Intro til python og pandas

Afleverings beskrivelse:

I skal i denne uge aflevere jeres kode i form af et link til jeres git repository. Inde i jeres git repository skal der ligge et README.md fil, og en mappe for hver opgave, som heder f.eks. opgave_1 eller lignende. README filen skal indeholde, en kort beskrivelse af hvordan man kører jeres kode, samt en kort beskrivelse af hvilke områder i syntes der var sværest, eller hvilke områder i gerne specielt vil have feedback på. Hvis I syntes at det var lige svært, eller at der ikke var noget I specifikt vil have feedback på, skal i kort beskrive hvordan i ville have arbejdet videre med opgaven.

Opgave 1: Intro til Python

Begynder

Det er vigtigt at kunne udføre grundlæggende dataanalyse, fordi det gør det muligt at træffe informerede beslutninger baseret på data. I en verden, hvor der produceres store mængder data dagligt, kan evnen til at analysere og forstå data give afgørende indsigt i trends, mønstre og sammenhænge, som ellers ville være skjulte.

Til denne opgave er der tilknyttet en liste af navne, som du skal sortere og derefter udskrive. Sorteringen vil være alfabetisk, og derefter efter længde. Til sidst skal du lave et Python dictionary over forekomsten af bogstaver i alle navne

Tilgang til Opgaven

- **Sortering af Navne**: Start med at oprette en liste af navne (eventuelt kopier fra listen her) Brug **Python's indbyggede sortering** med en brugerdefineret nøgle for at sortere navnene først alfabetisk og derefter efter længde.
- **Optælling af Bogstaver**: Gå igennem hvert navn i den sorterede liste og tæl forekomsten af hvert bogstav. Gem disse tællinger i et **dictionary**, hvor nøglen er bogstavet og værdien er antallet af forekomster.

Hints

- Brug **sorted()** funktionen med key parameteren for at sortere listen som ønsket.
- For at tælle bogstaver, kan du bruge en for løkke til at iterere igennem hvert navn og en indre for løkke til at iterere igennem hvert bogstav i navnet.
- Overvej at bruge **defaultdict** fra collections modulet for at forenkle optællingen af bogstaver.

Step by step anbefaling

- Start med at lave en liste med navne (eventuelt kopier fra listen her)
- Brug listens indbyggede **sort()** funktion til at sorter alfabetisk
- Print de første 10 elementer ud og se om det er som du forventer
- Brug listens indbyggede funktion **sort()** til at sortere efter længde (hint: parameteren til sort tager en ”key” som er en funktion den skal bruge til at vurderer sorteringen efter)
- Print de 10 første elementer af listen ud og se om det er hvad du forventede
- Lav et dictionary med alle bogstaver som key, og antal som value. Overvej om du vil tage højde for store og små bogstaver
- Lav to for each loops, den første over alle elementer i listen den anden efter hvert ord i elementet
- Slå op i dictionary ved brug af key og inkrementere den entry der tilsvarer det nyværende bogstav

Ressourcer

- Real Python - Python Data Structures Tutorial: https://realpython.com/python-data-structures
- Pluralsight: Core Python: Getting Started

Avanceret udvidelse

Hvis du er hurtigt færdig med denne opgave så kan du få yderligere udfordringer, enten med følgende udvidelse af opgaven eller ved at fortsætte med ugeopgave 2.

Opgavebeskrivelse

Avanceret dataanalyse og visualisering hjælper med at forstå komplekse datasæt på en intuitiv og dybere måde. Avanceret analyse kan identificere skjulte mønstre, tendenser og sammenhænge, som grundlæggende metoder ikke afslører. Ved at visualisere disse data kan man let kommunikere komplekse indsigter, så de bliver forståelige for beslutningstagere, der måske ikke har teknisk baggrund.

Byg videre på den oprindelige opgave ved at udføre en detaljeret analyse af navnedataene. Efter at have talt forekomsten af hvert bogstav i alle navne, udfør følgende:

- **Frekvensanalyse**: Beregn og visualiser frekvensen af hvert bogstav anvendt i navnelisten. Overvej at bruge biblioteket **matplotlib** eller **seaborn** til at lave et histogram- eller søjlediagram, der viser de mest almindelige bogstaver
- **Ordsky**: Generer en ordsky (word cloud) baseret på hyppigheden af hvert bogstav. Jo oftere et bogstav forekommer, desto større skal det vises i ordskyen. Dette kan gøres med **wordcloud biblioteket i Python**.
- **Navnelængde Analyse**: Analyser fordelingen af navnelængder i dit dataset. Beregn **gennemsnitlig** og **median** for navnelængderne, og visualiser dataen med passende plots.

Yderligere Overvejelser

Overvej at filtrere navnelisten for duplikater før analysen, og gentag kørslen og sammenlign resultaterne.

Ressourcer for Udvidet Opgave

- Matplotlib Dokumentation: For grundlæggende til avancerede plots: https://matplotlib.org
- Seaborn Dokumentation: For statistiske data visualiseringer: https://seaborn.pydata.org
- ~~WordCloud Dokumentation: For at generere ordsky billeder: https://github.com/amueller/word_cloud~~

Opgave 2: Logfil analyse

Formål

Udvikl et script, der læser en logfil, filtrerer bestemte typer af logmeddelelser (som f.eks. *ERROR* og *WARNING*), og opsummere disse i en ny fil. Automatisering af denne type loganalyse gør det muligt hurtigt at identificere og reagere på systemfejl og advarsler uden manuel gennemgang af store datamængder. Ved at opsummere kritiske logmeddelelser hjælper scriptet med at forbedre systemets pålidelighed og reducere nedetid.

Data til opgave:

Fil: app_log.txt - *Du må meget gerne teste med andre filer.*

Opgavebeskrivelse

Opret et script, der læser indholdet fra en logfil (app_log.txt) og filtrerer bestemte typer af logmeddelelser, som ERROR og WARNING.

Gem de filtrerede logmeddelelser i en ny fil, så kritiske hændelser er lette at overskue.

Færdigheder der udvikles

- **Filhåndtering**: Åbning, læsning og skrivning af filer.
- **Tekstbehandling**: Brug af Python til at finde, filtrere og opsummere tekstinformationer.

Step by step anbefaling:

- Start med at importerer os.path.join (Biblioteket hjælper med at sætte den korrekte path deliminator som er forskellig fra os til os)

- Open en ny fil for hver besked type du har vil haveseperaret. Filen behøver ikke at eksisterer i forvejen python kan lave den for dig
- Open filen med logbeskeder. Eventuelt skriv indeholdet ud i terminalen for at se om den har åbnet filen korrekt
- Gå hver linie af filen igennem for at se hvilken besked type det er, og lig linien i den tilsvarende fil
- Luk alle åbnede filer
- Check om resultatet er hvad du forventer

Udfordring

Sørg for korrekt filhåndtering (inkl. åbning og lukning af filer), og implementer en simpel, men effektiv metode til at filtrere logmeddelelser. Du er velkommen til at teste dit script med andre logfiler for at sikre fleksibiliteten i din løsning.

Opgave 3: Kort fejl håndtering

Formål

Udvikl et script, der kan migrere data fra en kildefil til en destinationsfil, samtidig med at det håndterer potentielle fejl som f.eks. manglende filer, formateringsfejl eller skrivebeskyttede filer. Et script med korrekt fejl- og undtagelseshåndtering hjælper med at sikre en sikker og pålidelig datamigrering, reducerer risikoen for datatab og sikrer, at brugeren får klare instruktioner, hvis der opstår problemer.

Data til opgave

Fil: source_data.csv - *Du må meget gerne teste med andre filer.*

Opgavebeskrivelse

- Opret et script, der læser data fra en kildefil (`source_data.csv`) og skriver dem til en destinationsfil.
- Scriptet skal håndtere forskellige typer fejl, som f.eks.:
    - **Manglende filer**: Hvad skal der ske, hvis kildefilen ikke eksisterer?
    - **Formateringsfejl**: Håndter situationer, hvor dataformatet er forkert eller uventet.
    - **Skrivebeskyttelse**: Håndter fejl, hvis destinationsfilen ikke kan skrives til.

Færdigheder der udvikles

- **Avanceret Filhåndtering**: Læsning fra og skrivning til filer med en struktureret tilgang.

- **Undtagelseshåndtering**: Brug af try-except blokke til at håndtere potentielle fejl og sikre scriptets robusthed.
- **Brugerdefinerede fejlmeddelelser**: Udvikling af klare fejlmeddelelser, der kan hjælpe brugeren med at forstå og løse problemer.

Udfordring

Implementer en robust undtagelseshåndtering, der kan håndtere uventede situationer effektivt og uden at scriptet går ned. Det er vigtigt at scriptet tydeligt informerer brugeren om eventuelle fejl og hvordan de kan afhjælpes.

Opgave 4: Pandas

Formål:

Formålet med denne opgave er at få erfaring/blive bekendt med data værktøjet panda. Panda opdeler data i hvad den kalder et dataframe. I skal derfor læse data ind i et dataframe, og eksperimentere med de forskellige funktioner

Data til opgave

Fil: DKHousingPricesSample100k.csv

Opgave beskrivelse

1. Læs filen ind
    a. Brug read_csv
    b. Print de første 10 data ud
2. Group by bruges til at sortere colouner efter grupper
    a. Lav et group by på region
    b. Lav et gennemsnit over purchases
    c. Print resultatet ud
3. Mere advanceret Group by
    a. Lav group by på house_type
    b. Lav et group by på region
    c. Lav et gennemsnit over purchases
    d. Plot det vedhjælp af matplotlib
4. Eksperimenter med at group by og avarage over forskellige coulumns og plot resulteter

Ressourcer

- Real Python - Functional Programming in Python: https://realpython.com/courses/functional-programming-python
- Faker: https://faker.readthedocs.io/en/master

- Pluralsight: Functional Programming in Python 3