Engineer

Datasett

en samling data publisert som én enhet

Hva det er

Et datasett er en avgrenset mengde data som folk behandler som én ting. Det har et navn, en kilde, en periode det dekker og en form som alle postene deler, og de fire opplysningene sier hvor det begynner og slutter. En måneds måleravlesninger i én fil er et datasett, og det er også en nattlig eksport av tre tabeller som hører sammen. På engelsk skrives ordet både dataset og data set, og begge formene dukker opp i norske team. Ordet sier ingenting om størrelse eller format, bare at dataene inni hører sammen.

Sidene på dette nettstedet bruker ordet i denne praktiske betydningen. Én omtaler et historisk datasett i JSONL-format og datasett på 1 dag og 30 dager brukt til ytelsesmålinger, en annen kuraterte datasett og tabeller eksportert for innlasting et annet sted, en tredje enorme datasett fra smartmålere. Hver gang trekker ordet en strek rundt dataene før noe blir gjort med dem.

W3Cs Data Catalog Vocabulary, DCAT, gir ordet en formell ramme. Versjon 3 ble W3C-anbefaling 22. august 2024 og definerer et datasett som en samling data, publisert eller kuratert av én aktør og tilgjengelig for tilgang eller nedlasting i én eller flere serialiseringer eller formater. Begrepet er bredt med vilje: tall, tekst, bilder og lyd teller alle sammen. Deretter skiller DCAT mellom tre ting som dagligtalen blander sammen:

- **Dataset**: dataene som idé, f.eks. timeforbruk per sone for 2024.
- **Distribution**: én tilgjengelig form av dem, f.eks. en CSV-fil du kan laste ned. Ett datasett kan ha flere.
- **Dataset series**: datasett som publiseres hver for seg, men har nok til felles til å grupperes, f.eks. én fil per måned.

To filer i ulike formater kan altså være ett datasett, og to månedsfiler under samme navn kan være to medlemmer av en serie.

I Google BigQuery har ordet en annen og smalere betydning. Der er et datasett en beholder for tabeller, visninger og andre ressurser, og det ligger inne i et prosjekt. En tabell må høre til et datasett, så datasettet opprettes før noen data lastes inn. Det adresseres som `project.dataset` i GoogleSQL og som `project:dataset` i kommandolinjeverktøyet bq. Sammenhengen avgjør hvilken betydning som gjelder.

Bruk begrepet i en beslutning

Bruk «Hva er et datasett?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Dashbord og rapporter som viser de få tallene en beslutning faktisk hviler på, oppdatert uten at noen må be om det.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Merkevarekommunikasjon ut fra publikums reaksjoner — merkevarekommunikasjon ut fra publikums reaksjoner

Merkevarekommunikasjon ut fra publikums reaksjoner Ta kontakt →

Hvorfor det er viktig

En grense gjør data sammenlignbare. To sett med avlesninger kan bare sammenlignes når begge dekker samme periode, samme steder og samme avstand mellom postene, og det er ved å navngi et datasett at de vilkårene blir festet. DCAT gir avstanden en egen egenskap, temporal resolution: den minste avstanden mellom elementene i datasettet, skrevet som en varighet, f.eks. `P1D` for ett døgn. Et datasett som oppgir oppløsningen sin, forteller leseren om radene er timer, kvarter eller noe uregelmessig før den første summen er regnet ut.

Beskrivelsen må følge dataene. DCAT lister opp hva en distribusjon kan si om seg selv: medietype, format, komprimeringsformat, pakkeformat, størrelse i byte og en kontrollsum. Kontrollsummen lar mottakeren bekrefte at innholdet er uendret underveis. Spesifikasjonens eget eksempel på et komprimeringsformat er gzip. En fil som kommer med disse opplysningene, kan sjekkes og åpnes; en fil som kommer naken, må gjettes på.

Formatet på distribusjonen avgjør hvem som kan lese den. Tre er vanlige:

- **CSV.** RFC 4180 fra 2005 beskriver oppsettet de fleste implementasjoner ser ut til å følge: én post per linje, en valgfri overskriftslinje, doble anførselstegn rundt et felt som inneholder komma, linjeskift eller dobbelt anførselstegn, og et dobbelt anførselstegn inne i et felt skrevet to ganger. Nesten ethvert verktøy åpner det.
- **JSON Lines.** Én gyldig JSON-verdi per linje, i UTF-8, med `\n` som linjeskille, lagret som `.jsonl` eller, gzip-komprimert, som `.jsonl.gz`. Et avsluttende linjeskift anbefales sterkt, men er ikke påkrevd. Programmer leser én post om gangen, og tekstverktøy i Unix-stil og skall-pipelines håndterer formatet godt.
- **Parquet.** Et kolonneorientert filformat med åpen kildekode, laget for effektiv lagring og henting, med komprimerings- og kodingsskjemaer for å håndtere komplekse data i store mengder.

Her er én post i de to første formatene. Kommaet inne i merknaden tvinger frem anførselstegn i CSV-en:

```
id,read_at,kwh,note
M-17,2024-01-08T10:00:00Z,3.21,"late, manual"

{"id":"M-17","read_at":"2024-01-08T10:00:00Z","kwh":3.21,"note":"late, manual"}
```

Slik ser det ut i praksis

Et datasett blir formet, flyttet og lest, og arbeidet nedenfor har ett prosjekt for hvert av de tre stegene.

Formet. Strømdata kom fra flere datakilder, hver med sin egen rapporteringsfrekvens, fra timesintervaller til 15-minuttersintervaller og i noen tilfeller uregelmessige tidsstempler. Det gjorde det vanskelig å lage et sammenhengende, sammenlignbart datasett. Arbeidet tok rå hendelsesdata fra et historisk JSONL-datasett og gjorde dem om til timesforbruk, med én rad per time og per sone. Produksjonsmiksen i hvert intervall ble summert, import lagt til og eksport trukket fra. Tre varianter ble skrevet, i Python, JQ og SQL. Python-pipelinen parset JSONL-filene til datarammer, samplet tidsserien om til timesintervaller og eksporterte resultatene som CSV eller lastet dem inn i en lettvekts SQLite-database for inspeksjon. Målt på historiske datasett på 1 dag og 30 dager var den raskest av de tre, med 34 ms for ett døgn og 180 ms for tretti. Arbeidet mellom inndata og utdata er en endring i granularitet: fra hver kildes egen takt til én fast rad per time og sone.

Flyttet. Et sett med seks ETL/ELT-pipelines fraktet data fra et eksternt datavarehus bak et IPsec-VPN inn i Google BigQuery. Dataene kom som komprimerte arkiver med Parquet-, CSV- og .bak-filer, og Bash- og Python-skript klassifiserte hver fil etter type og skjema. Etter at prosedyrer hadde renset, normalisert og beriket radene, ble kuraterte datasett og tabeller eksportert til mellomformater, komprimert med gzip og overført til en bucket i Google Cloud Storage. Derfra lastet bq CLI og Python-skript dem inn i BigQuery. Datasettene som før kostet 3+ timer manuelt arbeid, lå i BigQuery under 20 minutter etter at kjøringen startet, og de ble fornyet daglig i stedet for ukentlig.

Lest. Smartmålere i flere geografiske regioner produserte enorme datasett bak over 500 analyserapporter skrevet i ren SQL på PostgreSQL og BigQuery. Dataene omfattet geolokasjonskoordinater, måler-ID-er, tidsstemplet energibruk og miljømetadata. Hver rapport hadde kommentert SQL, slik at kolleger kunne reprodusere og etterprøve den, og feilsøkingsnotater dokumenterte vanlige datakvalitetsproblemer, som manglende GPS-koordinater eller korrupte målerverdier, sammen med anbefalt håndtering. Siden tilskriver arbeidet en forbedring på 10–15 % i effektiviteten i den operative planleggingen.

Hvor det går galt

Det meste av trøbbel med datasett er en opplysning om dataene som ble liggende i hodet på én person.

Granulariteten står ikke skrevet noe sted. Timesrader lagt sammen med kvartersrader teller noen intervaller fire ganger. Skriv granulariteten ned i én setning, f.eks. én rad per time per sone i UTC, og juster hver kilde etter den før noe summeres.

Et hull leses som null. Et tomt felt, en null-verdi og tallet 0 betyr tre forskjellige ting, og en sum behandler stille en manglende avlesning som null forbruk. Manglende GPS-koordinater og korrupte målerverdier var vanlige nok i smartmålerprosjektet til å få dokumentert håndtering. Bestem for hver kolonne om et hull skal droppes, fylles eller flagges, og skriv hvilket i notatene til datasettet.

CSV-dialekten glir. RFC 4180 beskriver det vanlige oppsettet, og virkelige filer avviker fra det: et annet skilletegn, en overskrift som mangler, et felt med komma der anførselstegnene er utelatt. RFC-en gjør overskriftslinjen valgfri og

signaliserer den bare med en MIME-parameter, som forsvinner i det øyeblikket filen kopieres til disk. Sjekk første linje i hver ny fil mot den forventede overskriften før innlasting.

En kopi glemmer hvor den kom fra. Et kuratert uttrekk lastet inn i et annet system ser en måned senere helt likt ut som originalen. Noter kilden, perioden, uttrekkstidspunktet og en kontrollsum ved siden av hver kopi, slik at en rapport kan si hvilket uttrekk den leste.

Et BigQuery-datasett havner på feil sted. Plasseringen til et datasett låses når det opprettes; å flytte det betyr å kopiere eller opprette datasettet på nytt. Alle tabellene i én spørring må ligge i datasett på samme plassering, så velg plassering før den første innlastingen.

Les mer

- Data Catalog Vocabulary (DCAT) - Version 3 — W3Cs definisjoner av dataset, distribution og dataset series, med egenskapene som beskriver hver av dem.
- BigQuery: Introduction to datasets — ordet i betydningen beholder: prosjekter, plasseringer, navngivning og grenser.
- JSON Lines — formatets tre regler og konvensjonene for filnavn.
- RFC 4180: Common Format and MIME Type for CSV Files — CSV-oppsettet de fleste verktøy følger, reglene for anførselstegn og typen text/csv.
- Apache Parquet: Overview — hva det kolonneorienterte filformatet er, og hvor spesifikasjonen finnes.

Ordliste

Velg ut fra deres faktiske rammer

Ta med et representativt datasett, spørsmålene det skal besvare og dagens flaskehals. Vi kan avtale målinger av kvalitet, gjennomstrømning og gjenoppretting.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er et datasett?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/dataset/