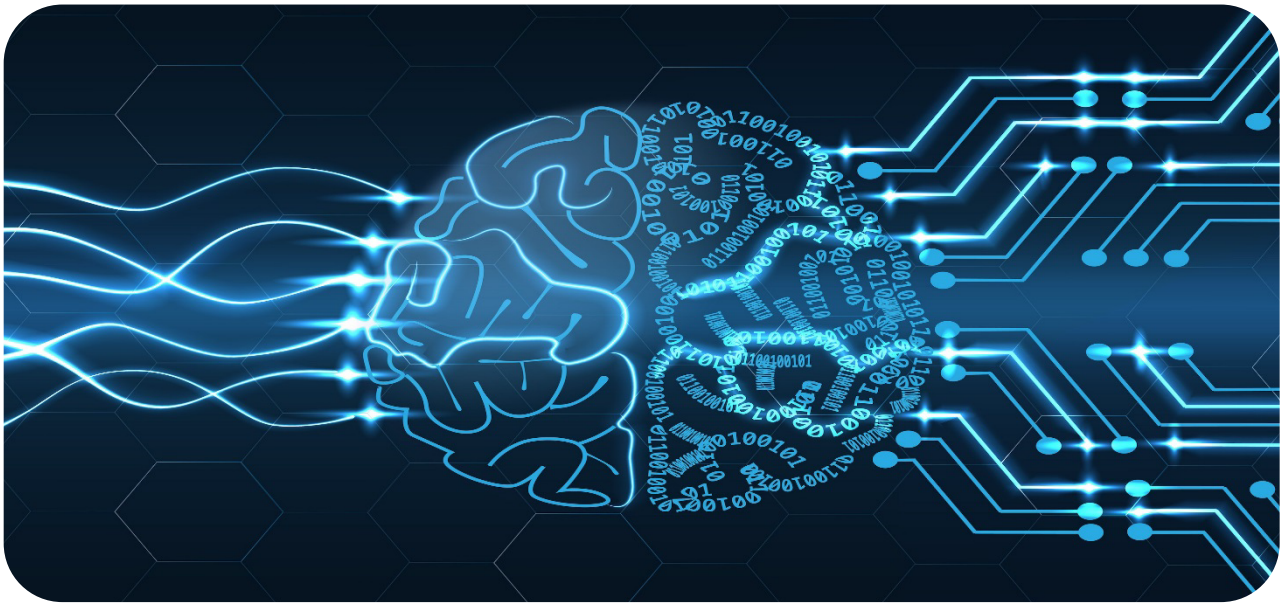
Obligatorisk læringsaktivitet i Flow 3, Dataanalyse



Afleveringsfrist:

Torsdag den 28. november 2024, kl. 18

Udarbejdet af:

Thorbjørn Wulf, adjunkt Cphbusiness

Indhold

Opgave 1 – Webscrape 3
Opgave 1.1 – Hente data fra Bilbasen 3
Opgave 1.2 – Rense data 3
Opgave 1.3 – Hente nye data - simuleret 3
Opgave 1.4 – Hente tyske data 3
Opgave 2 – SQL 4
Opgave 2.1 – Oprette skemaet for bilbasen 4
Opgave 2.2 – Gemme bilerne i database 4
Opgave 2.3 – Opdatere databasen ud fra den simulerede kørsel 4
Opgave 2.4 – Scrape & SQL med Miljødata. 4
Opgave 3 – Analyse af logfiler 5
Opgave 3.1 – Rapport fra en webserver 5

Opgave 1 – Webscrape

I alle besvarelser er det vigtigt at I dokumenterer processen med skærmdumps af jeres Trello-board samt link til github med jeres kode.

Bilerne som I henter i opgave 1 skal bruges i opgave 2 når de skal gemmes i en database.

Opgave 1.1 – Hente data fra Bilbasen

I skal hente udvalgte biler fra bilbasen.dk. Hent så mange som muligt i en "lukket" serie. I kan f.eks vælge VW, model Up, eldrevne. Koordiner med de andre grupper så I henter forskellige biler. Når I henter data, skal I ud over de oplagte data også huske at gemme linket så man kan hente flere data på hver bil. I skal også hente forhandleren – både hans id samt firmanavn, adresse samt cvr-nummer. Det kan involvere manuelt arbejde.

Opgave 1.2 – Rense data

Salgsteksten fra bilen indeholder en masse overflødige tegn. Sørg for at få renset teksten så der kun er almindelige karakterer og tegn (punktum, komma) tilbage. Sørg også for at "newline" erstattes af ". " og at mange mellemrum erstattes med ét mellemrum.

Opgave 1.3 – Hente nye data - simuleret

I skal nu lave en simuleret hentning af biler, hvor I skal tage udgangspunkt i de biler I hentede i 1.1. Den simulerede kørsel skal have en *scrapedate* som ligger én dag senere end 1.1 og den skal have 2 rækker med nye biler, 3 rækker med ændrede priser og så skal der mangle 5 rækker fra den oprindelige testkørsel så I kan simulere, at bilerne er blevet solgt.

Opgave 1.4 – Hente tyske data.

I skal nu hente samme bilsegment fra Tyskland, så man kan afgøre om det kan betale sig at køre til Tyskland og hente den. Jeg har kigget på 12gebrauchtwagen.de men det kan være der findes andre.

Opgave 2 – SQL

Opgave 2.1 – Oprette skemaet for bilbasen

For at kunne gemme bilerne i databasen er det vigtigt at oprette et skema, hvor man har invariante entiteter organiseret i tabeller med fremmed-nøgler så de kan linkes sammen, samt variende data i tidsserieagtige tabeller, hvor man linker til den sidst-forekommende observation. Jeres design skal beskrives i et ER-diagram og jeres DDL-statements skal ligge på github med filextension .sql.

Opgave 2.2 – Gemme bilerne i database

I skal nu gemme jeres første scrape-resultat i jeres database. I *kan* lade R-driveren gøre arbejdet med at oprette skemaet men I skal sørge for at der er plads til det, som R-driveren sætter på data-typerne. I skal desuden definere *carid* som primær nøgle. Jeres INSERT-statements skal også ligge på github.

Opgave 2.3 – Opdatere databasen ud fra den simulerede kørsel

Hvis I har fået lavet et korrekt skema – altså at I kan versionere prisen vha en pris-tabel – burde I kunne opdatere databasen med jeres simulerede nye scraping, hvor I altså opretter/ændrer en record med pris, dato og *carid* så man joine de to tabeller på carid.

Det gøres bedst ved at I laver et script eller en funktion som henter den forrige kørsel fra databasen (altså jeres data fra opgave 2.2) og sammenligner med den simulerede kørsel (altså kørsel-II). I skal finde

a) Nye records
b) Ændrede records (på prisen)
c) Missing records (solgte biler)

Og opdatere databasen efterfølgende. De solgte biler skal ikke slette men markeres som TRUE i *sold*.

Opgave 2.4 – Scrape & SQL med Miljødata.

I skal kigge på https://envs.au.dk/om-instituttet-1/faglige-omraader/luftforurening-udledninger-og-effekter/data-om-luftkvalitet/aktuelle-maalinger/tabeller og hente links fra H.C.Andersens Boulevard, Anholt, Banegårdsgade i Århus og Risø.

I skal lave et R-script som kan hente data fra de fire lokationer og gemme dem i fire tabeller. I skal derpå vente en dag og så hente data igen og opdatere tabellerne med nye data. I forbindelse med næste OLA skal scriptet kunne køre på en linux-server én gang i døgnet. Så jeres script skal helst kunne eksekveres i terminalen uafhængigt af Rstudio med målestationen som argument.

Opgave 3 – Analyse af logfiler

Opgave 3.1 – Rapport fra en webserver

I skal åbne logfilerne og lave en rapport over aktiviteten på webserveren.

Rapporten skal indeholde en optælling af aktive ip-adresser pr døgn. Plot med de mest aktive. whois-info på den mest aktive. Gruppering på 404, herunder en beskrivelse af "mistænksomme" requests.

Det værste der kan ske, er at en mistænksom request returnerer 200. Overvej hvordan man kan fange dem.