Obdelava in optimizacija velepodatkov (angl. Big Data)

Učni izidi

Po zaključku te učne enote bodo študenti sposobni:

- Pojasniti pojem velepodatkov in njihove značilnosti (obseg, hitrost, raznovrstnost).
- Prepoznati ključne vire okoljskih velepodatkov (satelite, senzorje, odprte repozitorije).
- Razumeti glavne izzive pri shranjevanju, čiščenju in obdelavi obsežnih podatkovnih nizov.
- Uporabiti ustrezna orodja za obdelavo velepodatkov (Google BigQuery, Spark, Hadoop itd.).
- Optimizirati podatkovne poizvedbe in delovne tokove z vidika zmogljivosti in stroškovne učinkovitosti.
- Vizualizirati velike podatkovne nize z nadzornimi ploščami ter razložiti rezultate za pridobivanje okoljskih spoznanj.
- Zasnovati in izvesti razširljiv delovni tok za obdelavo, analiziranje in vizualiziranje okoljskih podatkovnih nizov.

Uvod

Okoljske znanosti so vstopile v obdobje, v katerem obseg in kompleksnost podatkov ter hitrost njihovega nastajanja močno presegajo zmogljivosti, za katere so bile zasnovane tradicionalne metode upravljanja in analiziranja podatkov. Izraz velepodatki označuje podatkovne nize, ki so tako veliki, hitro spremenljivi ali raznovrstni, da pridobivanje smiselnih informacij iz njih zahteva napredna računska orodja, porazdeljene sisteme in nove analitične pristope. V okoljskih znanostih pri velepodatkih ne gre zgolj za velikost, temveč za neprekinjene podatkovne tokove, več oblik zapisa podatkov ter izrazito prostorsko in časovno razsežnost, ki jih je treba analizirati skupaj.

Velepodatke običajno opisujemo z več ključnimi značilnostmi, ki se pogosto imenujejo lastnosti »V«:

- Obseg – ogromne količine podatkov, od terabajtov do petabajtov, ki jih vsak dan ustvarijo sateliti, senzorska omrežja in simulacije.
- Hitrost – hitro nastajanje podatkov, na primer na vremenskih postajah, merilnikih kakovosti zraka in oceanskih bojah v realnem času.

- Raznovrstnost – različne vrste podatkov, vključno s številskimi meritvami, slikami, videoposnetki, časovnimi vrstami, besedilnimi poročili in geoprostorskimi sloji.
- Verodostojnost – različne ravni kakovosti, negotovosti in zanesljivosti podatkov, ki so posebej pomembne pri okoljskih meritvah.
- Vrednost – možnost pridobivanja uporabnega znanja za podnebno modeliranje, varstvo ekosistemov, preprečevanje nesreč in oblikovanje politik.

V okoljskih znanostih imajo velepodatki ključno vlogo pri razumevanju kompleksnih zemeljskih sistemov, odkrivanju dolgoročnih trendov in podpiranju odločitev za trajnostnost in varstvo okolja, ki temeljijo na dokazih.

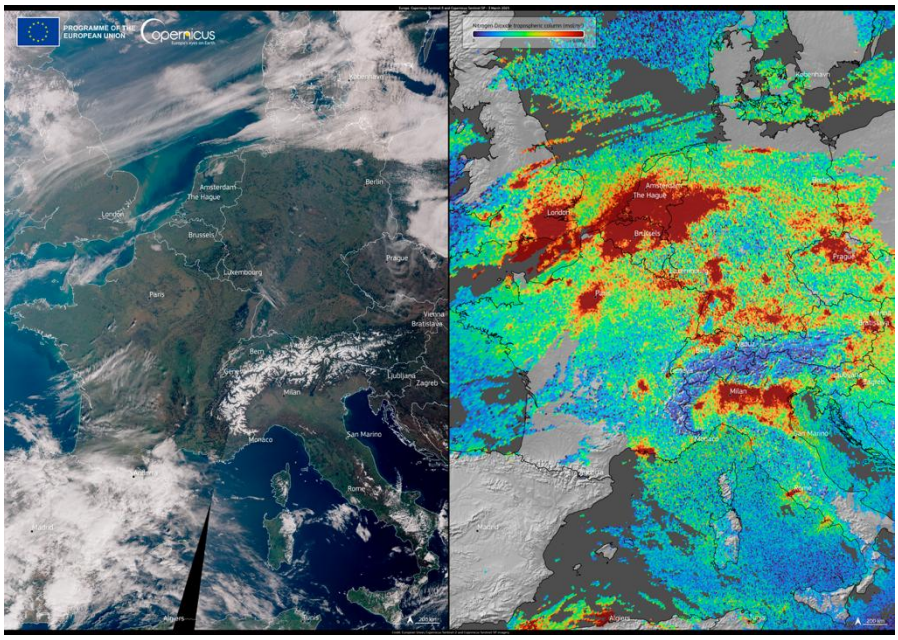
Viri okoljskih podatkov

Okoljski velepodatki izvirajo iz številnih virov, od katerih vsak ponuja edinstven pogled na naravne procese in procese pod vplivom človeka.

Satelitski podatki so eden največjih virov. Sateliti za opazovanje Zemlje neprekinjeno zbirajo podatke o rabi zemljišč, temperaturi morske površine, sestavi ozračja, stanju rastlinstva, ledenem pokrovu in dinamiki oblakov. Ti podatkovni nizi so običajno visokoločljivi, multispektralni in pokrivajo ves svet, zato so izjemno dragoceni, a tudi računsko zahtevni.

Senzorji in situ (meritve in situ zahtevajo, da je merilna oprema nameščena neposredno na opazovani lokaciji in v stiku s predmetom opazovanja) zagotavljajo referenčne meritve in visokofrekvenčne podatke. Mednje spadajo vremenske postaje, ki merijo temperaturo, vlažnost, tlak in padavine; oceanografske boje, ki spremljajo slanost, tokove in koncentracijo klorofila; senzorji kakovosti zraka, ki zaznavajo onesnaževala, kot so PM2.5, NO_2 in ozon; ter rečni in talni senzorji, ki spremljajo vodostaj, vlažnost in vsebnost hranil itd. Takšni senzorji pogosto delujejo neprekinjeno in ustvarjajo pretočne podatke, ki jih je treba obdelati skoraj v realnem času.

Svetovne odprte podatkovne zbirke dodatno bogatijo okoljske raziskave. Organizacije in pobude, kot so meteorološke agencije, vesoljske agencije in mednarodni raziskovalni konzorciji, objavljajo prosto dostopne podatkovne nize s področij podnebja, biotske raznovrstnosti, oceanografije in znanosti o ozračju. Ti podatkovni nizi raziskovalcem omogočajo združevanje lokalnih meritev z opazovanji na svetovni ravni ter s tem primerjalne in dolgoročne študije. Ti viri skupaj tvorijo kompleksen, medsebojno povezan podatkovni ekosistem, ki opredeljuje velepodatke v okoljskih znanostih.



Slika 6.1 Satelit Copernicus Sentinel-5P 3. marca 2025 s poudarjenimi koncentracijami dušikovega dioksida (NO_2).

Življenjski cikel velepodatkov

Upravljanje okoljskih velepodatkov vključuje strukturiran proces, imenovan življenjski cikel velepodatkov (Slika 6.2), ki obsega več medsebojno povezanih stopenj.

Zbiranje podatkov: podatki se pridobivajo iz satelitov, senzorjev, terenskih meritev, simulacij in podatkovnih zbirk. Na tej stopnji je treba upoštevati pogostost vzorčenja, prostorsko ločljivost in oblike zapisa podatkov, da se zagotovita združljivost in znanstvena ustreznost.

Shranjevanje podatkov: zaradi velikosti in kompleksnosti okoljskih podatkovnih nizov se za shranjevanje pogosto uporabljajo porazdeljeni sistemi, platforme v oblaku in specializirane podatkovne zbirke, optimizirane za časovne vrste in geoprostorske podatke. Učinkovito shranjevanje je bistveno za razširljivost in dolgoročno dostopnost.

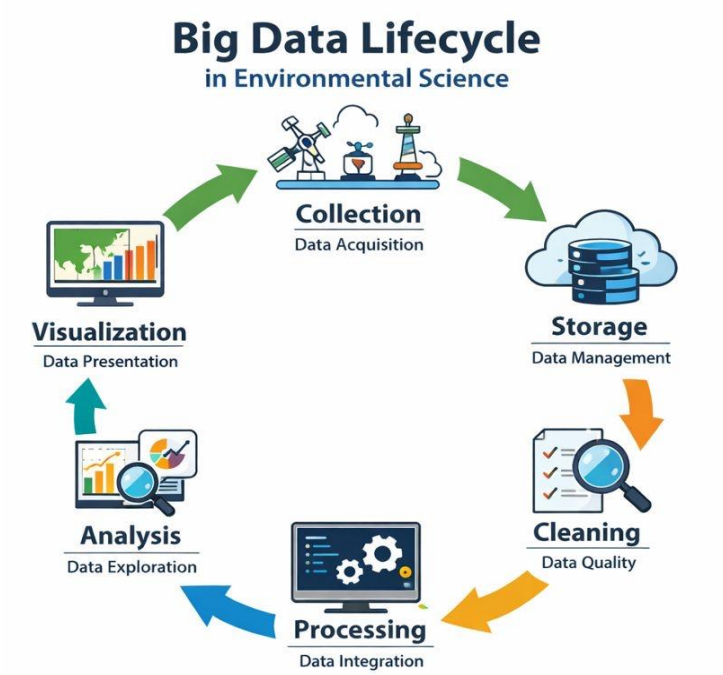
Čiščenje in predobdelava podatkov: okoljski podatki pogosto vsebujejo manjkajoče vrednosti, šum, osamelce ali nedosledne meritve. Čiščenje vključuje odkrivanje napak, zapolnjevanje vrzeli, poenotenje enot ter časovno ali prostorsko usklajevanje. Ta korak je ključnega pomena, saj lahko slaba kakovost podatkov privede do zavajajočih sklepov.

Obdelava podatkov: na tej stopnji se neobdelani podatki pretvorijo v strukturirane oblike, primerne za analizo. To lahko vključuje agregiranje, filtriranje, ponovno vzorčenje in povezovanje več

podatkovnih nizov. Pri velepodatkih je obdelava pogosto vzporedna, da je mogoče učinkovito obdelati velike količine podatkov.

Analiza podatkov: analitične metode segajo od statistične analize in odkrivanja trendov do strojnega učenja in napovednega modeliranja. V okoljskih znanostih se analiza uporablja za prepoznavanje vzorcev, ocenjevanje tveganj in simuliranje prihodnjih scenarijev.

Vizualizacija podatkov: na tej stopnji se kompleksni podatkovni nizi pretvorijo v intuitivne zemljevide, grafikone, nadzorne plošče in animacije. Učinkovita vizualizacija je bistvena za posredovanje rezultatov znanstvenikom, odločevalcem in javnosti, zlasti pri obravnavi prostorskih in časovnih pojavov.



Slika 6.2 Življenjski cikel velepodatkov

Ključni izzivi pri okoljskih velepodatkih

Okoljski velepodatki kljub svojemu potencialu prinašajo več pomembnih izzivov. Velika težava je heterogenost, saj se podatkovni nizi razlikujejo po obliki zapisa, ločljivosti, merilu in negotovosti. Združevanje satelitskih posnetkov z odčitki senzorjev in preteklimi zapisi zahteva skrbno usklajevanje.

Hitrost predstavlja izziv, kadar podatki prihajajo neprekinjeno in jih je treba obdelati v realnem času, na primer med skrajnimi vremenskimi dogodki ali ob opozorilih o onesnaženju. Tudi obseg

podatkov lahko preobremeni tradicionalne računalniške sisteme. Dolgoročni podnebni zapisi in visokoločljivi satelitski posnetki zahtevajo razširljivo infrastrukturo in učinkovite algoritme.

Kakovost in negotovost podatkov sta v okoljskih aplikacijah posebej pomembni. Okvare senzorjev, napake pri umerjanju in manjkajoči podatki so pogosti, negotovost pa je treba izrecno obravnavati, da se zagotovijo zanesljivi znanstveni sklepi.

Oblike zapisa velepodatkov in vrste shranjevanja

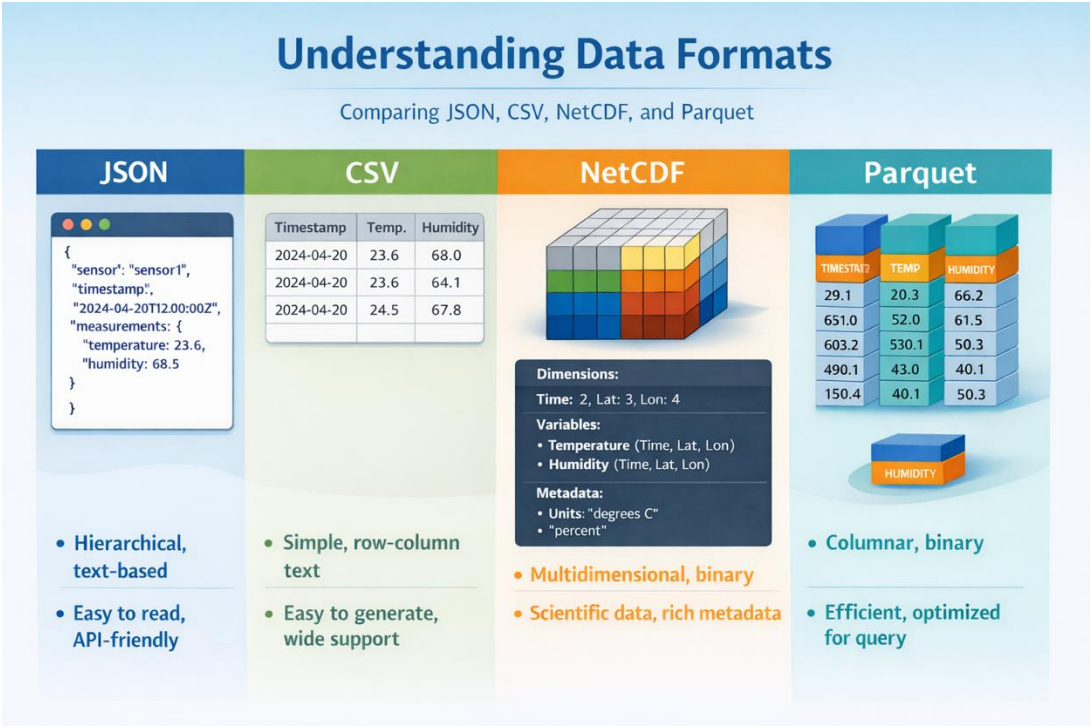
Ena najpreprostejših in najpogosteje uporabljenih oblik zapisa velepodatkov je CSV (Comma-Separated Values). Datoteke CSV hranijo podatke kot navadno besedilo, pri čemer vsaka vrstica predstavlja zapis, stolpci pa so ločeni z vejicami. Zaradi preprostosti je datoteke CSV enostavno ustvarjati, pregledovati in izmenjevati med platformami. Pogosto se uporabljajo za tabelarične podatke, kot so vremenska opazovanja, meritve onesnaženosti in statistični povzetki. Vendar datoteke CSV nimajo vgrajene podpore za metapodatke, slabo obravnavajo hierarhične podatke in pri zelo velikih podatkovnih nizih postanejo neučinkovite.

JSON (JavaScript Object Notation) je polstrukturirana oblika zapisa podatkov, zasnovana za predstavljanje hierarhičnih in ugnezdenih podatkov. JSON je posebej primeren za podatke, ki se izmenjujejo prek spletnih vmesnikov API, naprav IoT in storitev v oblaku. V sistemih za spremljanje okolja senzorska omrežja pogosto prenašajo podatke v obliki JSON, v kateri so meritve, časovni žigi in identifikatorji naprav združeni v strukturirane objekte. Čeprav je JSON človeku berljiv in prilagodljiv, je zaradi obsežnosti zapisa pri obdelavi zelo velikih podatkovnih nizov manj učinkovit od binarnih oblik.

Za odpravljanje omejitev zmogljivosti in razširljivosti sodobni sistemi velepodatkov pogosto uporabljajo stolpčne binarne oblike zapisa, kot je Parquet. Apache Parquet je optimiziran za analitične delovne obremenitve ter se pogosto uporablja v sistemih za porazdeljeno obdelavo, kot je Apache Spark, in podatkovnih skladiščih v oblaku. Ker podatke shranjuje po stolpcih in ne po vrsticah, Parquet omogoča hitrejše izvajanje poizvedb, boljše stiskanje in nižje stroške shranjevanja. Ta oblika je posebej učinkovita za velike zbirke preteklih podatkov, kot so podnebni zapisi ali dolgoročne okoljske meritve.

Na znanstvenih področjih, zlasti v klimatologiji in oceanografiji, je NetCDF (Network Common Data Form) zelo razširjena oblika zapisa. NetCDF je posebej zasnovan za shranjevanje večrazsežnih podatkov, kot so temperatura, slanost ali zračni tlak v prostoru in času. V nasprotju

s splošnimi oblikami zapisa NetCDF vključuje bogate metapodatke, ki opisujejo spremenljivke, enote in koordinatne sisteme, zato je zelo primeren za ponovljive znanstvene raziskave.



Slika 6.3 Razumevanje oblik zapisa podatkov – JSON, CSV, NetCDF in Parquet

Poleg oblik zapisa podatkov (slika 6.3) so sistemi velepodatkov odvisni od različnih vrst shranjevanja. Tradicionalno lokalno shranjevanje, na primer datoteke na osebnih računalnikih ali laboratorijskih strežnikih, je preprosto za upravljanje, vendar ima omejeno razširljivost in zanesljivost. Ko podatkovni nizi postajajo večji in kompleksnejši, lokalno shranjevanje ne zadošča več za sodelovalno izvajanje analiz v velikem obsegu.

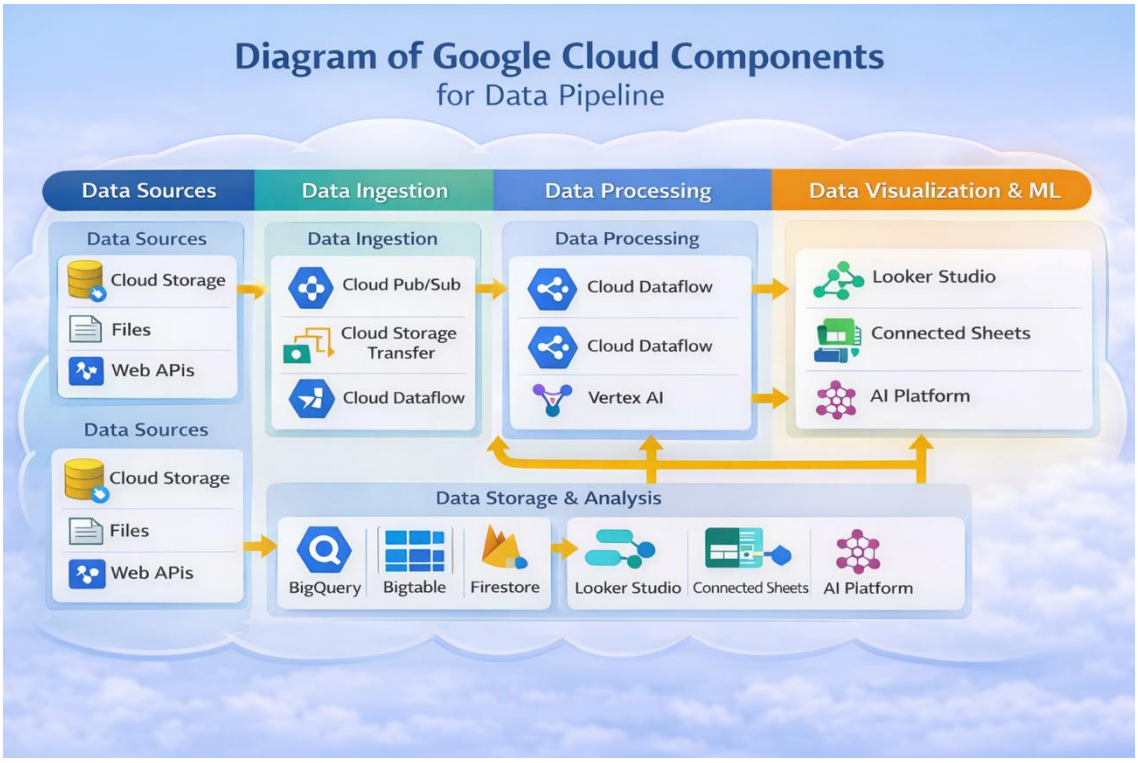
Obdelava velepodatkov zahteva usklajene programske ekosisteme, ki lahko upravljajo shranjevanje, računanje in dodeljevanje virov med številnimi računalniki. Eden najzgodnejših in najvplivnejših ekosistemov je Apache Hadoop, ki je uvedel koncept porazdeljenega shranjevanja (HDFS) v povezavi s porazdeljenim računanjem prek programskega modela MapReduce. Hadoop je organizacijam omogočil obdelavo podatkovnih nizov, ki so bili prej preveliki za posamezne računalnike.

Čeprav je Hadoop postavil temelje velepodatkov, so novejša ogrodja, kot je Apache Spark, zaradi izboljšane zmogljivosti in prilagodljivosti postala zelo razširjena. Spark podpira računanje v pomnilniku, kar znatno pospeši obdelavo podatkov. Zagotavlja tudi API-je na visoki ravni za analizo podatkov, strojno učenje in obdelavo grafov. V okoljski analitiki se Spark pogosto

uporablja za obdelavo velikih podatkovnih nizov senzorskih meritev ali arhivov satelitskih posnetkov.

V zadnjih letih so platforme v oblaku preoblikovale področje velepodatkov z upravljanimi storitvami, ki odpravljajo potrebo po ročnem upravljanju infrastrukture. Google Cloud Platform (GCP) – Slika 6.4, Amazon Web Services (AWS) in Microsoft Azure zagotavljajo integrirane ekosisteme, ki združujejo orodja za shranjevanje, obdelavo, analitiko in vizualizacijo. Te platforme raziskovalcem omogočajo, da se namesto na upravljanje sistemov osredotočijo na analizo podatkov.

Ekosistemi velepodatkov v oblaku so posebej privlačni za interdisciplinarna področja, kot so okoljske znanosti, na katerih je obseg podatkov velik, računske zahteve pa se sčasoma spreminjajo. Podpirajo elastično razširjanje, kar pomeni, da se lahko viri dinamično povečajo ali zmanjšajo glede na zahteve delovne obremenitve ter tako optimizirajo zmogljivost in stroške.



Slika 6.4 Komponente Google Cloud

Predobdelava in čiščenje podatkov v velikem obsegu

Neobdelani podatki so le redko primerni za takojšnjo analizo. V praksi velepodatki pogosto vsebujejo manjkajoče vrednosti, nedoslednosti, podvojene zapise in merske napake. Predobdelava in čiščenje podatkov sta zato ključna koraka vsakega delovnega toka za velepodatke, zlasti pri delu z okoljskimi podatkovnimi tokovi ali podatkovnimi tokovi IoT.

OpenRefine je zelo razširjeno odprtokodno orodje za interaktivno čiščenje in preoblikovanje podatkov. Uporabnikom omogoča raziskovanje podatkovnih nizov, odkrivanje anomalij, poenotenje vrednosti in odpravljanje napak brez programerskega znanja. OpenRefine je posebej uporaben za pripravo podatkovnih nizov, pridobljenih iz javnih repozitorijev ali terenskih raziskav.

Trifacta Wrangler, zdaj znan kot Google Cloud Dataprep, te zmožnosti razširja na obsežne podatkovne nize v oblaku. Zagotavlja vizualni vmesnik, ki temelji na pravilih in uporabnikom omogoča opredeliti korake pretvorbe podatkov, ki se lahko izvajajo v velikem obsegu. To je posebej uporabno pri pripravi velikih podatkovnih nizov, shranjenih v oblačnih okoljih, kot sta Google Cloud Storage ali BigQuery.

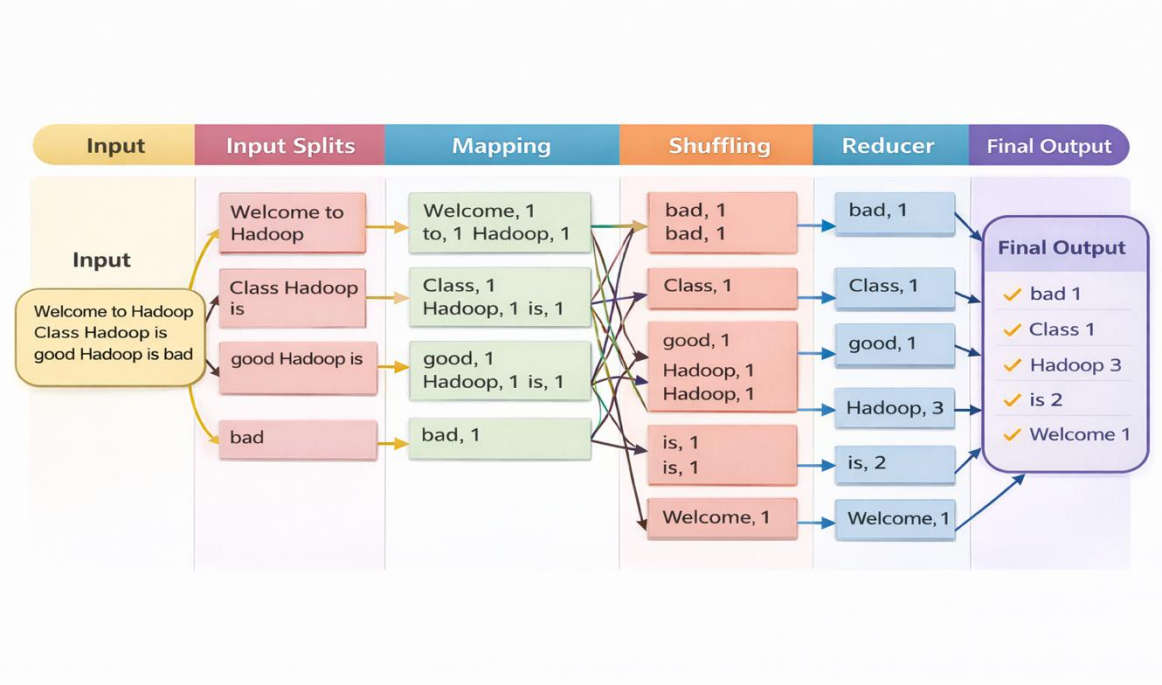
Uporabnikom s programerskim znanjem Pandas, knjižnica za Python, ponuja zmogljive možnosti za obdelavo podatkov. Pandas se pogosto uporablja skupaj z okolji Jupyter notebooks in Google Colab za čiščenje, filtriranje in preoblikovanje podatkovnih nizov pred obsežno obdelavo. Čeprav je Pandas najprimernejši za srednje velike podatkovne nize, ima ključno vlogo pri raziskovalni analizi podatkov in izdelavi prototipov.

Koncepti porazdeljene in vzporedne obdelave

Jedro analitike velepodatkov je koncept porazdeljene in vzporedne obdelave. Namesto zaporedne obdelave podatkov v enem računalniku sistemi velepodatkov razdelijo podatkovne nize na manjše dele, ki jih je mogoče hkrati obdelovati na več vozliščih.

MapReduce (Slika 6.5) je način obdelave zelo velikih podatkovnih nizov, pri katerem se delo razdeli na številne majhne naloge, ki se lahko izvajajo hkrati. Glavni razlog za njegovo priljubljenost je, da omogoča obdelavo podatkov, ki so »preveliki za en računalnik«, brez potrebe po enem izjemno zmogljivem računalniku. Namesto tega uporabimo več običajnih računalnikov (ali računalnikov v oblaku) in podatkovni niz razdelimo mednje. To je posebej uporabno, kadar so podatki shranjeni v več datotekah ali na več strežnikih, na primer dnevniki, odčitki senzorjev, satelitske ploščice ali dolgoročni okoljski arhivi.

MapReduce najlažje razumemo tako, da se osredotočimo na vlogi obeh funkcij. Funkcija Map neobdelani vhod pretvori v vmesne podatke, ki jih je pozneje enostavno združiti. Težave ne poskuša »dokončno rešiti«, temveč pripravi strukturirane gradnike (pare ključ–vrednost). Funkcija Reduce nato vse vmesne podatke z istim ključem združi v končni rezultat. V tej obliki je mogoče izraziti številne pomembne naloge nad velepodatki: štetje, seštevanje, združevanje v skupine, izračunavanje povprečij, iskanje največjih in najmanjših vrednosti ali izdelavo statistik po lokacijah.



Slika 6.5 Delovanje modela MapReduce

Korak »Shuffle« (ki se včasih obravnava kot del sistema in ne kot korak, ki ga določi uporabnik) je ključen, saj zagotavlja, da se vsi elementi z istim ključem znajdejo skupaj. V praksi je shuffle tudi ena najdražjih faz, ker pogosto zahteva prenos podatkov po omrežju med računalniki. Če na primer velik podatkovni niz preslikamo v ključe, kot sta »station_id« ali »city«, mora sistem vse zapise za vsako postajo poslati istemu izvajalcu funkcije Reduce. Ker je prenos po omrežju počasnejši od operacij v pomnilniku, dobra zasnova MapReduce zagotavlja majhen obseg vmesnih podatkov in uravnotežene ključe, da se noben izvajalec funkcije Reduce ne preobremeni.

Ena glavnih prednosti modela MapReduce je odpornost proti napakam. V velikih porazdeljenih sistemih je običajno, da nekateri računalniki odpovejo. Ogrodja MapReduce (kot je Hadoop) so zasnovana tako, da se ob odpovedi enega delovnega vozlišča njegov del naloge samodejno

dodeli drugemu delovnemu vozlišču in se obdelava nadaljuje. To je pomembno v resničnih okoljskih aplikacijah, v katerih lahko delovni tokovi na ogromnih količinah podatkov potekajo več ur; zanesljivost postane enako pomembna kot hitrost.

Način razmišljanja po zgledu MapReduce se pri analitiki okolja in kakovosti zraka pojavlja v številnih običajnih nalogah, tudi če uporabniki izrecno ne pišejo kode MapReduce. Predstavljajte si na primer večletne meritve PM2.5 s številnih merilnih postaj. Korak Map bi lahko za vsako meritev ustvaril par (station_id, pm25_value). Korak Reduce bi nato lahko izračunal povprečja po postajah, dnevne najvišje vrednosti ali število dni, ko je bila mejna vrednost presežena.

Številni sodobni sistemi (kot so Spark, BigQuery v ozadju in podatkovni cevovodi Dataflow) danes ne zahtevajo ročne implementacije modela MapReduce, vendar ta koncept ostaja zmogljiv miselni model. Pojasnjuje, zakaj orodja za velepodatke poudarjajo operacije, kot so group by, aggregate in porazdeljena združevanja – gre za različice iste temeljne zamisli: podatke preslikati v smiselne kategorije, jih prerazporediti tako, da se sorodni elementi srečajo, nato pa jih združiti v povzetke, ki jih ljudje lahko razložijo.

Indeksiranje, optimizacija poizvedb in upravljanje porabe energije

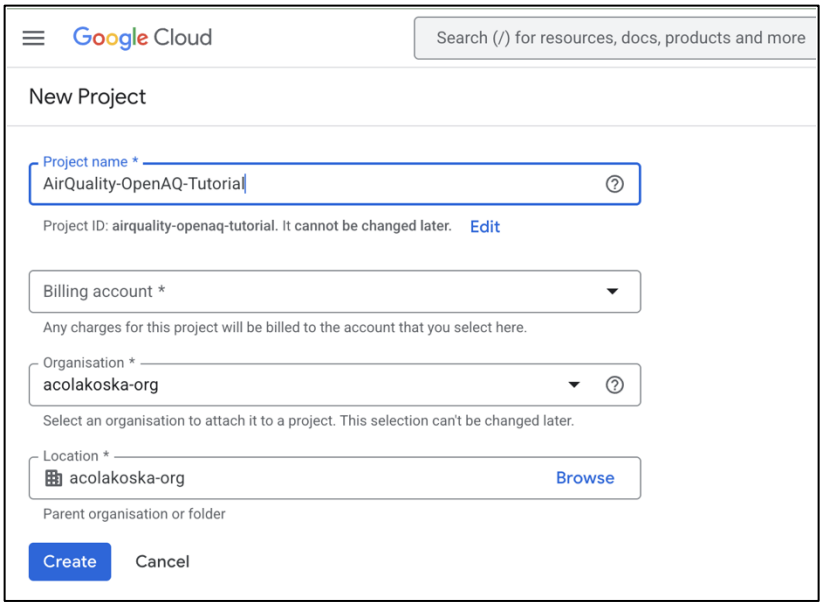
Z večanjem podatkovnih nizov postaja učinkovit dostop do podatkov vse pomembnejši. Indeksiranje je tehnika, ki izboljša zmogljivost poizvedb, saj sistemom omogoča hitro iskanje ustreznih podatkov brez pregledovanja celotnih podatkovnih nizov. V podatkovnih skladiščih v oblaku, kot je Google BigQuery, strategije indeksiranja in particioniranja znatno skrajšajo čas izvajanja poizvedb ter zmanjšajo njihove stroške.

Optimizacija poizvedb vključuje izbiro učinkovitih izvedbenih načrtov za analitične poizvedbe. Sodobni sistemi velepodatkov samodejno optimizirajo poizvedbe tako, da zmanjšujejo premikanje podatkov, izkoriščajo vzporednost in uporabljajo mehanizme predpomnjenja. Te optimizacije so posebej pomembne v oblačnih okoljih, kjer se računski viri obračunavajo glede na uporabo.

Poleg zmogljivosti sta pomembni vprašanji postali tudi upravljanje porabe energije in energetska učinkovitost. Veliki podatkovni centri porabijo znatne količine energije, kar spodbuja razvoj energijsko učinkovite strojne opreme in algoritmov za razporejanje. Ponudniki storitev v oblaku vse več vlagajo v obnovljive vire energije in energijsko ozaveščeno dodeljevanje virov, zaradi česar so oblačne rešitve za velepodatke bolj trajnostne od tradicionalnih lokalnih infrastruktur.

Primer uporabe: obdelava in analiza podatkov o kakovosti zraka v oblaku z orodji Google

Za ta primer uporabe je kot glavni vir podatkov izbran podatkovni niz OpenAQ, ki je na voljo prek Google Cloud Marketplace/ BigQuery. Podatkovni niz zagotavlja svetovna opazovanja kakovosti zraka (npr. PM2.5, PM10, NO_2, O_3), zbrana od več ponudnikov podatkov in s tisočev merilnih postaj. Posamezni zapisi običajno vključujejo prostorsko-časovne metapodatke (državo, mesto, merilno postajo in pogosto geografske koordinate), izmerjeno onesnaževalo/parameter, pripadajočo vrednost in enoto ter časovni žig. Zaradi teh značilnosti je podatkovni niz zelo primeren za obsežno okoljsko analitiko, vključno z analizo trendov časovnih vrst, prepoznavanjem prostorskih žarišč in poročanjem z nadzornimi ploščami z uporabo izvorno oblačnih orodij, kot sta BigQuery in Looker Studio. Splošneje si OpenAQ prizadeva sprejemati, agregirati, usklajevati in razširjati medopravilne podatke o kakovosti zraka v realnem času ter zgodovinske podatke, s čimer izboljšuje njihovo uporabnost in primerljivost med heterogenimi viri in oblikami zapisa.



Slika 6.6

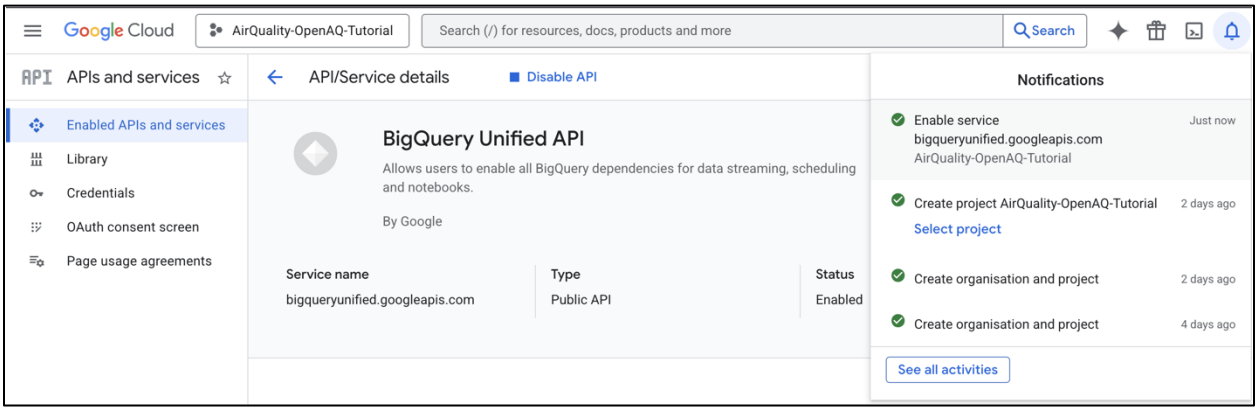
Ustvarjanje projekta Google Cloud

Za začetek moramo ustvariti namenski projekt Google Cloud (Slika 6.6), da bodo vsi viri za vajo o kakovosti zraka urejeni in ločeni od drugega dela. V Google Cloud Console uporabite izbirnik projekta v zgornji navigacijski vrstici (prikazuje ime trenutno aktivnega projekta) in izberite New Project. Projektu dodelite jasno, opisno ime, na primer AirQuality-OpenAQ-Tutorial, da ga boste pozneje pri preklapljanju med projekti, upravljanju dovoljenj ali pregledovanju podatkov o

obračunavanju zlahka prepoznali. Ko vnesete ime projekta (in izberete organizacijo ali lokacijo, če ste k temu pozvani), kliknite Create, da inicializirate projektni delovni prostor, v katerem boste upravljali vse nadaljnje podatkovne nize in poizvedbe BigQuery ter nadzorne plošče.

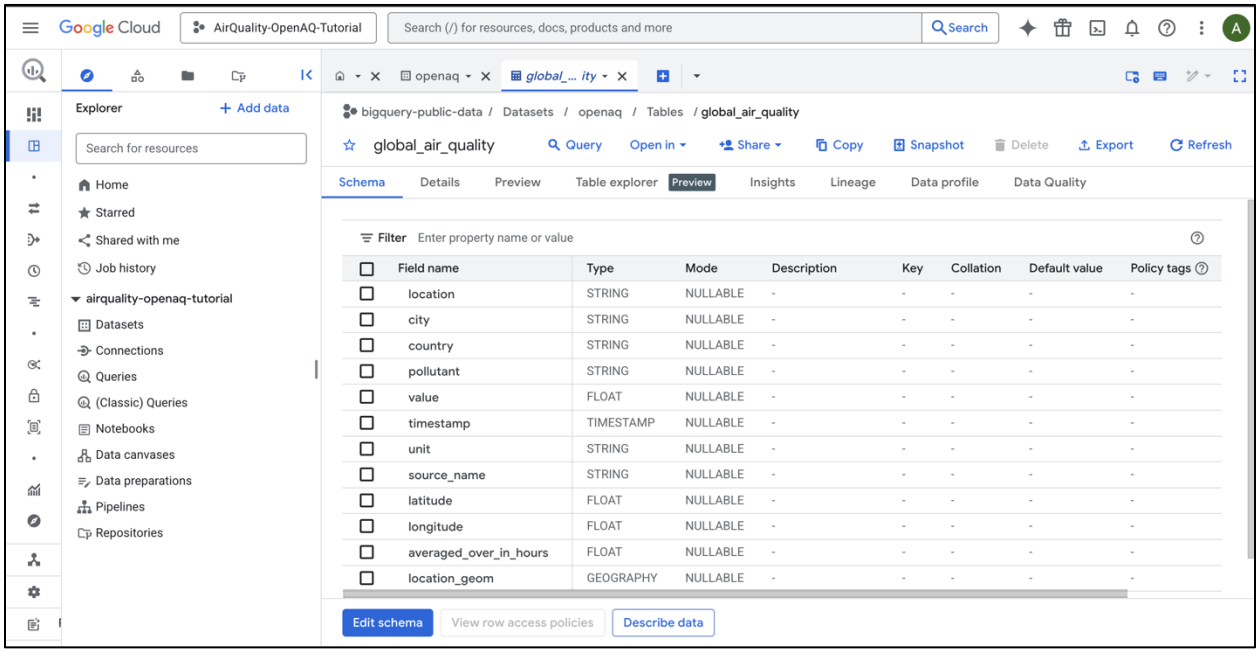
Nato moramo v Google Cloud Console odpreti BigQuery. V iskalno vrstico konzole vnesite »BigQuery« in odprite storitev BigQuery. Ko se naloži, z izbirnikom projekta v zgornji navigacijski vrstici preverite, ali delate v pravem projektu; s tem zagotovite, da bodo vsi podatkovni nizi, ki jih ustvarite, poizvedbe, ki jih izvedete, in pogledi, ki jih shranite, povezani z želenim projektom.

Prepričajte se, da je BigQuery API omogočen (Slika 6.7). V večini primerov se BigQuery API omogoči samodejno, ko prvič odprete BigQuery. Če pa naletite na napako pri dostopu ali napako, povezano z API, ga morate omogočiti ročno. Odprite APIs & Services → Enabled APIs & services in preverite, ali je BigQuery API naveden. Če ni omogočen, ga izberite in kliknite Enable, da bo projekt lahko izvajal operacije BigQuery, kot sta izvajanje poizvedb nad podatkovnimi nizi in ustvarjanje pogledov.



Slika 6.7 Omogočanje poenotenega vmesnika BigQuery API

Ko je to omogočeno, moramo dodati podatkovni niz OpenAQ. Najhitreje to storimo z uporabo Google Cloud Marketplace. Kliknite Add Data in izberite Public datasets, s čimer se odpre Marketplace. Nato poiščite OpenAQ, izberite ustrezen vnos in kliknite View dataset. S tem dejanjem postane podatkovni niz samodejno razpoložljiv v trenutnem projektu Google Cloud, ne da bi bilo treba premikati podatke ali plačati stroške shranjevanja (Slika 6.8).



Slika 6.8 Raziskovanje podatkovnega niza v Google BigQuery

V osrednji plošči je izbran pogled Schema, ki prikazuje strukturiran seznam polj (stolpcev) tabele global_air_quality. Tabela sheme vsebuje glave stolpcev, kot so Field name, Type in Mode, ter navaja atribute, povezane s kakovostjo zraka, kot so location, city, country, pollutant, value in timestamp, skupaj s polji, ki opisujejo okoliščine meritev, kot sta unit in source_name. Vključuje tudi geografske podatke, kot sta latitude in longitude, ter prostorsko polje location_geom tipa GEOGRAPHY, kar nakazuje, da podatkovni niz poleg običajnih poizvedb nad časovnimi vrstami podpira tudi kartiranje in prostorsko analizo.

Za ponovljivo analizo podatkov OpenAQ o kakovosti zraka, primerno za nadzorne plošče, moramo razviti več poizvedb BigQuery in jih shraniti kot poglede. Prva poizvedba bo meritve PM2.5 agregirala v dnevno časovno vrsto, kar bo omogočilo ocenjevanje dinamike onesnaženosti skozi čas in podprlo vizualizacijo trendov. Druga poizvedba bo prepoznala obdobja visoke onesnaženosti z razvrščanjem dni glede na njihovo povprečno koncentracijo PM2.5 ter tako poudarila skrajne dogodke, ki so lahko pomembni za javno zdravje. Tretja poizvedba bo izračunala povprečne ravni PM2.5 na merilnih postajah in pripravila primerjalno razvrstitev, ki bo olajšala odkrivanje trajnih žarišč in prostorske spremenljivosti izpostavljenosti. Četrta poizvedba bo za vsako lokacijo izbrala najnovejše opazovanje PM2.5 (ob veljavnih geografskih koordinatah), da bo ustvarila sloj, pripravljen za kartografsko vizualizacijo. Ker je podatkovni niz, ki je na voljo v BigQuery, zgodovinski posnetek (z najpoznejšim časovnim žigom iz leta 2022), poizvedbe uporabljajo časovna okna glede na najnovejši časovni žig podatkovnega niza in tako zagotavljajo dosledne rezultate kljub odsotnosti posodobitev skoraj v realnem času. Ti pogledi skupaj

zagotavljajo strukturirano analitično osnovo, ki jo je mogoče neposredno povezati z Looker Studio za interaktivne nadzorne plošče in pregledno poročanje (Slika 6.9).

1) Dnevni trend PM2.5 v zadnjih 90 dneh glede na max_ts

```
WITH t AS (
 SELECT MAX(timestamp) AS max_ts
 FROM `bigquery-public-data.openaq.global_air_quality`
)
SELECT
 DATE(g.timestamp) AS day,
 AVG(g.value) AS avg_pm25
FROM `bigquery-public-data.openaq.global_air_quality` g
CROSS JOIN t
WHERE g.pollutant = 'pm25'
 AND g.timestamp >= TIMESTAMP_SUB(t.max_ts, INTERVAL 90 DAY)
GROUP BY day
ORDER BY day;
```

Shrani kot pogled: Run → SAVE → Save view → podatkovni niz openaq_dashboard → ime v_pm25_daily_trend

2) Deset dni z najvišjimi vrednostmi PM2.5 v zadnjem letu glede na max_ts

```
WITH t AS (
 SELECT MAX(timestamp) AS max_ts
 FROM `bigquery-public-data.openaq.global_air_quality`
)
SELECT
 DATE(g.timestamp) AS day,
 AVG(g.value) AS avg_pm25,
 COUNT(*) AS measurements
FROM `bigquery-public-data.openaq.global_air_quality` g
CROSS JOIN t
WHERE g.pollutant = 'pm25'
 AND g.timestamp >= TIMESTAMP_SUB(t.max_ts, INTERVAL 365 DAY)
GROUP BY day
ORDER BY avg_pm25 DESC
LIMIT 10;
```

Shrani kot pogled: v_pm25_worst_days

3) Petnajst lokacij z najvišjo povprečno vrednostjo PM2.5 (v zadnjih 90 dneh glede na max_ts)

```
WITH t AS (
  SELECT MAX(timestamp) AS max_ts
  FROM `bigquery-public-data.openaq.global_air_quality`
)
SELECT
  COALESCE(g.country, 'Unknown') AS country,
  COALESCE(g.city, 'Unknown') AS city,
  g.location,
  AVG(g.value) AS avg_pm25,
  COUNT(*) AS measurements
FROM `bigquery-public-data.openaq.global_air_quality` g
CROSS JOIN t
WHERE g.pollutant = 'pm25'
  AND g.timestamp >= TIMESTAMP_SUB(t.max_ts, INTERVAL 90 DAY)
GROUP BY country, city, location
HAVING measurements >= 50
ORDER BY avg_pm25 DESC
LIMIT 15;
```

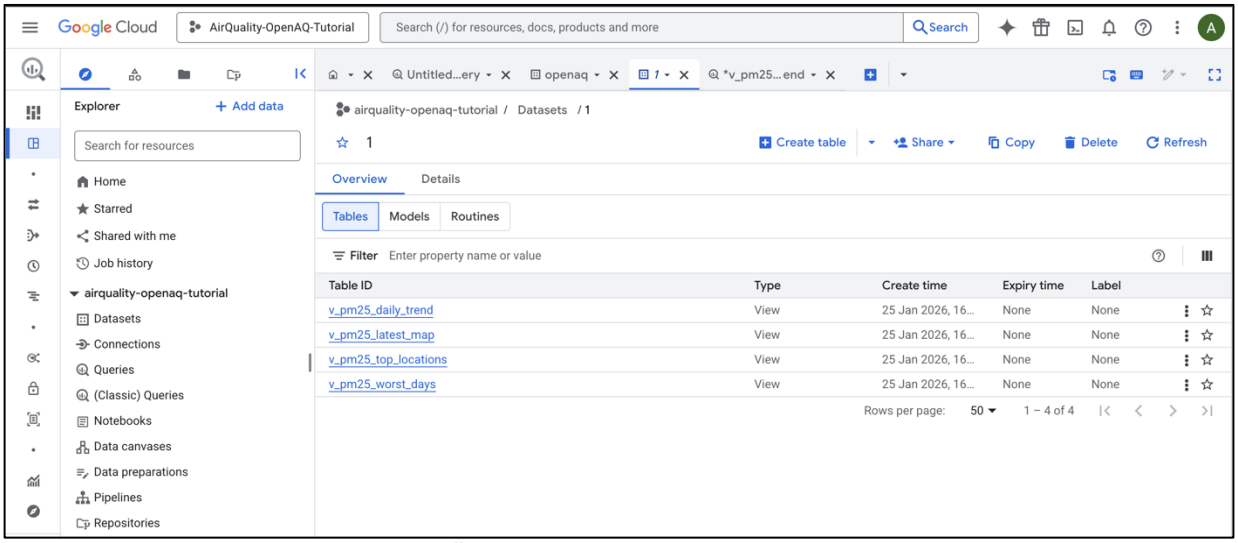
Shrani kot pogled: v_pm25_top_locations

4) Najnovejša vrednost PM2.5 za posamezno lokacijo (v zadnjih 14 dneh glede na max_ts)

```
WITH t AS (
  SELECT MAX(timestamp) AS max_ts
  FROM `bigquery-public-data.openaq.global_air_quality`
),
latest AS (
  SELECT
    g.location,
    g.city,
    g.country,
    g.latitude,
```

```
    g.longitude,
    g.value,
    g.unit,
    g.timestamp,
    ROW_NUMBER() OVER (PARTITION BY g.location ORDER BY g.timestamp DESC) AS rn
  FROM `bigquery-public-data.openaq.global_air_quality` g
  CROSS JOIN t
  WHERE g.pollutant = 'pm25'
    AND g.latitude IS NOT NULL
    AND g.longitude IS NOT NULL
    AND g.timestamp >= TIMESTAMP_SUB(t.max_ts, INTERVAL 14 DAY)
)
SELECT
  location,
  city,
  country,
  latitude,
  longitude,
  value AS latest_pm25,
  unit,
  timestamp AS latest_time
FROM latest
WHERE rn = 1;
```

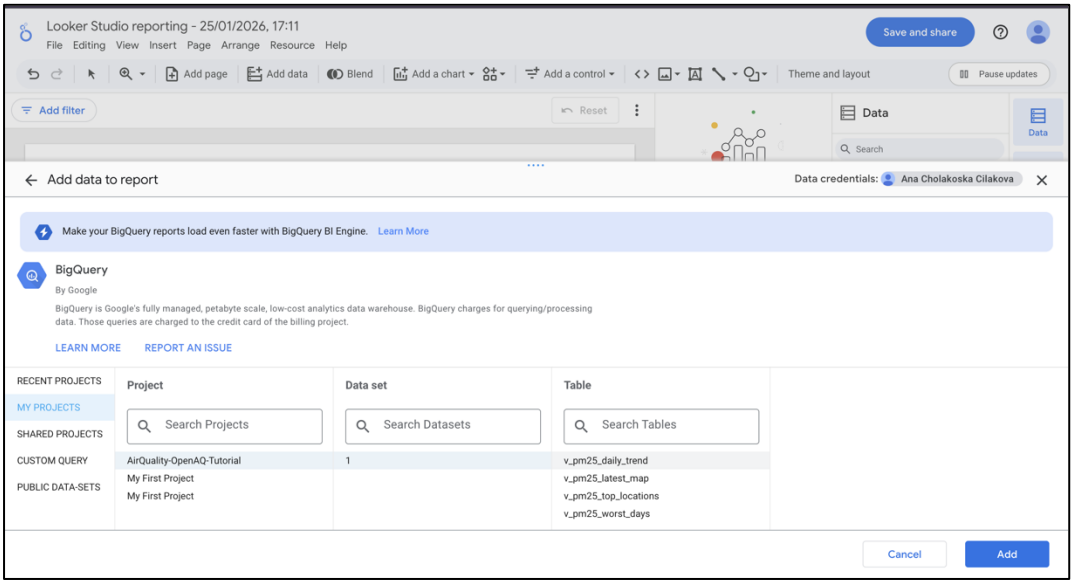
Rezultate (poglede) si lahko zdaj ogledamo na sliki 6.9.



Slika 6.9 Štirje pogledi – rezultat poizvedb SQL

Nato bomo z orodjem Looker Studio izdelali interaktivno nadzorno ploščo kakovosti zraka, ki jo bomo neposredno povezali s štirimi vnaprej izračunanimi pogledi BigQuery, izpeljanimi iz podatkovnega niza OpenAQ. Vsak pogled obravnavamo kot analitično komponento za večkratno uporabo: prvi prikazuje dnevni trend PM2.5, drugi ekstremne dogodke (»najslabši dan«), tretji razvrstitev po lokacijah, četrti pa sloj najnovejših opazovanj, pripravljen za kartografski prikaz. Tako delovni tok loči pripravo podatkov od vizualizacije ter izboljša preglednost in ponovljivost. Rezultat je povezana nadzorna plošča, ki z grafikoni, tabelami in geografskim kartiranjem podpira raziskovalno analizo ter jo je mogoče deliti za namene poročanja in sporočanja okoljskih spoznanj.

Za začetek odprite Looker Studio in izberite Create → Report, da ustvarite novo nadzorno ploščo. Če se prikaže poziv, se prijavite z računom Google in odobrite potrebna dovoljenja, da bo lahko Looker Studio dostopal do vaših virov Google Cloud. Ko se odpre prazno poročilo, povežite prvi podatkovni niz tako, da kliknete Add data in kot povezovalnik izberete BigQuery. V pregledovalniku BigQuery izberite pravi Project (npr. AirQuality-OpenAQ-Tutorial), nato podatkovni niz openaq_dashboard in pogled v_pm25_daily_trend. Nazadnje kliknite Add (in po potrebi potrdite Add to report), da pogled dnevnega trenda dodate kot začetni podatkovni vir nadzorne plošče (slika 6.10).



Slika 6.10 Dodajanje podatkov v poročilo

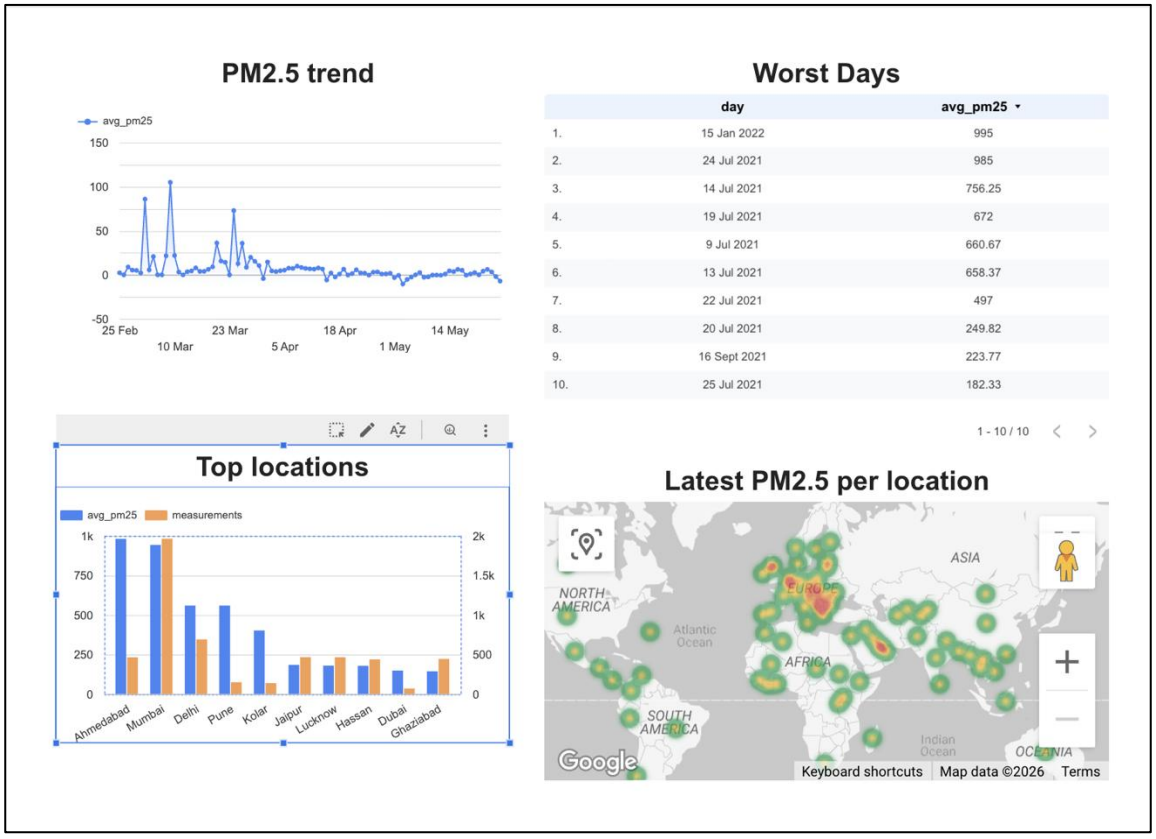
Ko pogled v_pm25_daily_trend dodate kot podatkovni vir, ustvarite grafikon Time Series za prikaz časovnega spreminjanja onesnaženosti zraka. Grafikon vstavite na platno poročila ter nastavite datumsko polje day kot glavno dimenzijo in povprečno vrednost PM2.5 (avg_pm25) kot metriko. Ta vizualizacija zagotavlja razumljiv povzetek dnevnih nihanj in dolgoročnejših trendov koncentracij delcev. Podpira prepoznavanje trajnih povečanj, zmanjšanj ali občasnih vrhov, ki so lahko povezani z meteorološkimi razmerami ali emisijskimi dogodki.

S pogledom v_pm25_top_locations dodajte grafikon Bar Chart za primerjavo povprečnih koncentracij PM2.5 med merilnimi postajami. Grafikon postavite v poročilo ter kot dimenzijo nastavite location (ali city, odvisno od želene ravni agregiranja), kot metriko pa avg_pm25. Stolpce razvrstite padajoče, da poudarite lokacije z največjo izpostavljenostjo. Grafikon omogoča hitro prepoznavanje žarišč z razvrščanjem merilnih postaj glede na njihovo povprečno koncentracijo v obdobju analize, dopolnite pa ga lahko s poljem measurements, ki pomaga ovrednotiti zanesljivost posamezne ocene.

Za poudarjanje dogodkov z izjemno onesnaženostjo vstavite vizualizacijo Table, ki temelji na pogledu v_pm25_worst_days. V tabeli nastavite day kot dimenzijo ter avg_pm25 (in, kjer je na voljo, measurements) kot metriko. Tabelo razvrstite po povprečni vrednosti PM2.5 v padajočem vrstnem redu, tako da bodo najhujši dnevi na vrhu. Tabelarični prikaz podpira natančno poročanje, saj prikazuje točne vrednosti in omogoča primerjavo med dnevi z najvišjimi

vrednostmi, kar je posebej uporabno za jedrnat povzetek obdobij okoljskega tveganja v obliki, primerni za revizijo.

Za podporo prostorski razlagi nazadnje ustvarite grafikon Google Maps s pogledom v_pm25_latest_map. Zemljevid dodajte na platno ter latitude in longitude določite kot geografski polji, latest_pm25 pa kot glavno metriko za prikaz velikosti zadnjih izmerjenih vrednosti (npr. z velikostjo ali intenzivnostjo oznak). Če opise orodij dopolnite s kontekstualnimi atributi, kot so location, city in latest_time, lahko uporabniki pregledajo posamezne merilne postaje. Tako se geografski kontekst združi s količinskimi vrednostmi in pokaže, kje so bile povišane koncentracije opažene nazadnje. Vse grafikone si lahko ogledamo na sliki 6.11.



Slika 6.11 Celotna nadzorna plošča

Po želji lahko nadaljujemo z optimizacijo četrte poizvedbe, da zmanjšamo računsko obremenitev in izboljšamo stroškovno učinkovitost, pri tem pa ohranimo enak analitični namen (tj. pridobivanje najnovejše meritve PM2.5 za vsako merilno postajo). Najprej skupni tabelni izraz in naknadni filter (WHERE rn = 1) nadomestimo s stavkom QUALIFY, kar okolju BigQuery omogoča, da filter okenske funkcije uporabi neposredno v enem samem pregledu podatkov; to poenostavi izvajanje in navadno izboljša zmogljivost. Nato dodamo izrecna preverjanja timestamp IS NOT NULL, da preprečimo nepotrebno obdelavo neveljavnih zapisov in zagotovimo dosledno delovanje časovnega filtra. Prvotni pogoj »zadnjih 14 dni od trenutnega datuma« nato preoblikujemo tako, da uporablja okno glede na največji razpoložljivi časovni žig podatkovnega niza (MAX(timestamp)). To je bistveno za zgodovinske posnetke (na primer tabele OpenAQ, ki se ne posodabljajo neprekinjeno) in zagotavlja neprazne rezultate. Ker nazadnje izberemo le polja, potrebna za kartiranje in poročanje (metapodatke lokacije, koordinate, vrednost, enoto in čas), poizvedba zmanjša število prebranih stolpcev, s tem pa tudi količino obdelanih bajtov in čas osveževanja nadzorne plošče. Celoten izsek kode je prikazan spodaj:

```
SELECT
 location,
 city,
 country,
 latitude,
 longitude,
 value AS latest_pm25,
 unit,
 timestamp AS latest_time
FROM `bigquery-public-data.openaq.global_air_quality`
WHERE pollutant = 'pm25'
 AND timestamp IS NOT NULL
 AND latitude IS NOT NULL
 AND longitude IS NOT NULL
 -- Use dataset's latest timestamp window if your snapshot is old:
 AND timestamp >= TIMESTAMP_SUB(
```

```
  (SELECT MAX(timestamp) FROM `bigquery-public-data.openaq.global_air_quality` WHERE timestamp IS NOT NULL),
  INTERVAL 14 DAY
 )
QUALIFY ROW_NUMBER() OVER (PARTITION BY location ORDER BY timestamp DESC) = 1;
```

Praktična vaja

Mini projekt: izberite okoljski podatkovni niz (npr. o kakovosti zraka, morskem okolju ali biotski raznovrstnosti). Z orodji Google Cloud podatke očistite in obdelajte, izvedite poizvedbe ter vizualizirajte rezultate. Delovni tok in ugotovitve predstavite v kratki predstavitvi (5–10 minut).

Preverjanje znanja

1. Katera možnost najbolje opisuje tri lastnosti »V« velepodatkov?

A) Volume (obseg) = natančnost podatkov, Velocity (hitrost) = velikost podatkov, Variety (raznovrstnost) = hitrost podatkov

B) Volume (obseg) = količina podatkov, Velocity (hitrost) = hitrost ustvarjanja/obdelave podatkov, Variety (raznovrstnost) = različne vrste/viri podatkov

C) Volume (obseg) = število uporabnikov, Velocity (hitrost) = hitrost interneta, Variety (raznovrstnost) = število strežnikov

D) Volume (obseg) = stroški shranjevanja, Velocity (hitrost) = stroški poizvedbe, Variety (raznovrstnost) = število nadzornih plošč

2. Katera povezava med oblikami zapisa in vrstami podatkov je najtočnejša?

A) CSV = nestrukturirani podatki, JSON = strukturirani podatki, slike = polstrukturirani podatki

B) CSV = strukturirani podatki, JSON = polstrukturirani podatki, slike/videoposnetki = nestrukturirani podatki

C) CSV = polstrukturirani podatki, JSON = nestrukturirani podatki, slike = strukturirani podatki

D) CSV = nestrukturirani podatki, JSON = nestrukturirani podatki, slike = strukturirani podatki

3. Potrebujete hitro analitiko za velike dnevnike dogodkov (s stolpci, kot so timestamp, device_id, PM2.5 in temperature), shranjene v objektni shrambi v oblaku. Katera izbira je najboljša?

A) Parquet/ORC (stolpčna oblika zapisa)

B) Navadno besedilo (.txt)

C) Slike PNG

D) XML z globoko ugnezdenimi oznakami

4. Kateri nabor pravilno povezuje storitve z nalogami?

A) BigQuery = objektna shramba, Cloud Storage = analitika SQL, Dataflow = vizualizacija

B) Cloud Storage = objektna shramba, BigQuery = brezstrežniško analitično podatkovno skladišče, Dataflow = porazdeljena obdelava podatkov (ETL)

C) BigQuery = čakalna vrsta sporočil, Pub/Sub = nadzorne plošče, Looker = zajem podatkov

D) Dataflow = hramba datotek, Cloud Storage = učenje modelov ML, BigQuery = upravljanje identitet

5. Katero dejanje običajno zmanjša stroške in čas izvajanja poizvedb za podatke časovnih vrst v podatkovnem skladišču v oblaku (npr. BigQuery)?

A) Uporabite SELECT *, da ne izpustite nobenega stolpca

B) Podatke razdelite na particije po datumu/času in jih gručite po pogosto uporabljenih ključih za filtriranje (npr. station_id/device_id)

C) Vse shranite v en sam velikanski stolpec JSON, da poenostavite shemo

D) Onemogočite predpomnjenje in poizvedbe večkrat znova izvedite, da »ogrejete« sistem

Viri / dodatno branje

[1] BigQuery public datasets (overview + how to access). Dostopno na: https://docs.cloud.google.com/bigquery/public-data, dostopano: 15. januarja 2026

[2] BigQuery: Introduction to optimizing query performance. Dostopno na: https://docs.cloud.google.com/bigquery/docs/best-practices-performance-overview, dostopano: 15. januarja 2026

[3] BigQuery: Optimize query computation (best practices). Dostopno na: https://docs.cloud.google.com/bigquery/docs/best-practices-performance-compute, dostopano: 15. januarja 2026

[4] Looker Studio: Connect to Google BigQuery. Dostopno na: https://docs.cloud.google.com/looker/docs/studio/connect-to-google-bigquery, dostopano: 15. januarja 2026

[5] Looker Studio Documentation. Dostopno na: https://docs.cloud.google.com/looker/docs/studio, dostopano: 15. januarja 2026

[6] Looker Studio quick start guide . Dostopno na: https://docs.cloud.google.com/looker/docs/studio/quick-start-guide, dostopano: 15. januarja 2026

[7] Cloud Billing: Budgets and budget alerts. Dostopno na: https://docs.cloud.google.com/billing/docs/how-to/budgets, dostopano: 15. januarja 2026

[8] Cloud Billing documentation hub. Dostopno na: https://docs.cloud.google.com/billing/docs, dostopano: 15. januarja 2026

[9] BigQuery sharing (formerly Analytics Hub): Introduction. Dostopno na: https://docs.cloud.google.com/bigquery/docs/analytics-hub-introduction, dostopano: 15. januarja 2026

[10] BigQuery Explained (architecture overview, Google Cloud Blog). Dostopno na: https://cloud.google.com/blog/products/data-analytics/new-blog-series-bigquery-explained-overview, dostopano: 15. januarja 2026