kan bruges til at beregne bmi ud fra højde og vægt.

```
import streamlit as st
st.title("BMI-beregner")
st.markdown("## Indtast din højde og vægt")
height = st.slider("Højde (cm)", 100, 250)
weight = st.slider("Vægt (kg)", 40, 200)
bmi = weight / (height / 100) ** 2
st.markdown(f"## Din BMI er {bmi:.2f}")
```

## Grafer af funktioner i Streamlit

I det følgende vil vi se på et eksempel på en Streamlit applikation. Denne applikation viser grafer af forskellige funktioner. Denne applikation kan bruges til at vise grafer af forskellige funktioner.

```
import streamlit as st
import matplotlib.pyplot as plt
import numpy as np
st.title("Grafer af funktioner")
st.markdown("## Indtast en funktion")
function = st.text_input("Funktion")
x = np.linspace(-10, 10, 100)
try:
    y = eval(function)
    plt.plot(x, y)
    st.pyplot()
except:
    pass
```

## Kort applikation i Streamlit

I det følgende vil vi se på et eksempel på en Streamlit applikation. Denne applikation viser et kort over København. Denne applikation kan bruges til at vise et kort over København.

```
import streamlit as st
import folium
st.title("Kort over København")
m = folium.Map(location=[55.676098, 12.568337])
st.map(m)
```

## Gemme data i Streamlit

Streamlit har indbygget støtte til at gemme data. Du kan bruge dette til at gemme data fra din applikation. For at gemme data kan du bruge følgende kode:

```
import streamlit as st
st.write("## Gem data")
data = st.text_input("Skriv noget data")
st.session_state["data"] = data
```

Dette vil lave en text input med teksten "Skriv noget data". Hvis du skriver noget data i text input feltet vil det blive gemt i session state.

## Hente data i Streamlit

Streamlit har indbygget støtte til at hente data. Du kan bruge dette til at hente data fra din applikation. For at hente data kan du bruge følgende kode:

```
import streamlit as st
st.write("## Hent data")
data = st.session_state["data"]
st.write(data)
```

Dette vil hente data fra session state og vise det.

## Vise billeder i Streamlit

Streamlit har indbygget støtte til at vise billeder. Du kan bruge dette til at vise billeder i din applikation. For at vise et billede kan du bruge følgende kode:

```
import streamlit as st
st.write("## Vis billede")
st.image("billede.jpg")
```

Dette vil vise et billede med filnavnet "billede.jpg". Det antages at billedet ligger i samme mappe som din applikation.



## Vise data fra danmarks statistik i Streamlit

I det følgende vil vi se på et eksempel på en Streamlit applikation. Denne applikation viser data fra Danmarks Statistik. Denne applikation kan bruges til at vise data fra Danmarks Statistik.

```
import streamlit as st
import pandas as pd
st.title("Data fra Danmarks Statistik")
st.markdown("## Indtast et tabelnummer")
table = st.text_input("Tabelnummer")
try:
    df = pd.read_csv(f"https://api.statbank.dk/v1/data/{table}")
    st.write(df)
except:
    pass
```

## Vise data fra en database i Streamlit

I det følgende vil vi se på et eksempel på en Streamlit applikation. Denne applikation viser data fra en database. Denne applikation kan bruges til at vise data fra en database.

```
import streamlit as st
import pandas as pd
import sqlite3
st.title("Data fra en database")
st.markdown("## Indtast et tabelnavn")
table = st.text_input("Tabelnavn")
try:
    conn = sqlite3.connect("data.db")
    df = pd.read_sql(f"SELECT * FROM {table}", conn)
    st.write(df)
except:
    pass
```

# Publicere streamlit app på nettet med Heroku I

Hvis du vil publicere din streamlit app på nettet kan du bruge Heroku. For at bruge Heroku skal du først oprette en konto på Heroku. Derefter kan du publicere din app ved at køre følgende kommando i en terminal:

```
heroku login
```

Dette vil åbne en browser, hvor du kan logge ind med din konto. Når du har logget ind kan du publicere din app ved at køre følgende kommando i en terminal:

```
heroku create
```

Dette vil publicere din app på Heroku. Du kan nu dele et link til din app med andre.

# Publicere streamlit app på nettet med streamlit share I

Hvis du vil publicere din streamlit app på nettet kan du bruge streamlit share. For at bruge streamlit share skal du først oprette en konto på streamlit share. Derefter kan du publicere din app ved at køre følgende kommando i en terminal:

```
streamlit login
```

## Publicere streamlit app på nettet med streamlit share II

Dette vil åbne en browser, hvor du kan logge ind med din konto. Når du har logget ind kan du publicere din app ved at køre følgende kommando i en terminal:

```
streamlit deploy app.py
```

Dette vil publicere din app på streamlit share. Du kan nu dele et link til din app med andre.

# Hvis du vil vide mere

Hvis du vil vide mere om Streamlit kan du læse mere på Streamlits hjemmeside: <https://streamlit.io/>

# Forslag til opgaver

- ▶ Lav promilleberegner i Streamlit
- ▶ Lav en CPR-validering i Streamlit
- ▶ Lav en valutaomregner i Streamlit
- ▶ Lav en randomiseret quote/citat generator i Streamlit
- ▶ Lav en lommeregner i Streamlit
- ▶ Lav et program der viser billeder i Streamlit
- ▶ Lav en todo-liste i Streamlit
- ▶

## Flere forslag til opgaver

- ▶ Lav en bmi-beregner i Streamlit
- ▶ Lav grafer af funktioner i Streamlit
- ▶ Lav en kort applikation i Streamlit
- ▶ Lav en applikation der viser data fra Danmarks Statistik i Streamlit
- ▶ Lav en kalorieberegner i Streamlit
- ▶ Lav en applikation der gemmer data i Streamlit
- ▶ Lav en applikation der henter data i Streamlit
- ▶ Lav en applikation der viser billeder i Streamlit



## Endnu flere forslag til opgaver

- ▶ Ordoptælling: Opret en applikation, der tager en tekst som input og viser antallet af ord, sætninger og tegn i teksten.
- ▶ Farvepaletgenerator: Opret en applikation, der genererer en tilfældig farvepalet baseret på brugerens valg af farveskala og antal farver.
- ▶ Spørgeundersøgelse: Lav en enkel spørgeundersøgelse, hvor brugere kan besvare nogle spørgsmål ved hjælp af forskellige input-widgets som tekstinput, radioknapper eller flervalgsfelter.
- ▶ Portfolio Tracker: Lav en enkel applikation, hvor brugere kan indtaste aktier eller kryptovalutaer, de ejer, og applikationen viser den aktuelle værdi baseret på realtime-markedsdata.
- ▶ Tipkalkulator: Lav en kalkulator, der beregner et anbefalet tipbeløb baseret på den samlede regning og en procentdel.