AI og data mini projekt

Indhold:

1. Lektion 2: Databaser
2. Lektion 3: Dataoprensning og præprocessering
3. Lektion 5: Intro til Pandas og Datamanipulering
4. Lektion 7: Interaktiv datavisualisering

Student: Jacob Selbo  
Studie nummer: 20233533

GitHub: <https://github.com/jacobselbok/ai-og-data-mini-projekt>

Herinde er alt kode der er brugt til at løse opgaverne + eventuelle billeder og filer.

# Lektion 2: Databaser

1. Skriv et Python script der opretter en SQLite database ‘school.db’ (se <https://realpython.com/python-sql-libraries/> )
2. Udvid scriptet til at oprette to tabeller ‘Students’ og ‘Courses’.
   1. Students skal indeholde student\_id, name, og major
   2. Courses skal indeholde course\_id, course\_name, og instructor
3. Indsæt mindst 5 records i hver tabel.
4. Opret en tredje tabel ‘Enrollments’ der opretter relationer mellem Students og Course.
   1. Enrollments skal indeholde enrollment\_id, student\_id, og course\_id
5. Lav forespørgsler (queries) der
   1. Vælger alle kurser som en specific studerende er tilmeldt
   2. Vælger alle studerende der er tilmeldt et specifikt kursus.
6. Opgave 1:

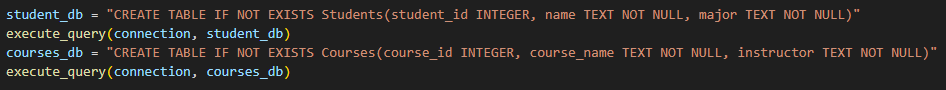
For at oprette en SQLite database, impoter jeg sqlite3. Herefter opretter jeg en connection object til en ”path” der kaldes school.db:  


1. Opgave 2:

Jeg executer ”CREATE TABEL IF NOT EXISTS” på cursor objectet.



Querierne laver 2 tabeller i school.db databasen, kaldet henholdsvis ”Students” og ”Courses”. Begge tabeller får deres egne id for at de nemmere kan samles senere.



1. Opgave 3:

Jeg executer INSERT INTO på tabellen og dens 3 columns og angiver 5 unikke studenter. Det samme bliver gjort for Courses.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse

1. Opgave 4:

Jeg laver en ny tabel til Enrollments på samme måde som ved Students og Courses i opgave 2. Herefter tilføjer jeg 10 rækker til Enrollments hvor student\_id og course\_id kan relateres til hinanden.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

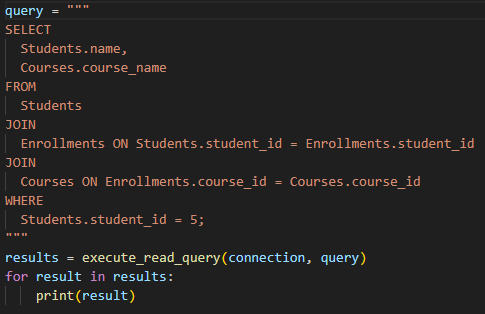
Automatisk genereret beskrivelse

1. Opgrave 5:

Her bruges .fetchall() funktionen til at returnere en liste af tuples der indeholder sammenhængen mellem de valgte records.



Jeg SELECTer først at jeg vil have student navnene og course navnene. Herefter JOINer jeg Students tabellen til Enrollments tabellen via student\_id. Så JOINer jeg det resulterende tabel sammen med courses tabellen via course\_id. Så bruger jeg WHERE til at specificere at jeg, i den resulterende tabel, kun vil have information fra den student med student\_id = 5. Til sidst printer jeg de tupler som .fetchall() returnerede.



Den samme fremgangsmåde blev brugt til at finde alle studenter der er i en specifik course. Herunder er de returnerede tupler:

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, typografi

Automatisk genereret beskrivelse

# Lektion 3: Dataoprensning og præprocessering

Opgave1:

* Skriv et Python script der tager et rent gråskala billede som input, og tilføjer salt-pepper støj.
* Efter tilføjelse af støj skal:
  + X % af pixels skal være støj (heraf halvdelen hvide pixels og den anden halvdel sorte pixels).
* Implementer middelværdisfiltrering med en kernel på 3x3, og anvend det på det støjfulde billede.
* Implementer median filtrering og anvend det på det støjfulde billede.

Opgave 2:

* Skriv et Python script der tager et rent gråskala billede som input, og tilføjer Gaussisk støj.
* Scriptet kan I antage at:
  + \mu = 0
  + \sigma er en brugerparameter.
* Implementer middelværdisfiltrering med en kernel på 3x3, og anvend det på det støjfulde billede.
* Implementer median filtrering og anvend det på det støjfulde billede.

Opgave 3:

* Følg følgende guide (delen der vedrører lineær PCA), og forstå hvordan PCA kan anvendes til støjreduktion.
* <https://scikit-learn.org/stable/auto_examples/applications/plot_digits_denoising.html>
* Eksperimentér med forskellige antal af komponenter I rekonstruktionen.

Opgave 1:

Jeg starter med at indlæse et billede med cv2 og ændre det til gray scale:



Herefter laver jeg en funktion der først finder det totale mænde af pixel. Så sætter den 10% tilfældigt udvalgte pixels, af alle pixels, og sætter dem til hvid og gør det samme med 10% til sort.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, nummer/tal

Automatisk genereret beskrivelse

Et billede, der indeholder pattedyr, stort kattedyr, tiger, Amurtiger

Automatisk genereret beskrivelse

Så tilføjer jeg middelværdisfiltrering med .blur funktion og median filtrering med .medianblue funktionen:

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, linje/række

Automatisk genereret beskrivelse

Et billede, der indeholder pattedyr, stort kattedyr, tiger, Store kattedyr

Automatisk genereret beskrivelse

Et billede, der indeholder pattedyr, stort kattedyr, tiger, Store kattedyr

Automatisk genereret beskrivelse

Opgave 2:

Indlæsning af billedet skete på samme måde som opgave 1.

Herefter blev der lavet en ny funktion hvor der først blev udregnet gaussian støj for alle pixels og så blev det lagt til gray scale billedet:



Herefter blev np.clip brugt til at sørge for at alle tal er imellem 0 og 255:



Til sidst blev dataframe tretuneret med .astype(np.uint8), hvilket afrunder eventuelle floats og ændre tallene til integers:



Her kan sigma ændres som brugerparameter:



Middelværdisfiltrering og median filtrering blev implementeret på samme måde som opgave 1. Billederne kan ses i github.

Opgave 3:

For de nedenstående billeder har n\_components stået på henholdsvis 10, 32 og 80:



Et billede, der indeholder tekst, typografi, Font/skrifttype, sort-hvid

Automatisk genereret beskrivelse Et billede, der indeholder tekst, typografi, Font/skrifttype, sort-hvid

Automatisk genereret beskrivelse Et billede, der indeholder tekst, sort-hvid, typografi, Font/skrifttype

Automatisk genereret beskrivelse

# Lektion 5: Intro til Pandas og Datamanipulering

1. Følg tutorial (under 'Forberedelse') grundigt på egen hånd. Opret en Google Colab notebook (https://colab.research.google.com/), og implementer og afprøv alle eksempler på egen hånd.
2. Anvend metoderne på datasættet fra kursusgang 4:
   * indlæs datasættet i en DataFrame
   * fjern manglende værdier
   * plot numeriske variabler i datasættet (https://python.pages.doc.ic.ac.uk/lessons/pandas/06-methods/05-plot.html)

1. Opgave 1:

Jeg lavede en fil kaldet ”pandas\_intro.py” i lektion 5 mappen hvor jeg implementerede alle funktionerne jeg testede fra Pandas tutorialen.

1. Opgave 2:

Jeg indlæser datasættet i Pandas med følgende funktion:



Jeg printer .shape på dataframe før og efter jeg bruger .dropna funktionen:

Et billede, der indeholder Font/skrifttype, tekst, skærmbillede, Grafik

Automatisk genereret beskrivelse



Jeg bruger .info til at finde hvilke columns der har numeriske værdier:



Herefter plotter jeg for ”Size(L)” og ”Color”:

­­Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, design

Automatisk genereret beskrivelse­

Et billede, der indeholder tekst, skærmbillede, display/skærm/fremvisning, Rektangel

Automatisk genereret beskrivelse

Et billede, der indeholder tekst, skærmbillede, diagram, display/skærm/fremvisning

Automatisk genereret beskrivelse

# Lektion 7: Interaktiv datavisualisering

* + Følg tutorial (https://youtu.be/yVxnQQpBg-Q?si=mAvevB6T-yvkd9X0)\* og lav jeres eget scatter plot (kode)
  + Følg tutorials og lave jeres eget interaktive scatter plot
    1. tutorial 1 (https://youtu.be/j4VI47GlmGA?si=C9vZ3vwjEVear8ap)\* (kode)
    2. tutorial 2 (https://youtu.be/kyWZf3sNhtg?si=3ocH-ApWBpnVYZZ5)\* (kode)
  + Opdatér jeres kode til at bruge et valgfrit datasæt med features (f.eks. fra jeres projekt, eller AutoMPG)

Opgave 1:

Der startes med at indlæse csv med pandas:



Herefter bruges .info() til at få informationer om csv filen:



Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Der laves et scatterplot udfra ”sepal\_length” og ”sepal\_width” med target på ”species” ved hjælp af seaborn og matplotlib.pyplot:



Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Et billede, der indeholder tekst, skærmbillede, diagram

Automatisk genereret beskrivelse

Opgave 2:

Der laves et interaktiv scatterplot, med hjælp af plotly.express:



Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Et billede, der indeholder tekst, skærmbillede, diagram, Kurve

Automatisk genereret beskrivelse

På dette scatterplot kan der zoomes ind og ud, man kan ”pan”, plot kan downloades som png osv, hvilket gør det til et interaktivt scatterplot.