UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

Corso di Laurea Magistrale in Informatica

Basi di Dati 2

SOS Earth

Sistema di Monitoraggio per la Sostenibilità Ambientale

Candidate:

Raffaella Spagnuolo

Alessia Ture

Professori:

Prof. Genoveffa Tortora

Dr. Luigi Di Biasi

ANNO ACCADEMICO 2023/2024

Indice

1 Introduzione **4**

1.1 Obiettivi del Progetto 5

1.2 Struttura del Documento 5

2 Descrizione del Progetto **7**

2.1 Panoramica 7

2.2 Struttura del Progetto 7

2.3 Funzionalità del Sito *SOS Earth* 8

2.3.1 Home 9

2.3.2 Problem 9

2.3.3 Help 9

2.3.4 Project 10

2.3.5 Calcolatore di Foodprint 10

2.3.6 Indicators 11

3 Datasets **14**

3.1 Introduzione 14

3.2 Fonti dei Dati 14

3.3 Descrizione dei Dataset 15

3.3.1 Dati Economici e Sociali (World Bank) 15

3.3.2 Dati Atmosferici e Climatologici (NASA) 20

3.3.3 Dati sulle Aree Protette (IUCN) 21

3.3.4 Dati sugli Impatti della Produzione Alimentare (Our World In Data) 22

3.3.5 Soluzioni ai Problemi Ambientali 24
3.3.6 Correlazioni dei Sistemi Terrestri 25

4 Dettagli del Processo ETL **26**
4.1 Estrazione (Extract) 26
4.2 Trasformazione (Transform) 27
4.2.1 Trasformazioni Specifiche per Dataset 27
4.3 Caricamento (Load) 28
4.3.1 Caricamento dei Dati 28
4.3.2 Caricamento dei Metadati 29
4.4 Creazione del Cubo di Dati 29
4.4.1 Tabelle di Dimensione 29
4.4.2 Tabella dei Fatti 29
4.4.3 Integrazione e Caricamento nel Cubo 30

5 Query **31**
5.1 Estrazione dei Dati Vitali 31
5.1.1 Dettagli delle Query 31
5.2 Top 10 Emissioni di CO_2 38
5.3 Top 10 Emissioni di Metano 39
5.4 Correlazione tra Emissioni di Metano e Uso del Suolo Agricolo . . 39
5.5 Top 10 Specie di Uccelli Minacciate 41
5.6 Top 10 Aree Protette 42
5.7 Correlazione tra Emissioni di CO_2 e Crescita della Popolazione . . 43
5.8 Correlazione tra Emissioni di CO_2 e Uso del Carbone 44
5.9 Calcolo dell'Impatto Ambientale degli Alimenti 46
5.10 Dati sulle Emissioni dei Prodotti Alimentari 47
5.11 Aree Protette per Paese 48
5.12 Uso del Suolo Agricolo 49
5.13 Emissioni di PM2.5 50
5.14 Specie di Uccelli Minacciate 51
5.15 Emissioni di CO_2 da Combustibili Solidi 52

5.16 Emissioni nette di GHG 53
5.17 Specie di Pesci Minacciate 53
5.18 Query Interattive 54
5.18.1 Tipologie di Query 54
5.19 Correlazione tra Uso di Energia Rinnovabile ed Emissioni Totali di Gas Serra 55
5.20 Top 10 Emissioni di Gas Serra dai Prodotti Alimentari 58
5.21 Top 10 Uso del Suolo Agricolo 59
5.22 Top 10 Emissioni di PM2.5 60
5.23 Top 10 Indice di Rischio Climatico 61
5.24 Aumento Percentuale Maggiore delle Emissioni di CO2 63
5.25 Variazione delle Emissioni di CO_2 65

6 Tecnologie Utilizzate 68
6.1 Data Warehouse 68
6.2 Linguaggi e Librerie 69
6.2.1 Python per il Processo ETL 69
6.2.2 Flask per le API REST 70
6.2.3 JavaScript per l'Interattività 70
6.3 Motivazioni delle Scelte Tecnologiche 71

7 Conclusioni e Lavori Futuri 72
7.1 Conclusioni 72
7.2 Possibili Miglioramenti 73
7.3 Sviluppi Futuri 74

Capitolo 1

Introduzione

Negli ultimi decenni, i problemi ambientali sono diventati una delle principali preoccupazioni a livello globale. Il cambiamento climatico, la perdita di biodiversità, l'inquinamento dell'aria e dell'acqua, e la degradazione del suolo rappresentano sfide significative per l'umanità e per il pianeta. Questi problemi non solo minacciano l'ambiente naturale, ma hanno anche impatti profondi sulla salute umana, sull'economia e sulla società nel suo complesso.

Il progetto *SOS Earth* nasce con l'obiettivo di aumentare la consapevolezza sui problemi ambientali globali attraverso l'uso di visualizzazioni interattive dei dati. Questo progetto mira a fornire agli utenti uno strumento intuitivo e coinvolgente per esplorare e comprendere meglio le complesse dinamiche dei problemi ambientali. Utilizzando dati accurati e aggiornati, *SOS Earth* permette di visualizzare l'entità e l'evoluzione delle crisi ambientali in diverse regioni del mondo, rendendo più tangibili e comprensibili le connessioni tra i vari fattori.

L'innovatività del progetto risiede nella sua capacità di trasformare i dati grezzi in narrazioni visive che possono essere facilmente comprese da un pubblico ampio e diversificato.

Le visualizzazioni interattive consentono agli utenti di esplorare scenari futuri, analizzare le tendenze storiche e valutare l'impatto delle politiche ambientali, facilitando una maggiore partecipazione e consapevolezza.

1.1 Obiettivi del Progetto

Il principale obiettivo del progetto è quello di educare e sensibilizzare il pubblico sui problemi ambientali, fornendo uno strumento che renda accessibili e comprensibili i dati complessi.

Attraverso la visualizzazione interattiva dei dati, il progetto mira a:

- Fornire informazioni accurate e aggiornate sui vari indicatori ambientali.
- Evidenziare le cause e gli effetti dei principali problemi ambientali.
- Facilitare la comprensione delle interconnessioni tra diversi fenomeni ambientali.
- Promuovere un maggiore impegno verso la sostenibilità e la conservazione dell'ambiente.

1.2 Struttura del Documento

Questo documento è strutturato come segue:

- **Introduzione:** Introduzione generale al progetto *SOS Earth*, inclusi gli obiettivi del progetto e una panoramica della struttura del documento.
- **Descrizione del Progetto:** Una panoramica del progetto. Vengono descritte le funzionalità principali del sito web, come la home, la sezione dei problemi, la sezione di aiuto, il progetto, il calcolatore di Foodprint e gli indicatori.
- **Datasets:** Introduzione ai dataset utilizzati nel progetto, le loro fonti, e una descrizione dettagliata di ciascun dataset.
- **Dettagli del Processo ETL:** Una spiegazione approfondita delle fasi di estrazione, trasformazione e caricamento dei dati.
- **Query:** Descrizione delle query principali utilizzate per estrarre i dati.

- **Tecnologie Utilizzate:** Una descrizione delle tecnologie e degli strumenti utilizzati, con le motivazioni per la loro scelta.
- **Risultati e Benefici:** I risultati ottenuti dal progetto e i benefici in termini di impatti ambientali ed educativi.
- **Conclusioni e Lavori Futuri:** Riflessioni finali sul progetto, suggerimenti per possibili miglioramenti e idee per future espansioni. Vengono delineate le conclusioni tratte dal progetto e le potenziali direzioni per sviluppi futuri.

Capitolo 2

Descrizione del Progetto

Questa sezione descrive dettagliatamente cosa è stato fatto nel progetto *SOS Earth*.

2.1 Panoramica

Il progetto *SOS Earth* è stato concepito per affrontare le crescenti preoccupazioni ambientali attraverso l'uso di visualizzazioni interattive dei dati.

L'obiettivo principale è quello di sensibilizzare il pubblico sui problemi ambientali globali, fornendo uno strumento accessibile per esplorare e comprendere vari indicatori ambientali.

Attraverso l'integrazione di dati da 34 dataset diversi, il progetto offre una visione completa e dettagliata delle problematiche ambientali, evidenziando cause, effetti e soluzioni possibili.

2.2 Struttura del Progetto

La struttura del progetto *SOS Earth* è composta da tre componenti principali: frontend, backend e processo ETL.

Il backend del progetto utilizza un data warehouse per integrare e analizzare dati provenienti da 34 dataset distinti, raccolti da una varietà di fonti affidabili e aggiornate. Questi dati includono informazioni su vari indicatori ambientali

critici, coprendo un'ampia gamma di categorie come la biosfera, gli oceani, l'atmosfera, la criosfera, le dimensioni umane, la superficie terrestre, le interazioni Sole-Terra, la Terra solida e l'idrosfera terrestre.

Il processo ETL è fondamentale per la gestione efficace dei dati. La fase di estrazione (Extract) prevede il recupero dei dati dai diversi dataset. La fase di trasformazione (Transform) comporta la pulizia, la normalizzazione e l'arricchimento dei dati per garantire coerenza e qualità. Infine, nella fase di caricamento (Load), i dati trasformati vengono inseriti nel data warehouse, pronti per essere interrogati e analizzati.

Il frontend è una piattaforma web interattiva progettata per fornire agli utenti un'esperienza di esplorazione dei dati intuitiva e informativa. Attraverso grafici dinamici e dettagliati, gli utenti possono visualizzare e interagire con i dati ambientali in modo approfondito. Ogni grafico è corredato da descrizioni esplicative che forniscono contesto e approfondimenti, aiutando a comprendere l'importanza dei dati visualizzati e il loro impatto sull'ambiente e sulla società.

La piattaforma è costruita utilizzando tecnologie moderne di sviluppo web, garantendo una user experience fluida e reattiva. Le visualizzazioni sono generate tramite librerie di grafica avanzate, che permettono di rappresentare i dati in modo chiaro e accattivante. Inoltre, il sistema supporta la personalizzazione delle visualizzazioni, consentendo agli utenti di filtrare e manipolare i dati secondo le proprie esigenze specifiche.

2.3 Funzionalità del Sito *SOS Earth*

Questa sezione fornisce una panoramica delle varie pagine del sito *SOS Earth*, descrivendo le loro funzioni e obiettivi specifici. Ogni pagina è progettata per offrire agli utenti strumenti interattivi e informazioni dettagliate sui problemi ambientali e le soluzioni possibili.

Di seguito vengono presentate le principali pagine del sito:

2.3.1 Home

La pagina *Home* (come illustrato nella Figura 2.1) accoglie gli utenti con una panoramica generale del progetto *SOS Earth*. Qui, gli utenti possono trovare un'introduzione al sito e alle sue principali funzionalità, oltre a collegamenti rapidi alle diverse sezioni. La pagina *Home* serve come punto di partenza per navigare il sito e accedere alle informazioni e agli strumenti disponibili.

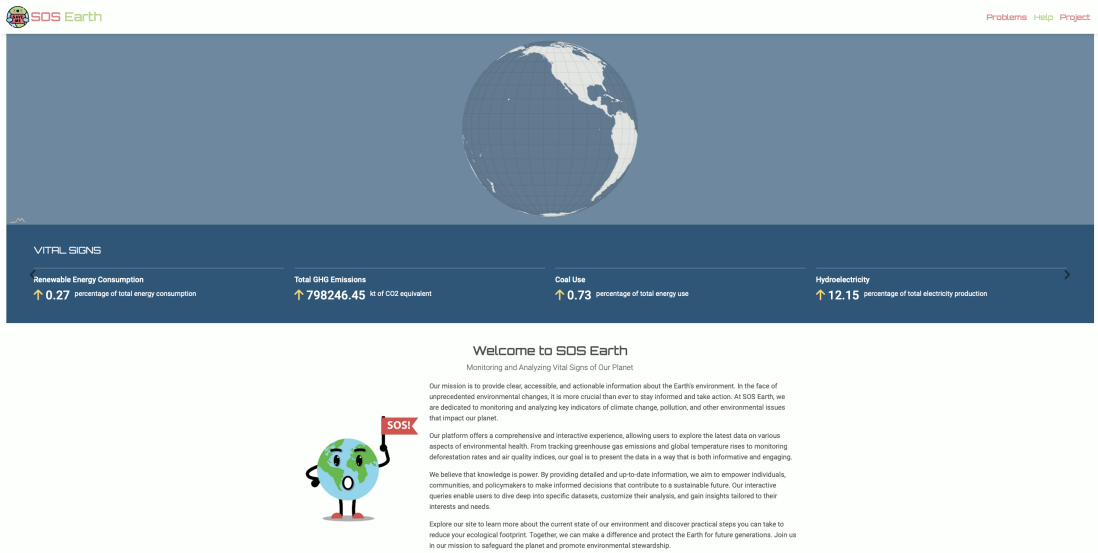

Figura 2.1: Pagina Home

2.3.2 Problem

La pagina *Problem* (come illustrato nella Figura 2.2) offre un'analisi approfondita dei principali problemi ambientali globali. Attraverso dati e visualizzazioni interattive, gli utenti possono esplorare i diversi aspetti delle crisi ambientali, come il cambiamento climatico, l'inquinamento, la deforestazione e la perdita di biodiversità, etc. Questa sezione mira a educare il pubblico sulla gravità e la complessità dei problemi che il nostro pianeta affronta.

2.3.3 Help

La pagina *Help* (come illustrato nella Figura 2.3) è dedicata alla presentazione di possibili soluzioni agli indicatori ambientali critici. Questa sezione della

Figura 2.2: Pagina Problems

piattaforma fornisce raccomandazioni pratiche e strategie per mitigare gli effetti negativi identificati dai vari indicatori ambientali. Gli utenti possono esplorare soluzioni specifiche per problemi come l'inquinamento atmosferico, la perdita di biodiversità e il cambiamento climatico, etc.

2.3.4 Project

La pagina *Project* (come illustrato nella Figura 2.4) include un planisfero interattivo che visualizza le aree protette dei vari paesi. Gli utenti possono cliccare su un paese per visualizzare alcune statistiche sulle sue aree protette, inclusi i parchi nazionali, le riserve naturali e altre zone di conservazione. Questo strumento è utile per educare il pubblico sull'importanza delle aree protette e promuovere la conservazione della biodiversità.

2.3.5 Calcolatore di Foodprint

Il *Calcolatore di Foodprint* (come illustrato nella Figura 2.5) consente agli utenti di calcolare l'impatto di CO_2 dei vari cibi. Questo strumento interattivo aiuta gli utenti a comprendere come le loro scelte alimentari influenzano l'ambiente. Il cal-

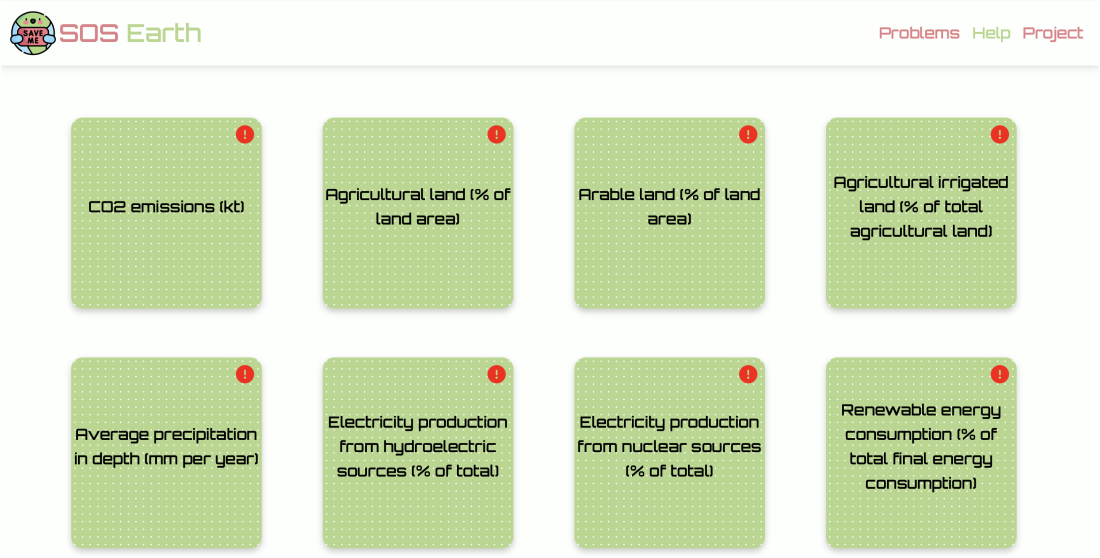

Figura 2.3: Pagina Help

colatore considera vari fattori, tra cui la produzione, il trasporto e la lavorazione degli alimenti.

2.3.6 Indicators

La sezione degli *Indicators* (come illustrato nella Figura 2.6) permette agli utenti di esplorare i dati storici per ciascun indicatore ambientale. Questa funzione offre una visione a lungo termine delle tendenze ambientali, aiutando a identificare cambiamenti e pattern nel tempo. Gli utenti possono visualizzare grafici storici che mostrano l'evoluzione degli indicatori.

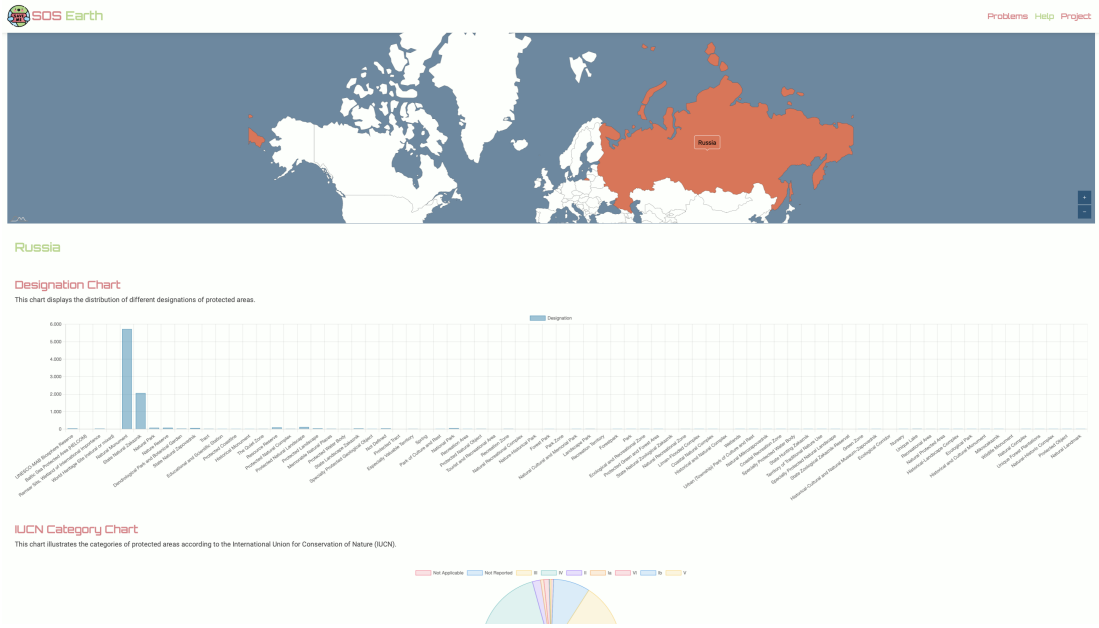

Figura 2.4: Pagina Project

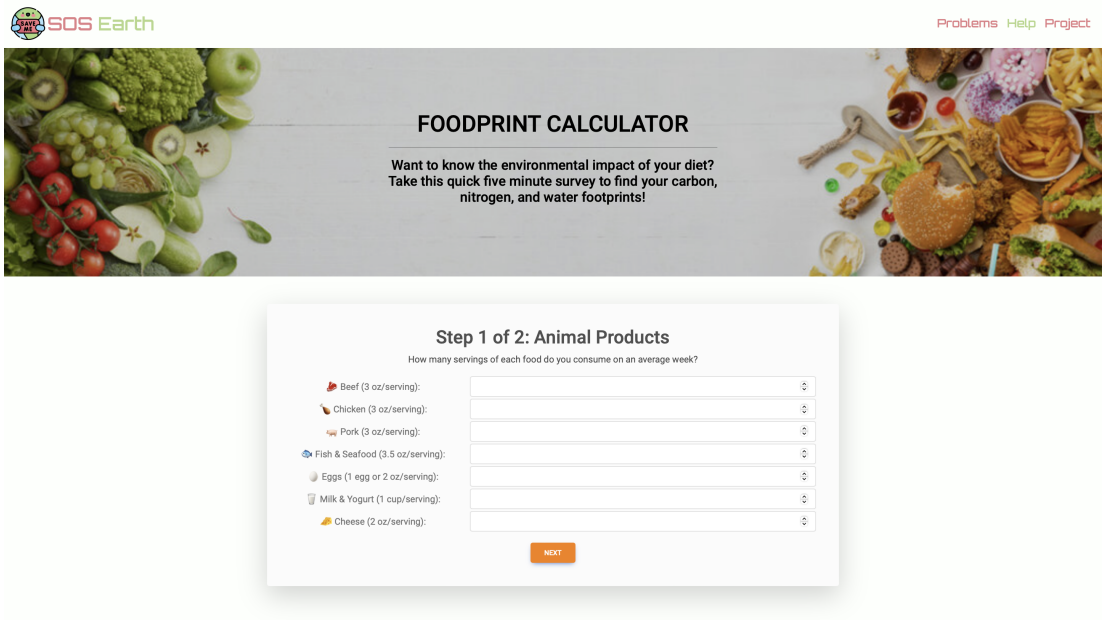

Figura 2.5: Pagina FoodPrintCalculator

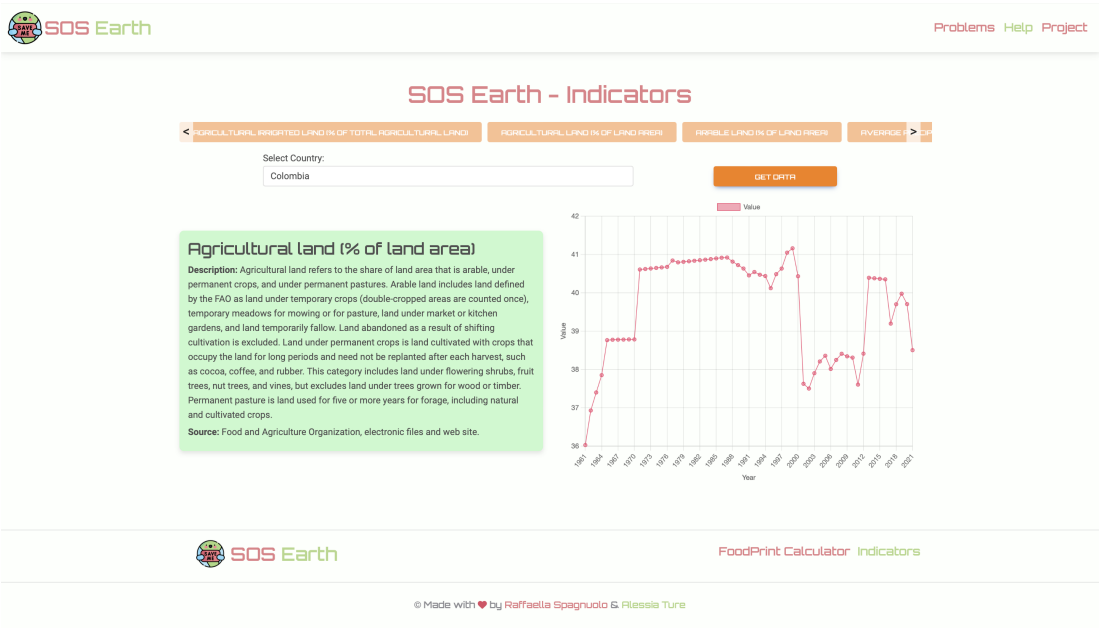

Figura 2.6: Pagina Indicators

Capitolo 3

Datasets

3.1 Introduzione

In questo capitolo, verranno descritti i dataset utilizzati nel progetto *SOS Earth*. Ogni dataset è stato selezionato con attenzione per garantire la qualità e la rilevanza dei dati. I dataset provengono da fonti affidabili e coprono vari aspetti degli indicatori ambientali critici. La loro integrazione e analisi sono fondamentali per fornire visualizzazioni accurate e informative nella piattaforma.

3.2 Fonti dei Dati

I dati utilizzati nel progetto provengono da una varietà di fonti internazionali e nazionali, tra cui:

- **World Bank:** Dati economici e sociali relativi all'ambiente. I dataset presi da questo sito sono composti da tre file distinti: un file di metadati degli indicatori, un file di metadati dei paesi e il dataset vero e proprio. Quest'ultimo file presenta i paesi come righe e gli anni come colonne, fornendo così un quadro dettagliato delle variabili ambientali nel tempo. [1]

- **National Aeronautics and Space Administration (NASA**:) Dati satellitari sull'atmosfera, la criosfera e la superficie terrestre. I dati presi dalla

[1] `https://data.worldbank.org`

NASA sono contenuti in file testuali che forniscono informazioni dettagliate anno per anno e mese per mese, con valori medi e incertezze associate per ciascun periodo. [2]

- **International Union for Conservation of Nature (IUCN):** Dati sulle aree protette e le categorie di conservazione.[3] [4]
- **Our World In Data (OWID):** Questa piattaforma fornisce dati sugli impatti ambientali della produzione alimentare, comprese le emissioni di gas serra associate ai vari alimenti. Questi dati aiutano a comprendere e mitigare l'impatto della produzione alimentare sull'ambiente. [5]

3.3 Descrizione dei Dataset

Ogni dataset è stato selezionato per coprire specifici indicatori ambientali. Di seguito vengono descritti i dataset utilizzati:

3.3.1 Dati Economici e Sociali (World Bank)

Questo dataset fornisce informazioni su indicatori economici e sociali che influenzano e sono influenzati dai cambiamenti ambientali. Include dati su emissioni di CO_2, consumo energetico, uso del suolo, politiche ambientali, etc.

I dataset sono formati da 266 righe e 69 colonne, coprendono un periodo che va dal 1960 al 2020. Le colonne includono il nome del paese, il codice del paese, il nome dell'indicatore, il codice dell'indicatore e i valori annuali per ciascun indicatore. I tipi di dati sono principalmente stringhe per identificare i paesi e gli indicatori, e numerici per i valori annuali.

[2] `https://science.nasa.gov/climate-change/`
[3] `https://www.kaggle.com/datasets/mexwell/world-database-on-protected-areas`
[4] `https://www.kaggle.com/datasets/mexwell/world-database-on-protected-areas`
[5] `https://www.kaggle.com/datasets/amandaroseknudsen/foodproductemissions`

Dettagli dati dataset

- **Agricultural Land (% of land area):** Questo indicatore misura la percentuale di superficie terrestre destinata all'agricoltura, compresi i terreni arabili, i terreni destinati alle colture permanenti e i prati permanenti. È un parametro chiave per valutare l'uso del suolo e la sostenibilità delle pratiche agricole in diversi paesi.

- **Arable Land (% of land area):** Questo indicatore rappresenta la percentuale di superficie terrestre che è arabile e adatta alla coltivazione di colture temporanee. Include anche i terreni che richiedono nuove coltivazioni o terreni incolti utilizzati per la produzione agricola. È un indicatore importante per comprendere la capacità produttiva agricola di una nazione.

- **Forest Area (sq. km):** Questo indicatore misura l'estensione delle aree forestali in chilometri quadrati. Le foreste sono cruciali per la biodiversità, la regolazione del clima e la fornitura di servizi ecosistemici. Monitorare le aree forestali aiuta a valutare gli sforzi di conservazione e i cambiamenti nell'uso del suolo.

- **Irrigated Land (% of total agricultural land):** Questo indicatore quantifica la percentuale di terreno agricolo totale che è irrigato. L'irrigazione è essenziale per migliorare la produttività agricola, specialmente in regioni aride e semi-aride. Fornisce informazioni sull'intensità dell'uso delle risorse idriche per l'agricoltura.

- **Precipitation (mm per year):** Questo indicatore misura la quantità media annua di precipitazioni in millimetri. Le precipitazioni sono un fattore critico per la disponibilità di risorse idriche, la produttività agricola e la sostenibilità degli ecosistemi naturali.

- **Nuclear Electricity (% of total electricity):** Questo indicatore misura la percentuale di energia elettrica totale generata da impianti nucleari. L'energia nucleare è una fonte significativa di energia a basse emissioni

di carbonio e questo indicatore fornisce informazioni sull'uso dell'energia nucleare in diversi paesi.

- **Renewable Energy Consumption (% of total final energy consumption):** Questo indicatore quantifica la percentuale del consumo finale totale di energia che proviene da fonti rinnovabili, inclusi solare, eolico, geotermico e biomassa. È un parametro fondamentale per valutare i progressi verso una transizione energetica sostenibile.

- **Energy Use (kg of oil equivalent per capita)**: Questo indicatore misura il consumo energetico totale per capita in chilogrammi di equivalente petrolio. Include l'energia utilizzata in vari settori come il trasporto, l'industria, le abitazioni e il commercio. È un indicatore chiave per comprendere i modelli di consumo energetico e l'efficienza energetica.

- CO_2 **Emissions (kt):** Questo indicatore misura le emissioni di anidride carbonica in chilotonnellate. Le emissioni di CO_2 sono il principale contributore al cambiamento climatico e questo indicatore è cruciale per monitorare gli sforzi di mitigazione delle emissioni di gas serra.

- CO_2 **Emissions (metric tons per capita):** Questo indicatore misura le emissioni di anidride carbonica per persona in tonnellate metriche. Le emissioni pro capite di CO_2 sono un indicatore cruciale per comprendere l'impatto di ciascun individuo sul cambiamento climatico e aiutano a monitorare i progressi verso la riduzione delle emissioni di gas serra.

- **Total Greenhouse Gas Emissions (kt of CO_2 equivalent):** Questo indicatore quantifica le emissioni totali di gas serra in chilotonnellate di equivalente CO_2. Include gas come metano, protossido di azoto e altri gas serra, fornendo una visione complessiva delle emissioni che contribuiscono al cambiamento climatico.

- **Methane Emissions (kt of CO_2 equivalent):** Questo indicatore misura le emissioni di metano in chilotonnellate di equivalente CO_2. Il metano è

un potente gas serra e questo indicatore è importante per valutare le fonti di emissioni di metano e i loro contributi al riscaldamento globale.

- **Nitrous Oxide Emissions (kt of CO_2 equivalent):** Questo indicatore rappresenta le emissioni di protossido di azoto in chilotonnellate di equivalente CO_2. Il protossido di azoto è un altro gas serra significativo e monitorare le sue emissioni aiuta a comprendere il suo impatto sul clima.

- **PM2.5 Air Pollution (micrograms per cubic meter):** Questo indicatore misura la concentrazione di particolato fine (PM2.5) nell'aria in microgrammi per metro cubo. L'inquinamento da PM2.5 è dannoso per la salute umana, causando malattie respiratorie e cardiovascolari, e monitorare questo indicatore è essenziale per la gestione della qualità dell'aria.

- **Number of Threatened Bird Species:** Questo indicatore conta il numero di specie di uccelli minacciate. La perdita di biodiversità è un problema critico e questo indicatore aiuta a monitorare lo stato di conservazione degli uccelli e gli sforzi necessari per proteggerli.

- **Disaster Risk Reduction Progress Score:** Questo indicatore valuta il punteggio di progresso nella riduzione del rischio di disastri. Un punteggio più alto indica migliori capacità di gestione e mitigazione dei rischi legati ai disastri naturali.

- **Greenhouse Gas Emissions from Energy (million metric tons of CO_2 equivalent):** Questo indicatore misura le emissioni di gas serra derivanti dal settore energetico in milioni di tonnellate metriche di equivalente CO_2. È importante per monitorare l'impatto del consumo di energia sulle emissioni totali di gas serra.

- **Population Affected by Natural Disasters (% of total population):** Questo indicatore rappresenta la percentuale della popolazione totale colpita da disastri naturali. Fornisce informazioni sull'impatto umano dei disastri e aiuta a identificare le aree più vulnerabili.

- **Number of Threatened Fish Species:** Questo indicatore conta il numero di specie di pesci minacciate. Monitorare le specie di pesci minacciate è fondamentale per la conservazione della biodiversità marina e per la gestione sostenibile delle risorse ittiche.

- **Number of Threatened Amphibian Species:** Questo indicatore misura il numero di specie di anfibi minacciate. La perdita di biodiversità tra gli anfibi è un problema critico, poiché questi animali sono spesso indicatori della salute degli ecosistemi.

- **Population Ages 0-14 (% of total population):** Questo indicatore rappresenta la percentuale della popolazione totale costituita da bambini e ragazzi di età compresa tra 0 e 14 anni. È un indicatore demografico importante per comprendere la struttura della popolazione e le future esigenze di risorse e servizi.

- **Urban Population in the Largest City (% of total population):** Questo indicatore misura la percentuale della popolazione totale che vive nella città più grande del paese. Fornisce informazioni sulla distribuzione della popolazione urbana e sulle dinamiche di urbanizzazione.

- **Internal Renewable Water Resources (cubic kilometers):** Questo indicatore quantifica le risorse idriche rinnovabili interne in chilometri cubi. Include le risorse idriche generate dalle precipitazioni e altre fonti interne, ed è fondamentale per la gestione delle risorse idriche di un paese.

- **Terrestrial and Marine Protected Areas (% of total territorial area):** Questo indicatore misura la percentuale dell'area territoriale totale che è protetta, sia terrestre che marina. Le aree protette sono cruciali per la conservazione della biodiversità e degli ecosistemi.

- **Population Growth (annual %):** Questo indicatore rappresenta la crescita annuale della popolazione in percentuale. È un indicatore chiave per comprendere le dinamiche demografiche e pianificare lo sviluppo sostenibile.

- **Hydroelectricity (% of total electricity):**Questo indicatore rappresenta la percentuale di energia elettrica totale prodotta da impianti idroelettrici. L'energia idroelettrica è una fonte rinnovabile e sostenibile di energia, e il monitoraggio di questo indicatore aiuta a valutare la dipendenza di un paese da fonti energetiche rinnovabili.

3.3.2 Dati Atmosferici e Climatologici (NASA)

Questi dataset includono una vasta gamma di dati satellitari sulla composizione atmosferica, la temperatura, le precipitazioni e altri parametri climatici. Questi dati sono essenziali per monitorare il cambiamento climatico e i suoi effetti.

Dettagli dei Dataset

- **Dati sulla Temperatura:** Questo file di testo contiene l'indice di temperatura terra-oceano dal 1880 al 2023. I dati sono forniti sia in forma grezza, senza livellamento, sia livellati utilizzando il metodo Lowess (regressione locale). Questa doppia presentazione consente di analizzare sia le variazioni annuali grezze che le tendenze a lungo termine. Ad esempio, i dati mostrano come le temperature medie globali siano aumentate progressivamente, fornendo un chiaro indicatore del riscaldamento globale. Il dataset è fondamentale per monitorare le tendenze climatiche e valutare l'impatto delle attività umane sul clima.

- **Dati sul Metano:** Questi dati mensili globali coprono il periodo dal 1983 al 2013 e mostrano le concentrazioni di metano in atmosfera, espresse in parti per miliardo (ppb). Questi dati sono cruciali per monitorare le emissioni di metano, un potente gas serra, e comprendere il suo contributo al cambiamento climatico. La presenza di incertezze associate ai valori medi mensili, indicate nel dataset, permette di valutare la precisione delle misurazioni e l'affidabilità delle tendenze osservate.

- **Dati sulla CO_2:** Questo file contiene dati mensili sulla concentrazione di anidride carbonica, misurata a Mauna Loa dal 1958 al 2023. I dati sono

espressi in parti per milione (ppm) e sono fondamentali per comprendere come le concentrazioni di CO_2 siano aumentate nel tempo, evidenziando l'impatto delle attività umane, come la combustione di combustibili fossili, sull'atmosfera terrestre. Le concentrazioni di CO_2 sono un indicatore chiave del cambiamento climatico, e questi dati sono essenziali per modellare le previsioni climatiche future.

- **Global Mean Sea Level (GMSL) da Satellite:** Il livello medio globale del mare è un indicatore chiave per monitorare i cambiamenti climatici globali. Questo dataset fornisce una serie temporale delle variazioni del livello del mare rilevate da misurazioni satellitari dal 1992 al 2023. Le variazioni del GMSL sono influenzate da diversi fattori, tra cui la fusione dei ghiacciai e l'espansione termica dell'acqua oceanica dovuta al riscaldamento globale. Monitorare queste variazioni è cruciale per comprendere l'impatto del cambiamento climatico sugli ecosistemi costieri e sulle comunità umane.

Questi dataset, con la loro ricchezza di dati e la loro lunga copertura temporale, rappresentano strumenti fondamentali per la ricerca e l'analisi del cambiamento climatico. Essi offrono una base solida per studi sulle tendenze climatiche globali e sulle politiche ambientali necessarie per affrontare le sfide future. La disponibilità di dati precisi e dettagliati consente agli scienziati e ai decisori politici di prendere decisioni informate per mitigare gli effetti del cambiamento climatico e adattarsi alle sue conseguenze.

3.3.3 Dati sulle Aree Protette (IUCN)

Il dataset dell'IUCN fornisce informazioni dettagliate sulle aree protette in tutto il mondo, comprese le categorie di conservazione e le designazioni specifiche.

Il dataset contiene un'enorme quantità di informazioni, con 285,418 righe e 31 colonne. Le colonne coprono una vasta gamma di dati, tra cui il tipo di area protetta (ad esempio, terrestre o marina), l'identificativo unico dell'area, il nome e la designazione dell'area, la categoria IUCN, lo stato e l'anno di istituzione, e i dettagli sulla gestione e la proprietà.

Ecco una descrizione più dettagliata di alcune delle colonne principali:

- **Tipo di Area Protetta:**Indica se l'area è terrestre o marina, o un altro tipo di protezione specifico.
- **Nome e Designazione:** Forniscono il nome ufficiale e la designazione dell'area protetta, sia nella lingua originale che in inglese
- **Categoria IUCN:** Classifica l'area protetta secondo le categorie di conservazione dell'IUCN, che vanno da Ia (Riserve naturali strettamente protette) a VI (Aree protette con uso sostenibile delle risorse naturali).
- **Superficie:** Indicano la superficie dell'area protetta in chilometri quadrati, sia per l'area rappresentativa che per quella calcolata con sistemi GIS.
- **Gestione e Proprietà:** Dettagli sul tipo di gestione (ad esempio, nazionale, regionale, privata) e sul tipo di proprietà dell'area.

Questi dati sono utilizzati per diverse applicazioni, come la visualizzazione delle aree protette su mappe interattive, l'analisi della distribuzione e dell'efficacia delle aree protette, e la pianificazione della conservazione. Ad esempio, si può analizzare come le diverse categorie di conservazione IUCN sono distribuite nelle varie regioni del mondo, o confrontare la quantità di superficie protetta terrestre rispetto a quella marina.

Inoltre, i dati sulla gestione e la proprietà sono fondamentali per comprendere chi è responsabile della protezione e della manutenzione di queste aree, e come sono implementate le politiche di conservazione.

3.3.4 Dati sugli Impatti della Produzione Alimentare (Our World In Data)

Il dataset di *Our World In Data* fornisce una panoramica dettagliata sugli impatti ambientali della produzione alimentare. Include informazioni cruciali come le emissioni di gas serra, l'uso delle risorse idriche, il consumo di terra e altri fattori legati alla produzione e al consumo di cibo. Questi dati sono essenziali

per comprendere l'impatto delle scelte alimentari sull'ambiente e per promuovere pratiche più sostenibili.

Il dataset contiene 43 righe e 23 colonne, ognuna delle quali rappresenta un aspetto diverso dell'impatto ambientale della produzione alimentare. Ecco alcuni dei principali elementi inclusi nel dataset:

- **Prodotto Alimentare:** Il nome del prodotto alimentare, ad esempio carne di manzo, riso, latte, ecc.
- **Cambiamento nell'uso del Suolo:** Il cambiamento nell'uso del suolo associato alla produzione del prodotto. Questo può includere la conversione di foreste in terreni agricoli, che ha un impatto significativo sull'ambiente.
- **Emissioni di Gas Serra:** Le emissioni di gas serra sono suddivise in diverse categorie come l'alimentazione animale, le emissioni derivanti dall'attività agricola diretta, la lavorazione, il trasporto, il confezionamento e la vendita al dettaglio. Inoltre, viene fornita una somma delle emissioni totali per ciascun prodotto.
- **Uso dell'Acqua:** I dati sull'uso dell'acqua dolce e l'uso ponderato delle risorse idriche scarse mostrano quanta acqua è necessaria per produrre una certa quantità di cibo, sia per 1000 kcal, per 100 grammi di proteine, che per chilogrammo di prodotto.
- **Uso del Suolo:** Questo parametro misura la quantità di terra necessaria per produrre il cibo, sia per 1000 kcal, per chilogrammo di prodotto, che per 100 grammi di proteine.
- **Emissioni Eutrofizzanti:** Le emissioni eutrofizzanti, espresse in grammi di equivalente fosfato, indicano il potenziale del prodotto di contribuire alla proliferazione di alghe nei corpi idrici, un fenomeno che può danneggiare gli ecosistemi acquatici.

Questi dati possono essere utilizzati per analizzare e confrontare l'impatto ambientale di diversi prodotti alimentari. Ad esempio, è possibile confrontare le

emissioni di gas serra associate alla produzione di carne rispetto a quelle della produzione di verdure. Inoltre, si può valutare l'uso delle risorse idriche per la produzione di cereali rispetto alla produzione di frutta, identificando così le scelte alimentari più sostenibili.

Un esempio di analisi potrebbe essere il calcolo delle emissioni totali di gas serra per una dieta specifica. Questo potrebbe aiutare a identificare quali alimenti contribuiscono maggiormente all'impronta di carbonio di una persona e suggerire alternative più sostenibili. Un'altra analisi potrebbe concentrarsi sul consumo di acqua dolce, aiutando a capire quali prodotti alimentari richiedono più risorse idriche e quindi potrebbero essere meno sostenibili in regioni con scarsità d'acqua.

3.3.5 Soluzioni ai Problemi Ambientali

Questo dataset elenca diverse soluzioni per affrontare i problemi ambientali, come la tassazione del carbonio per ridurre le emissioni di CO_2. Ogni soluzione è accompagnata da prove e cause che ne giustificano l'implementazione.

Il dataset contiene 23 righe e 2 colonne. Ogni riga rappresenta una soluzione a un problema ambientale specifico. Le colonne includono:

- **Indicator Name:** Il nome dell'indicatore ambientale o del problema che viene affrontato. Ad esempio, potrebbe trattarsi delle emissioni di carbonio, della deforestazione o dell'inquinamento delle acque.

- **Solution:** La descrizione della soluzione proposta per affrontare il problema. Questo potrebbe includere politiche come la tassazione del carbonio, il rewilding (ripristino di ecosistemi naturali), l'adozione di tecnologie verdi, ecc.

Questi dati sono utili per identificare e analizzare diverse strategie per mitigare i problemi ambientali. Ad esempio, si può esaminare l'efficacia delle tasse sul carbonio nel ridurre le emissioni di CO_2 o esplorare altre soluzioni come la conservazione delle foreste per assorbire carbonio e preservare la biodiversità.

3.3.6 Correlazioni dei Sistemi Terrestri

Questo dataset contiene dati sulle correlazioni tra vari indicatori dei sistemi terrestri, come la temperatura, le precipitazioni e le concentrazioni di gas serra. Questi dati sono essenziali per comprendere le interazioni complesse tra i diversi componenti del sistema terrestre.

Il dataset contiene 9 righe e 4 colonne, con informazioni dettagliate sulle correlazioni tra vari indicatori dei sistemi terrestri. Le colonne includono:

- **Category:** La categoria del sistema terrestre, come la temperatura, le precipitazioni, o i gas serra. Evidence: Le prove che supportano l'esistenza della correlazione. Questo potrebbe includere dati storici, studi scientifici o modelli climatici.

- **Causes:** Le cause che determinano la correlazione. Ad esempio, l'aumento delle concentrazioni di CO_2 può causare un aumento della temperatura globale.

- **Effects:** Gli effetti osservati delle correlazioni. Questi potrebbero includere eventi climatici estremi, cambiamenti nei pattern delle precipitazioni, scioglimento dei ghiacci, ecc.

Capitolo 4

Dettagli del Processo ETL

Questo capitolo fornisce una descrizione dettagliata del processo ETL (Extract, Transform, Load) utilizzato nel progetto *SOS Earth* per gestire i dati ambientali raccolti da diverse fonti. Inoltre, viene descritto il processo di creazione del cubo di dati, essenziale per le analisi multidimensionali.

4.1 Estrazione (Extract)

L'estrazione dei dati è il primo passo cruciale del processo ETL. In questo progetto, i dati sono stati estratti da una varietà di fonti affidabili.

Per ogni dataset, è stato necessario identificare il formato specifico e le caratteristiche del file, come il delimitatore utilizzato, la presenza di righe di intestazione o commenti, e l'eventuale necessità di saltare alcune righe iniziali. Questo ha permesso di configurare correttamente il processo di estrazione per garantire che tutti i dati rilevanti fossero letti e importati correttamente nei nostri strumenti di analisi. Ad esempio, i dati sulle temperature globali sono stati estratti da un file di testo con delimitatori specifici, mentre i dati sulle emissioni di CO_2 sono stati estratti da file CSV con intestazioni complesse.

4.2 Trasformazione (Transform)

La trasformazione dei dati rappresenta la fase più complessa e delicata del processo ETL. Una volta estratti, i dati grezzi spesso contengono duplicati, valori mancanti, e formattazioni incoerenti. Pertanto, il primo passo nella trasformazione è stato quello di pulire i dati, rimuovendo duplicati e colonne non necessarie. Questo processo è fondamentale per garantire che i dati siano accurati e utilizzabili per le analisi successive.

Successivamente, i dati sono stati riformattati per garantire coerenza e completezza. Ad esempio, i nomi delle colonne sono stati standardizzati, e le date sono state convertite in un formato uniforme. Per gli indicatori ambientali, i dati sono stati arricchiti con metadati rilevanti, come informazioni specifiche sui paesi e sugli indicatori stessi. Questo arricchimento ha permesso di aggiungere contesto ai dati, facilitando l'analisi successiva.

Un aspetto cruciale della trasformazione è stato l'adattamento dei dati per includere informazioni sui paesi e sugli anni. Questo ha richiesto una mappatura accurata tra i dati estratti e i metadati dei paesi, che includevano codici paese standardizzati e altre informazioni pertinenti. Per i dati annuali, è stato necessario assicurarsi che ogni record includesse un anno specifico, trasformando le serie temporali in un formato che potesse essere facilmente analizzato e visualizzato.

4.2.1 Trasformazioni Specifiche per Dataset

Alcuni dataset hanno richiesto trasformazioni specifiche. Ad esempio:

- **Dati di Temperatura**: I dati sulle temperature globali sono stati trasformati per includere solo gli anni numerici e le temperature senza smoothing. Questo ha permesso di analizzare le tendenze delle temperature nel tempo con maggiore precisione.

- **Dati di Emissioni di CH_4**: Le colonne nei dati delle emissioni di metano sono state rinominate per riflettere chiaramente l'anno, il mese, e la

concentrazione di CH_4. Questa riformattazione ha facilitato l'analisi delle tendenze annuali delle emissioni di metano.

- **Dati di CO_2**: I dati delle emissioni di CO_2 sono stati formattati per gestire date decimali e concentrazioni di CO_2, rendendo possibile l'analisi dettagliata delle tendenze temporali delle emissioni di CO_2.
- **Dati di GMSL (Global Mean Sea Level)**: I dati sul livello medio del mare sono stati trasformati per includere solo l'anno e il livello del mare, permettendo di monitorare i cambiamenti.
- **Dati sulle Aree Protette**: Le informazioni sulle aree protette sono state riformattate per includere nome, designazione, categoria IUCN, e codice ISO del paese, facilitando l'analisi delle aree protette a livello globale.

4.3 Caricamento (Load)

Il caricamento dei dati nel data warehouse è l'ultima fase del processo ETL. Una volta trasformati, i dati vengono caricati nel database MongoDB. Questa fase assicura che i dati siano pronti per essere interrogati e analizzati in modo efficiente.

4.3.1 Caricamento dei Dati

Ogni dataset trasformato è stato convertito in un formato compatibile con MongoDB e caricato nelle collezioni appropriate. Questo processo ha coinvolto la conversione dei DataFrame di `pandas` in dizionari e l'inserimento di questi dizionari nelle collezioni MongoDB corrispondenti. Per garantire un caricamento efficiente e senza errori, sono state eseguite verifiche sui dati per assicurarsi che non ci fossero duplicati o valori mancanti critici.

4.3.2 Caricamento dei Metadati

Oltre ai dati principali, anche i metadati sono stati caricati nel data warehouse. Questo include metadati sui paesi e sugli indicatori, che sono essenziali per fornire contesto e consentire analisi più approfondite. I metadati sui paesi includono informazioni come codici paese, nomi ufficiali, e altre caratteristiche rilevanti. I metadati sugli indicatori forniscono descrizioni dettagliate di ciascun indicatore, inclusi codici e definizioni. Questi metadati sono stati caricati in collezioni separate all'interno del database, permettendo un facile accesso e collegamento con i dati principali.

4.4 Creazione del Cubo di Dati

La creazione del cubo di dati rappresenta un passaggio cruciale per abilitare le analisi multidimensionali. Un cubo di dati consente di esplorare i dati da diverse prospettive, facilitando l'analisi delle tendenze e delle correlazioni tra variabili.

4.4.1 Tabelle di Dimensione

Per costruire il cubo di dati, sono state create diverse tabelle di dimensione:

- **Tabella delle Dimensioni del Paese**: Questa tabella include informazioni sui paesi, come il nome del paese e il codice del paese. È stata creata a partire dai dati di metadati dei paesi, garantendo che ogni record nel cubo di dati possa essere collegato a un paese specifico.
- **Tabella delle Dimensioni dell'Anno**: Questa tabella contiene gli anni dal 1960 al 2024. Ogni record di dati è stato collegato a un anno specifico, permettendo di analizzare le tendenze temporali.

4.4.2 Tabella dei Fatti

La tabella dei fatti è il cuore del cubo di dati e contiene le misurazioni effettive che si desidera analizzare. In questo progetto, la tabella dei fatti include dati come il

valore degli indicatori ambientali per ciascun paese e anno, insieme ai metadati pertinenti. La creazione della tabella dei fatti ha comportato la combinazione dei dati trasformati con le tabelle di dimensione, utilizzando chiavi di collegamento per garantire l'integrità referenziale.

4.4.3 Integrazione e Caricamento nel Cubo

Una volta create le tabelle di dimensione e dei fatti, i dati sono stati caricati nel cubo di dati all'interno del database. Questo processo ha richiesto una serie di operazioni di join e merge per combinare correttamente le tabelle di dimensione con la tabella dei fatti, assicurando che ogni record nel cubo di dati fosse completo e accurato.

Il cubo di dati risultante consente analisi multidimensionali, permettendo agli utenti di esplorare i dati da diverse angolazioni, come il tempo, il paese e l'indicatore specifico. Questa flessibilità è essenziale per comprendere meglio le tendenze ambientali e supportare decisioni informate basate sui dati.

Capitolo 5

Query

In questa sezione verranno descritte in dettaglio le query utilizzate per estrarre, elaborare e visualizzare i dati, spiegando i passaggi e le logiche alla base di ciascuna operazione.

5.1 Estrazione dei Dati Vitali

Questa funzione raccoglie i dati più recenti su vari indicatori ambientali. I dati vitali includono informazioni chiave come la concentrazione di CO_2, le emissioni di metano, le emissioni di ossido di azoto, l'inquinamento da PM2.5, il consumo di energia rinnovabile, le emissioni totali di gas serra, l'uso del carbone, la produzione idroelettrica, le specie di uccelli minacciate, l'indice di rischio climatico, l'estensione minima del ghiaccio marino artico e il livello medio globale del mare.

5.1.1 Dettagli delle Query

CO_2 Concentration

La query cerca l'ultima misurazione della concentrazione di CO_2 nella collezione `co2_data`, ordinando i dati in base alla data decimale in ordine decrescente. Se i dati sono presenti, vengono aggiunti alla lista dei segni vitali con un titolo, valore, tendenza e descrizione appropriati.

Listing 5.1: Query per ottenere i dati

```
co2_data = mongo.db.co2_data.find_one(sort=[("decimal_date", -1)])
if co2_data:
    vital_signs.append({
        "title": "CO2 Concentration",
        "value": co2_data.get("co2_concentration", "N/A"),
        "trend": "up",
        "description": "parts per million"
    })
```

Methane Emissions

Simile alla query precedente, questa query cerca l'ultima misurazione delle emissioni di metano nella collezione `methane_emissions`. I dati vengono ordinati in base alla data decimale e, se trovati, vengono aggiunti alla lista dei segni vitali.

Listing 5.2: Query per ottenere i dati

```
methane_data =
    mongo.db.methane_emissions.find_one(sort=[("decimal_date", -1)])
if methane_data:
    vital_signs.append({
        "title": "Methane Emissions",
        "value": methane_data.get("Value", "N/A"),
        "trend": "up",
        "description": "parts per billion"
    })
```

Nitrous Oxide Emissions

Questa query cerca l'ultima misurazione delle emissioni di ossido di azoto nella collezione `nitrous_oxide_emissions`, ordinando i dati per anno in ordine decrescente. I dati trovati vengono aggiunti alla lista dei segni vitali con il rispettivo titolo, valore, tendenza e descrizione.

Listing 5.3: Query per ottenere i dati vitali più recenti

```
nitrous_oxide_data =
    mongo.db.nitrous_oxide_emissions.find_one(sort=[("year", -1)])
if nitrous_oxide_data:
    vital_signs.append({
        "title": "Nitrous Oxide Emissions",
        "value": nitrous_oxide_data.get("Value", "N/A"),
        "trend": "up",
        "description": "thousand metric tons of CO2 equivalent"
    })
```

PM2.5 Air Pollution

La query cerca l'ultima misurazione dell'inquinamento da PM2.5 nella collezione `pm25_air_pollution`, ordinando i dati per anno in ordine decrescente. Se i dati sono presenti, vengono aggiunti alla lista dei segni vitali.

Listing 5.4: Query per ottenere i dati vitali più recenti

```
pm25_data = mongo.db.pm25_air_pollution.find_one(sort=[("year",
    -1)])
if pm25_data:
    vital_signs.append({
        "title": "PM2.5 Air Pollution",
        "value": pm25_data.get("Value", "N/A"),
        "trend": "up",
        "description": "micrograms per cubic meter"
    })
```

Renewable Energy Consumption

Questa query cerca l'ultima misurazione del consumo di energia rinnovabile nella collezione `renewable_energy`, ordinando i dati per anno in ordine decrescente. I dati trovati vengono aggiunti alla lista dei segni vitali.

Listing 5.5: Query per ottenere i dati vitali più recenti

```
renewable_energy_data =
    mongo.db.renewable_energy.find_one(sort=[("year", -1)])
if renewable_energy_data:
    vital_signs.append({
        "title": "Renewable Energy Consumption",
        "value": renewable_energy_data.get("Value", "N/A"),
        "trend": "up",
        "description": "percentage of total energy consumption"
    })
```

Total GHG Emissions

La query cerca l'ultima misurazione delle emissioni totali di gas serra nella collezione `total_ghg_emissions`, ordinando i dati per anno in ordine decrescente. I dati trovati vengono aggiunti alla lista dei segni vitali.

Listing 5.6: Query per ottenere i dati vitali più recenti

```
total_ghg_emissions_data =
    mongo.db.total_ghg_emissions.find_one(sort=[("year", -1)])
if total_ghg_emissions_data:
    vital_signs.append({
        "title": "Total GHG Emissions",
        "value": total_ghg_emissions_data.get("Value", "N/A"),
        "trend": "up",
        "description": "kt of CO2 equivalent"
    })
```

Coal Use

La query cerca l'ultima misurazione dell'uso del carbone nella collezione `coal_use`, ordinando i dati per anno in ordine decrescente. Se i dati sono presenti, vengono aggiunti alla lista dei segni vitali.

Listing 5.7: Query per ottenere i dati vitali più recenti

```
coal_use_data = mongo.db.coal_use.find_one(sort=[("year", -1)])
if coal_use_data:
    vital_signs.append({
        "title": "Coal Use",
        "value": coal_use_data.get("Value", "N/A"),
        "trend": "up",
        "description": "percentage of total energy use"
    })
```

Hydroelectricity

Questa query cerca l'ultima misurazione della produzione di elettricità da fonti idroelettriche nella collezione `hydroelectricity`, ordinando i dati per anno in ordine decrescente. I dati trovati vengono aggiunti alla lista dei segni vitali.

Listing 5.8: Query per ottenere i dati vitali più recenti

```
hydroelectricity_data =
    mongo.db.hydroelectricity.find_one(sort=[("year", -1)])
if hydroelectricity_data:
    vital_signs.append({
        "title": "Hydroelectricity",
        "value": hydroelectricity_data.get("Value", "N/A"),
        "trend": "up",
        "description": "percentage of total electricity production"
    })
```

Threatened Bird Species

La query cerca l'ultima misurazione del numero di specie di uccelli minacciate nella collezione `threatened_bird_species`, ordinando i dati per anno in ordine decrescente. I dati trovati vengono aggiunti alla lista dei segni vitali.

Listing 5.9: Query per ottenere i dati vitali più recenti

```
threatened_bird_species_data =
    mongo.db.threatened_bird_species.find_one(sort=[("year", -1)])
if threatened_bird_species_data:
    vital_signs.append({
        "title": "Threatened Bird Species",
        "value": threatened_bird_species_data.get("Value", "N/A"),
        "trend": "up",
        "description": "number of species"
    })
```

Climate Risk Index

La query cerca l'ultima misurazione dell'indice di rischio climatico nella collezione `climate_risk_index`, ordinando i dati per anno in ordine decrescente. I dati trovati vengono aggiunti alla lista dei segni vitali.

Listing 5.10: Query per ottenere i dati vitali più recenti

```
climate_risk_index_data =
    mongo.db.climate_risk_index.find_one(sort=[("year", -1)])
if climate_risk_index_data:
    vital_signs.append({
        "title": "Climate Risk Index",
        "value": climate_risk_index_data.get("Value", "N/A"),
        "trend": "up",
        "description": "index"
    })
```

Arctic Sea Ice Minimum Extent

Questa query cerca l'ultima misurazione dell'estensione minima del ghiaccio marino artico nella collezione `temperature_data`, ordinando i dati per anno in ordine decrescente. Se i dati sono presenti, vengono aggiunti alla lista dei segni vitali.

Listing 5.11: Query per ottenere i dati vitali più recenti

```
arctic_ice_data = mongo.db.temperature_data.find_one(sort=[("year",
    -1)])
if arctic_ice_data:
    vital_signs.append({
        "title": "Arctic Sea Ice Minimum Extent",
        "value": arctic_ice_data.get("temperature", "N/A"),
        "trend": "down",
        "description": "temperature anomaly"
    })
```

Global Mean Sea Level

La query cerca l'ultima misurazione del livello medio globale del mare nella collezione `gmsl_data`, ordinando i dati per anno decimale in ordine decrescente. I dati trovati vengono aggiunti alla lista dei segni vitali.

Listing 5.12: Query per ottenere i dati vitali più recenti

```
gmsl_data = mongo.db.gmsl_data.find_one(sort=[("decimal_year", -1)])
if gmsl_data:
    vital_signs.append({
        "title": "Global Mean Sea Level",
        "value": gmsl_data.get("gmsl", "N/A"),
        "trend": "up",
        "description": "millimeters"
    })

return vital_signs
```

5.2 Top 10 Emissioni di CO_2

Questa query restituisce i primi 10 paesi con le emissioni medie di CO_2 più alte. Utilizza un'aggregazione su MongoDB per calcolare la media delle emissioni di CO_2 per ogni paese e ordina i risultati in modo decrescente.

Listing 5.13: Query per ottenere i primi 10 paesi con le emissioni medie di CO_2 più alte

```
@app.route('/api/top_10_co2_emissions', methods=['GET'])
def get_top_10_co2_emissions():
    try:
        pipeline = [
            {"$group": {"_id": "$Country Name", "avg_co2_emissions":
                {"$avg": "$Value"}}},
             emissioni medie di CO2
            {"$sort": {"avg_co2_emissions": -1}},
            {"$limit": 10}
        ]
        data = list(mongo.db.co2_emissions.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching top 10 CO2 emissions: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

La pipeline di aggregazione è composta dai seguenti passaggi:

- `$group`: Raggruppa i documenti in base al nome del paese (`Country Name`) e calcola la media delle emissioni di CO_2 (`Value`) per ciascun paese. Il risultato di questa operazione è un documento per ogni paese con la media delle sue emissioni di CO_2, salvata nel campo `avg_co2_emissions`.
- `$sort`: Ordina i risultati in ordine decrescente basato sul valore di `avg_co2_emissions`, in modo da avere i paesi con le emissioni medie di CO_2 più alte in cima alla lista.

- `$limit`: Limita il numero di risultati ai primi 10 paesi.

5.3 Top 10 Emissioni di Metano

Questa query restituisce i primi 10 paesi con le emissioni medie di metano più alte.

Listing 5.14: Query per ottenere i primi 10 paesi con le emissioni medie di metano più alte

```
@app.route('/api/top_10_methane_emissions', methods=['GET'])
def get_top_10_methane_emissions():
    try:
        pipeline = [
            {"$group": {"_id": "$Country Name",
                "avg_methane_emissions": {"$avg": "$Value"}}},
            {"$sort": {"avg_methane_emissions": -1}},
            {"$limit": 10}
        ]
        data = list(mongo.db.methane_emissions.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching top 10 methane emissions: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

5.4 Correlazione tra Emissioni di Metano e Uso del Suolo Agricolo

Questa query calcola la correlazione tra le emissioni di metano e l'uso del suolo agricolo.

Listing 5.15: Query per ottenere la correlazione tra emissioni di metano e uso del suolo agricolo

```
@app.route('/api/correlation_methane_agricultural_land',
    methods=['GET'])
def get_correlation_methane_agricultural_land():
    try:
        pipeline = [
            {"$match": {"Year": {"$gte": 2000, "$lte": 2018}}},
            {"

$addFields": {"CountryYear": {"$concat": ["$Country Name", "-",
    {"$toString": "$Year"}]}}},
            {"$lookup": {
                "from": "agricultural_land",
                "let": {"country_year": "$CountryYear"},
                "pipeline": [
                    {"$addFields": {"CountryYear": {"$concat":
                        ["$Country Name", "-", {"$toString": "$Year"}]}}},
                    {"$match": {"$expr": {"$eq": ["$CountryYear",
                        "$$country_year"]}}}
                ],
                "as": "agricultural_land_data"
            }},
            {"$unwind": "$agricultural_land_data"},
            {"$project": {
                "Country": "$Country Name",
                "Year": "$Year",
                "Methane_Emissions": "$Value",
                "Agricultural_Land_Use": "$agricultural_land_data.Value"
            }},
            {"$match": {"Methane_Emissions": {"$ne": None},
                "Agricultural_Land_Use": {"$ne": None}}}
        ]
```

```
        data = list(mongo.db.methane_emissions.aggregate(pipeline))
        for document in data:
            document.pop('_id', None)
            document.pop('agricultural_land_data', None)
        if not data:
            return jsonify(
                {"error": "No data found for the correlation between
                    methane emissions and agricultural land use"}), 404
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching methane emissions and
            agricultural land use correlation data: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

5.5 Top 10 Specie di Uccelli Minacciate

Questa query restituisce i primi 10 paesi con il maggior numero di specie di uccelli minacciate.

Listing 5.16: Query per ottenere i primi 10 paesi con il maggior numero di specie di uccelli minacciate

```
@app.route('/api/top_10_threatened_bird_species', methods=['GET'])
def get_top_10_threatened_bird_species():
    try:
        pipeline = [
            {"$match": {"Year": 2018}},
            {"$sort": {"Value": -1}},
            {"$limit": 10},
            {"$project": {"Country": "$Country Name",
                "Threatened_Bird_Species": "$Value"}}
        ]
        data =
            list(mongo.db.threatened_bird_species.aggregate(pipeline))
```

```
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching top 10 threatened bird species:
            {e}")
        return jsonify({"error": "Internal server error"}), 500
```

5.6 Top 10 Aree Protette

Questa query restituisce i primi 10 paesi con la maggiore superficie di aree protette.

Listing 5.17: Query per ottenere i primi 10 paesi con la maggiore superficie di aree protette

```
@app.route('/api/top_10_protected_land', methods=['GET'])
def get_top_10_protected_land():
    try:
        pipeline = [
            {"$match": {"Year": 2020}},
            {"$sort": {"Value": -1}},
            {"$limit": 10},
            {"$project": {"Country": "$Country Name", "Protected_Area":
                "$Value"}}
        ]
        data = list(mongo.db.protected_areas.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching top 10 protected land: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

5.7 Correlazione tra Emissioni di CO_2 e Crescita della Popolazione

Questa query calcola la correlazione tra le emissioni di CO_2 e la crescita della popolazione.

Listing 5.18: Query per ottenere la correlazione tra emissioni di CO_2 e crescita della popolazione

```
@app.route('/api/correlation_co2_population', methods=['GET'])
def get_correlation_co2_population():
    try:
        pipeline = [
            {"$match": {"Year": {"$gte": 2000, "$lte": 2020}}},
            {"$addFields": {"CountryYear": {"$concat": ["$Country Name",
                "-", {"$toString": "$Year"}]}}},
            {"$lookup": {
                "from": "population_growth",
                "let": {"country_year": "$CountryYear"},
                "pipeline": [
                    {"$addFields": {"CountryYear": {"$concat":
                        ["$Country Name", "-", {"$toString": "$Year"}]}}},
                    {"$match": {"$expr": {"$eq": ["$CountryYear",
                        "$$country_year"]}}}
                ],
                "as": "population_growth_data"
            }},
            {"$unwind": "$population_growth_data"},
            {"$project": {
                "Country": "$Country Name",
                "Year": "$Year",
                "CO2_Emissions": "$Value",
                "Population_Growth": "$population_growth_data.Value"
            }},
```

```
            {"$match": {"CO2_Emissions": {"$ne": None},
                "Population_Growth": {"$ne": None}}}
        ]
        data = list(mongo.db.co2_emissions.aggregate(pipeline))
        for document in data:
            document.pop('_id', None)
            document.pop('population_growth_data', None)
        if not data:
            return jsonify(
                {"error": "No data found for the correlation between CO2
                    emissions and population growth"}), 404
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching CO2 emissions and population
            growth correlation data: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

5.8 Correlazione tra Emissioni di CO_2 e Uso del Carbone

Questa query calcola la correlazione tra le emissioni di CO_2 e l'uso del carbone.

Listing 5.19: Query per ottenere la correlazione tra emissioni di CO_2 e uso del carbone

```
@app.route('/api/correlation_co2_coal_use', methods=['GET'])
def get_correlation_co2_coal_use():
    try:
        pipeline = [
            {"$match": {"Year": {"$gte": 2000, "$lte": 2020}}},
            {"$addFields": {"CountryYear": {"$concat": ["$Country Name",
                "-", {"$toString": "$Year"}]}}},
            {"$lookup": {
```

```
                "from": "coal_use",
                "let": {"country_year": "$CountryYear"},
                "pipeline": [
                    {"$addFields": {"CountryYear": {"$concat":
                        ["$Country Name", "-", {"$toString": "$Year"}]}}},
                    {"$match": {"$expr": {"$eq": ["$CountryYear",
                        "$$country_year"]}}}
                ],
                "as": "coal_use_data"
            }},
            {"$unwind": "$coal_use_data"},
            {"$project": {
                "Country": "$Country Name",
                "Year": "$Year",
                "CO2_Emissions": "$Value",
                "Coal_Use": "$coal_use_data.Value"
            }},
            {"$match": {"CO2_Emissions": {"$ne": None}, "Coal_Use":
                {"$ne": None}}}
        ]
        data = list(mongo.db.co2_emissions.aggregate(pipeline))
        for document in data:
            document.pop('_id', None)
            document.pop('coal_use_data', None)
        if not data:
            return jsonify({"error": "No data found for the correlation
                between CO2 emissions and coal use"}), 404
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching CO2 emissions and coal
use correlation data: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Questa è una panoramica dettagliata delle principali query eseguite nel tuo progetto, insieme a spiegazioni dettagliate su come sono state implementate. Ogni sezione descrive il contesto della query, il suo scopo, e fornisce una spiegazione di come la query viene eseguita, senza includere il codice effettivo nel corpo della documentazione.

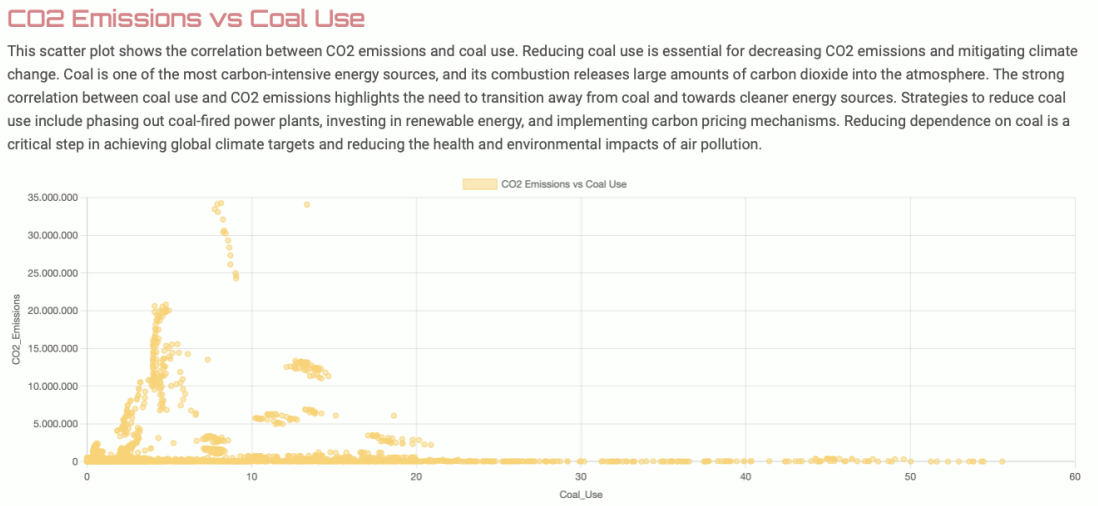

Figura 5.1: Scatterplot

5.9 Calcolo dell'Impatto Ambientale degli Alimenti

Questa query calcola l'impatto ambientale totale dei diversi alimenti basato sulle loro emissioni di gas serra. Prende in input una lista di alimenti con le quantità consumate e restituisce l'impatto totale in termini di emissioni di gas serra.

Listing 5.20: Query per calcolare l'impatto ambientale degli alimenti

```
@app.route('/calculate', methods=['POST'])
def calculate():
    data = request.json
    results = {}
    total_impact = 0.0
```

```
    food_products = ['beef', 'chicken', 'pork', 'fish', 'eggs', 'milk',
        'cheese', 'beans', 'fruits', 'vegetables',
                    'grains', 'rice', 'fats', 'nuts']
    for product in food_products:
        if product in data:
            product_data =
                mongo.db.food_product_emissions.find_one({'food_product':
                product.capitalize()}, {'_id': 0})
            if product_data:
                quantity = data[product]
                product_data['quantity'] = quantity
                product_data['total_impact'] =
                    product_data['total_ghg_emissions_per_kg'] * quantity
                total_impact += product_data['total_impact']
                results[product] = product_data
    return jsonify({'results': results, 'total_impact': total_impact})
```

Questa funzione prende in input un JSON con la quantità di diversi alimenti consumati e calcola l'impatto totale in termini di emissioni di gas serra. La query cerca ogni alimento nel database e moltiplica la quantità per le emissioni di gas serra per kg, sommando l'impatto totale.

5.10 Dati sulle Emissioni dei Prodotti Alimentari

Questa query restituisce i dati sulle emissioni per un prodotto alimentare specificato.

Listing 5.21: Query per ottenere i dati sulle emissioni dei prodotti alimentari

```
@app.route('/get_emissions_data', methods=['GET'])
def get_emissions_data():
    food_product = request.args.get('food_product')
```

```
    if not food_product:
        return jsonify({'error': 'No food product specified'}), 400
    emissions_data =
        mongo.db.food_product_emissions.find_one({'food_product':
        food_product}, {'_id': 0})
    if not emissions_data:
        return jsonify({'error': 'Food product not found'}), 404
    return jsonify(emissions_data)
```

Questa funzione accetta un parametro di query `food_product` e restituisce i dati di emissione per il prodotto alimentare specificato. Se il prodotto non è specificato o non è trovato, viene restituito un errore.

5.11 Aree Protette per Paese

Questa query restituisce i dati delle aree protette per un determinato paese.

Listing 5.22: Query per ottenere i dati delle aree protette per paese

```
@app.route('/country_protected_areas', methods=['GET'])
def country_protected_areas():
    country_name = request.args.get('country_name')
    if not country_name:
        return jsonify({"error": "Country name is required"}), 400
    country_iso3 = country_name_to_iso3.get(country_name)
    if not country_iso3:
        return jsonify({"error": f"ISO3 code not found for country
            {country_name}"}), 404
    protected_areas = list(mongo.db.protected_areas_1.find({"iso3":
        country_iso3}))
    if not protected_areas:
        app.logger.info(f"No protected areas found for country:
            {country_name}")
        return jsonify([])
```

```
    results = []
    for area in protected_areas:
        results.append({
            "name": area.get("name", "No Name Available"),
            "designation": area.get("designation", "Not Available"),
            "iucn_category": area.get("iucn_category", "Not Available")
        })
    return jsonify(results)
```

Questa funzione accetta un parametro di query `country_name` e restituisce una lista delle aree protette in quel paese. Utilizza un dizionario di mapping per convertire il nome del paese in un codice ISO3 e cerca nel database le aree protette corrispondenti.

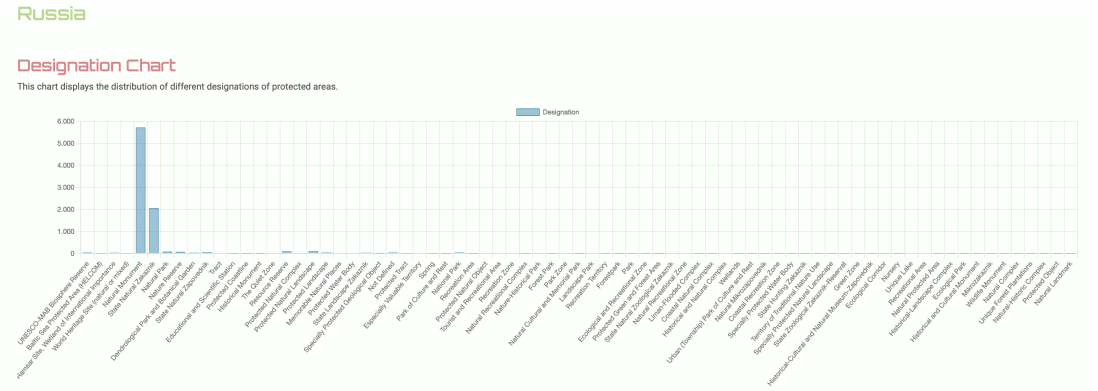

Figura 5.2: Descrizione dell'immagine.

5.12 Uso del Suolo Agricolo

Questa query restituisce i primi 10 paesi con l'uso del suolo agricolo più alto per l'anno 2018.

Listing 5.23: Query per ottenere l'uso del suolo agricolo

```
@app.route('/api/agricultural_land_use', methods=['GET'])
def agricultural_land_use():
    pipeline = [
```

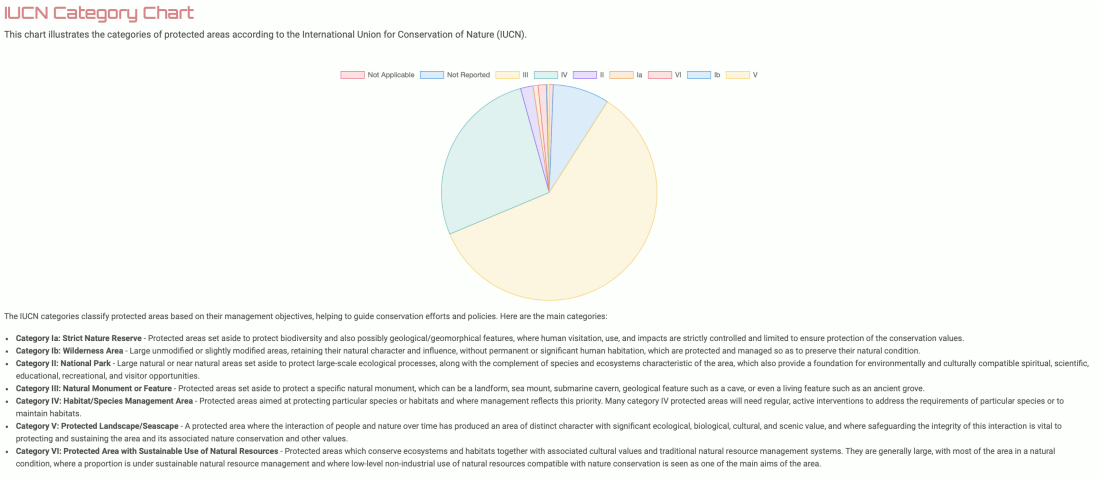

Figura 5.3: Descrizione dell'immagine.

```
        {"$match": {"Year": 2018, "Value": {"$ne": None}, "Country
            Name": {"$ne": None}}},
        {"$group": {"_id": "$Country Name", "Agricultural_Land_Use":
            {"$avg": "$Value"}}},
        {"$sort": {"Agricultural_Land_Use": -1}},
        {"$limit": 10}
    ]
    data = list(mongo.db.agricultural_land.aggregate(pipeline))
    data = convert_and_filter(data)
    return jsonify(data)
```

Questa funzione utilizza una pipeline di aggregazione per filtrare, raggruppare, ordinare e limitare i dati.

5.13 Emissioni di PM2.5

Questa query restituisce le emissioni medie di PM2.5 per paese dal 2000 al 2018.

Listing 5.24: Query per ottenere le emissioni di PM2.5

```
@app.route('/api/pm25_emissions')
def pm25_emissions():
    try:
```

```
        pipeline = [
            {"$match": {"Year": {"$gte": 2000, "$lte": 2018}, "Value":
                {"$ne": None}, "Country Name": {"$ne": None}}},
            {"$group": {"_id": "$Country Name",
                "Average_PM25_Emissions": {"$avg": "$Value"}}}
        ]
        data = list(mongo.db.pm25_air_pollution.aggregate(pipeline))
        data = convert_and_filter(data)
        if not data:
            return jsonify({"error": "No data found for PM2.5
                emissions"}), 404
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching PM2.5 emissions data: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Questa funzione utilizza una pipeline di aggregazione per filtrare i dati per anno, raggrupparli per paese e calcolare la media delle emissioni.

5.14 Specie di Uccelli Minacciate

Questa query restituisce il numero di specie di uccelli minacciate per paese per l'anno 2018.

Listing 5.25: Query per ottenere le specie di uccelli minacciate

```
@app.route('/api/threatened_bird_species')
def threatened_bird_species():
    pipeline = [
        {"$match": {"Year": 2018}},
        {"$group": {"_id": "$Country Name", "Threatened_Bird_Species":
            {"$sum": "$Value"}}}
    ]
    data = list(mongo.db.threatened_bird_species.aggregate(pipeline))
```

```
    data = convert_and_filter(data)
    return jsonify(data)
```

Questa funzione utilizza una pipeline di aggregazione per raggruppare i dati per paese e sommare il numero di specie minacciate.

5.15 Emissioni di CO_2 da Combustibili Solidi

Questa query calcola le emissioni medie di CO_2 derivanti dall'uso di combustibili solidi per ciascun paese tra il 2000 e il 2020.

Listing 5.26: Query per ottenere le emissioni di CO_2 da combustibili solidi

```
@app.route('/api/co2_solid_fuel_emissions', methods=['GET'])
def get_co2_solid_fuel_emissions():
    try:
        pipeline = [
            {"$match": {"Year": {"$gte": 2000, "$lte": 2020}}},
            {"$group": {"_id": "$Country Name",
                "Average_CO2_Solid_Fuel_Emissions": {"$avg": "$Value"}}}
        ]
        data =
            list(mongo.db.co2_solid_fuel_emissions.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching CO2 solid fuel emissions data:
            {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Questa query utilizza una pipeline di aggregazione per filtrare i dati in base agli anni specificati, raggruppare i dati per paese e calcolare la media delle emissioni di CO_2.

5.16 Emissioni nette di GHG

Questa query calcola le emissioni nette medie di gas serra (GHG) per ciascun paese tra il 2000 e il 2020.

Listing 5.27: Query per ottenere le emissioni nette di gas serra

```
@app.route('/api/net_ghg_emissions', methods=['GET'])
def get_net_ghg_emissions():
    try:
        pipeline = [
            {"$match": {"Year": {"$gte": 2000, "$lte": 2020}}},
            {"$group": {"_id": "$Country Name",
                "Average_Net_GHG_Emissions": {"$avg": "$Value"}}}
        ]
        data = list(mongo.db.net_ghg_emissions.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching net GHG emissions data: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Questa query utilizza una pipeline di aggregazione per filtrare i dati in base agli anni specificati, raggruppare i dati per paese e calcolare la media delle emissioni nette di gas serra.

5.17 Specie di Pesci Minacciate

Questa query restituisce il numero di specie di pesci minacciate per ciascun paese per l'anno 2018.

Listing 5.28: Query per ottenere le specie di pesci minacciate

```
@app.route('/api/threatened_fish_species', methods=['GET'])
def get_threatened_fish_species():
    pipeline = [
```

```
        {"$match": {"Year": 2018}},
        {"$group": {"_id": "$Country Name", "Threatened_Fish_Species":
            {"$sum": "$Value"}}}
    ]
    data = list(mongo.db.threatened_fish_species.aggregate(pipeline))
    data = convert_and_filter(data)
    return jsonify(data)
```

Questa query utilizza una pipeline di aggregazione per filtrare i dati in base all'anno specificato, raggruppare i dati per paese e sommare il numero di specie di pesci minacciate per ciascun paese.

5.18 Query Interattive

Questa sezione descrive le query interattive che permettono di estrarre dati dinamici basati su diversi parametri e tipi di query.

La funzione `execute_query` permette di eseguire diverse tipologie di query su differenti indicatori ambientali ed economici. Gli utenti possono specificare l'indicatore e il tipo di query per ottenere risultati dinamici.

5.18.1 Tipologie di Query

- **Top 10:** Questa query restituisce i primi 10 paesi con i valori medi più alti per l'indicatore specificato.

Listing 5.29: Esempio di query "Top 10"

```
pipeline = [
    {"$group": {"_id": "$Country Name", "Value": {"$avg":
        "$Value"}}},
    {"$sort": {"Value": -1}},
    {"$limit": 10}
]
```

- **Bottom 10:** Questa query restituisce i 10 paesi con i valori medi più bassi per l'indicatore specificato.

```
pipeline = [
    {"$group": {"_id": "$Country Name", "Value": {"$avg":
        "$Value"}}},
    {"$sort": {"Value": 1}},
    {"$limit": 10}
]
```

- **Media per Regione:** Questa query calcola la media dei valori per ogni regione o paese (se la regione non è disponibile) per l'indicatore specificato.

```
pipeline = [
{"$group": {
    "_id": {"$ifNull": ["$Region", "$Country Name"]},
    "Value": {"$avg": "$Value"}
}},
{"$sort": {"Value": -1}}
]
```

La funzione accetta due parametri di query, *indicator* e *queryType*, che specificano l'indicatore di interesse e il tipo di query da eseguire. La funzione utilizza una mappa (`indicator_to_collection_map`) per ottenere il nome della collezione MongoDB corrispondente all'indicatore specificato. In base al tipo di query, costruisce una pipeline di aggregazione per eseguire la query desiderata.

5.19 Correlazione tra Uso di Energia Rinnovabile ed Emissioni Totali di Gas Serra

Questa query calcola la correlazione tra l'uso di energia rinnovabile e le emissioni totali di gas serra per ogni paese tra il 2000 e il 2020.

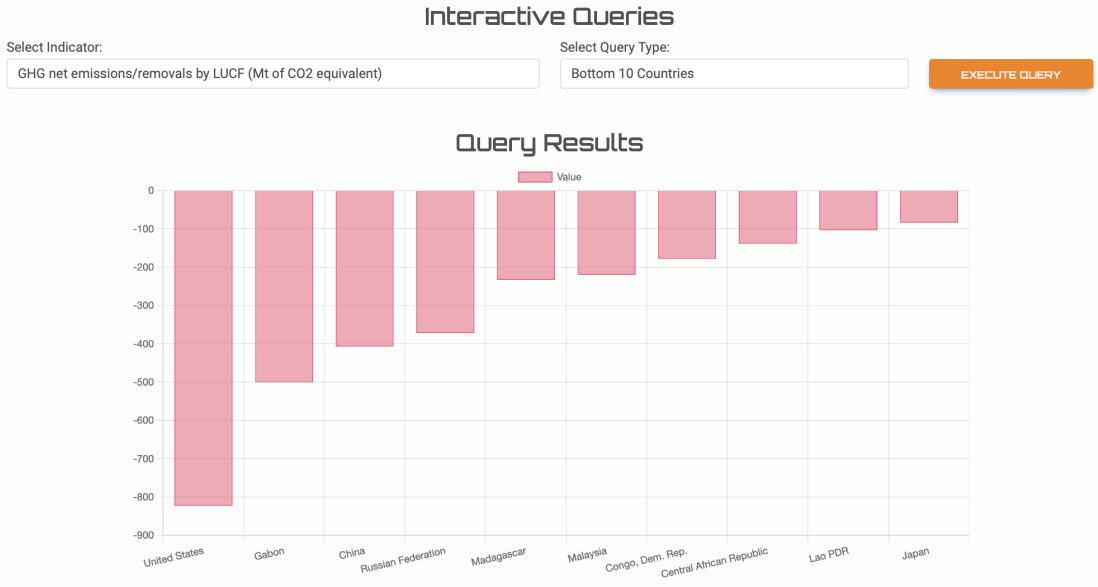

Figura 5.4: Descrizione dell'immagine.

Listing 5.30: Funzione per eseguire query interattive

```
@app.route('/api/correlation_renewable_energy_ghg', methods=['GET'])
def get_correlation_renewable_energy_ghg():
    try:
        pipeline = [
            {"$match": {"Year": {"$gte": 2000, "$lte": 2020}}},
            {"$addFields": {"CountryYear": {"$concat": ["$Country Name",
                "-", {"$toString": "$Year"}]}}},
            {"$lookup": {
                "from": "total_ghg_emissions",
                "let": {"country_year": "$CountryYear"},
                "pipeline": [
                    {"$addFields": {"CountryYear": {"$concat":
                        ["$Country Name", "-", {"$toString": "$Year"}]}}},
                    {"$match": {"$expr": {"$eq": ["$CountryYear",
                        "$$country_year"]}}}
                ],
                "as": "total_ghg_emissions_data"
            }},
```

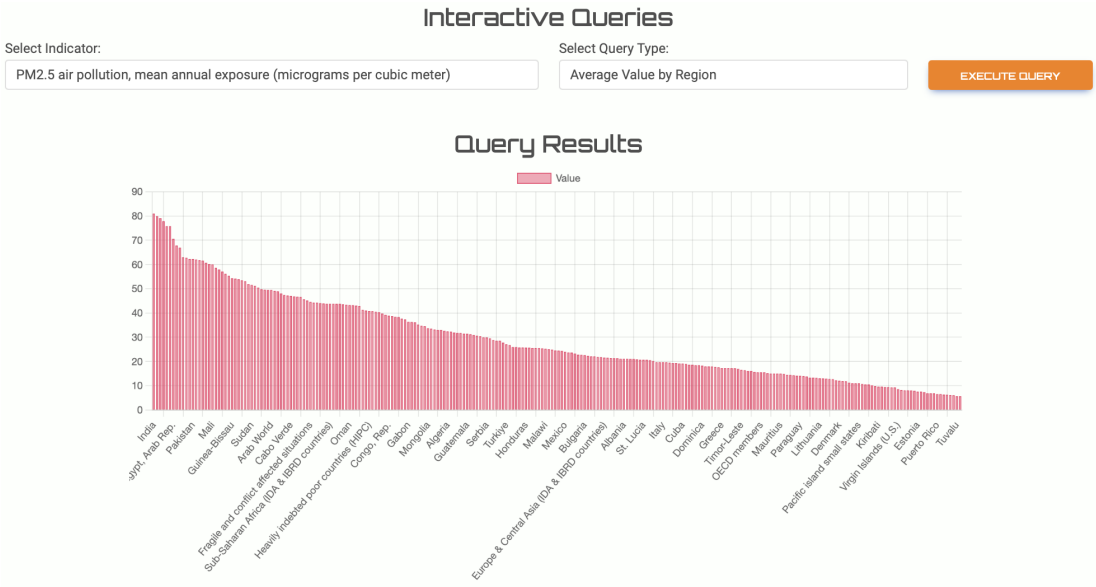

Figura 5.5: Descrizione dell'immagine.

```
        {"$unwind": "$total_ghg_emissions_data"},
        {"$project": {
            "Country": "$Country Name",
            "Year": "$Year",
            "Renewable_Energy_Use": "$Value",
            "Total_GHG_Emissions": "$total_ghg_emissions_data.Value"
        }},
        {"$match": {"Renewable_Energy_Use": {"$ne": None},
            "Total_GHG_Emissions": {"$ne": None}}}
    ]
    data = list(mongo.db.renewable_energy.aggregate(pipeline))

    for document in data:
        document.pop('_id', None)
        document.pop('total_ghg_emissions_data', None)

    if not data:
        return jsonify({
            "error": "No data found for the correlation between
                renewable energy use and total GHG emissions"}), 404
```

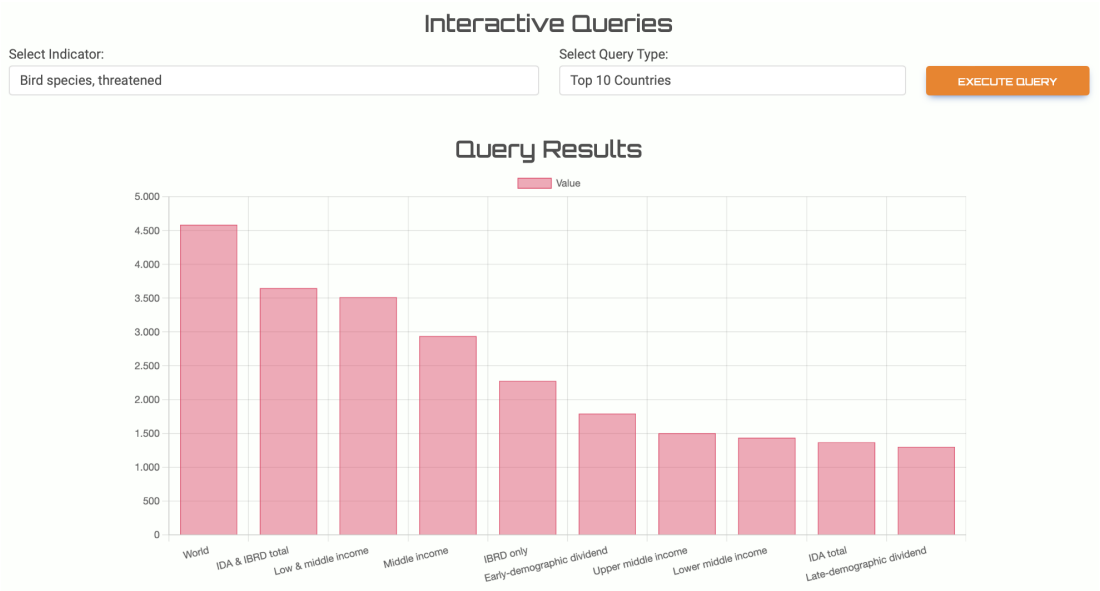

Figura 5.6: Descrizione dell'immagine.

```
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching renewable energy use and total
            GHG emissions correlation data: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Questa funzione utilizza una pipeline di aggregazione per combinare i dati dall'uso di energia rinnovabile e dalle emissioni totali di gas serra, basata su una chiave composta da paese e anno.

5.20 Top 10 Emissioni di Gas Serra dai Prodotti Alimentari

Questa query restituisce i 10 prodotti alimentari con le emissioni totali di gas serra per chilogrammo più alte.

Listing 5.31: Funzione per eseguire query interattive

```
@app.route('/api/top_10_food_ghg_emissions', methods=['GET'])
def get_top_10_food_ghg_emissions():
```

```
    try:
        data = list(mongo.db.food_product_emissions.find({}, {'_id': 0,
            'food_product': 1,
        'total_ghg_emissions_per_kg': 1}).sort(
        'total_ghg_emissions_per_kg', -1).limit(10))
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching top 10 food GHG emissions: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Questa funzione utilizza una ricerca e ordinamento sui dati delle emissioni di prodotti alimentari per identificare i prodotti con il maggiore impatto ambientale.

5.21 Top 10 Uso del Suolo Agricolo

La query seguente è progettata per recuperare i dati dai primi 10 paesi con il più alto uso di suolo agricolo nel 2020. La pipeline di aggregazione su MongoDB filtra, ordina e seleziona i campi rilevanti per fornire i risultati desiderati.

Listing 5.32: Funzione per ottenere i dati sull'uso del suolo agricolo

```
@app.route('/api/top_10_agricultural_land_use', methods=['GET'])
    def get_top_10_agricultural_land_use():
        try:
            pipeline = [
            {"$match": {"Year": 2020}},
            {"$sort": {"Value": -1}},
            {"$limit": 10},
            {"$project": {"Country": "$Country Name",
                "Agricultural_Land_Use": "$Value"}}
            ]
        data = list(mongo.db.agricultural_land.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
```

```
    except Exception as e:
        logging.error(f"Error fetching top 10 agricultural land use:
            {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Utilizza il filtro $match per selezionare i dati relativi all'anno 2020, $sort per ordinare i paesi in base al valore dell'uso del suolo agricolo in ordine decrescente (Value: -1), $limit per restringere i risultati ai primi 10 paesi con l'uso del suolo agricolo più alto e $project per selezionare e rinominare i campi Country e Agricultural_Land_Use dai risultati aggregati.

5.22 Top 10 Emissioni di PM2.5

La query seguente è progettata per recuperare i dati sui primi 10 paesi con le emissioni medie di PM2.5 più alte. Utilizza una pipeline di aggregazione su MongoDB per raggruppare, calcolare la media, ordinare e limitare i risultati ai primi 10 paesi.

Listing 5.33: Funzione per ottenere i dati sulle emissioni di PM2.5

```
@app.route('/api/top_10_pm25_emissions', methods=['GET'])
def get_top_10_pm25_emissions():
    try:
        pipeline = [
            {"$group": {"_id": "$Country Name", "avg_pm25_emissions":
                {"$avg": "$Value"}}},
            {"$sort": {"avg_pm25_emissions": -1}},
            {"$limit": 10}
        ]
        data = list(mongo.db.pm25_air_pollution.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching top 10 PM2.5 emissions: {e}")
```

```
        return jsonify({"error": "Internal server error"}), 500
```

Utilizza il raggruppamento $group per aggregare i dati in base al nome del paese (Country Name). Durante questo processo, calcola la media delle emissioni di PM2.5 (avg_pm25_emissions) per ciascun paese. Attraverso $sort ordina i paesi in base al valore medio delle emissioni di PM2.5 in ordine decrescente (avg_pm25_emissions: -1), mentre utilizza $limit per restringere i risultati ai primi 10 paesi con le emissioni medie di PM2.5 più alte.

5.23 Top 10 Indice di Rischio Climatico

La query seguente è progettata per recuperare i dati sui primi 10 paesi con l'indice di rischio climatico medio più alto. Utilizza una pipeline di aggregazione su MongoDB per raggruppare, calcolare la media, ordinare e limitare i risultati ai primi 10 paesi.

Listing 5.34: Funzione per ottenere i dati sull'indice di rischio climatico

```
@app.route('/api/top_10_climate_risk_index', methods=['GET'])
def get_top_10_climate_risk_index():
    try:
        pipeline = [
            {"$group": {"_id": "$Country Name",
                "avg_climate_risk_index": {"$avg": "$Value"}}},
            {"$sort": {"avg_climate_risk_index": -1}},
            {"$limit": 10}
        ]
        data = list(mongo.db.climate_risk_index.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching top 10 climate risk index: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Utilizza il raggruppamento $group per aggregare i dati in base al nome del paese (Country Name). Durante questo processo, calcola la media dell'indice di rischio climatico (avg_climate_risk_index) per ciascun paese, inoltre utilizza $sort per ordinare i paesi in base al valore medio dell'indice di rischio climatico in ordine decrescente ('avg_climate_risk_index: -1') e $limit per restringere i risultati ai primi 10 paesi con l'indice di rischio climatico medio più alto.

Il risultato di questa query sarà una lista JSON contenente i primi 10 paesi con l'indice di rischio climatico medio più alto. Ogni oggetto JSON nel risultato conterrà il nome del paese e il valore medio dell'indice di rischio climatico.

Questa funzione fornisce una panoramica rapida dei paesi più vulnerabili ai cambiamenti climatici, identificando le aree con il più alto rischio climatico. L'indice di rischio climatico considera diversi fattori, tra cui la frequenza e la gravità dei disastri naturali, l'esposizione e la vulnerabilità della popolazione, e la capacità di risposta e adattamento ai cambiamenti climatici.

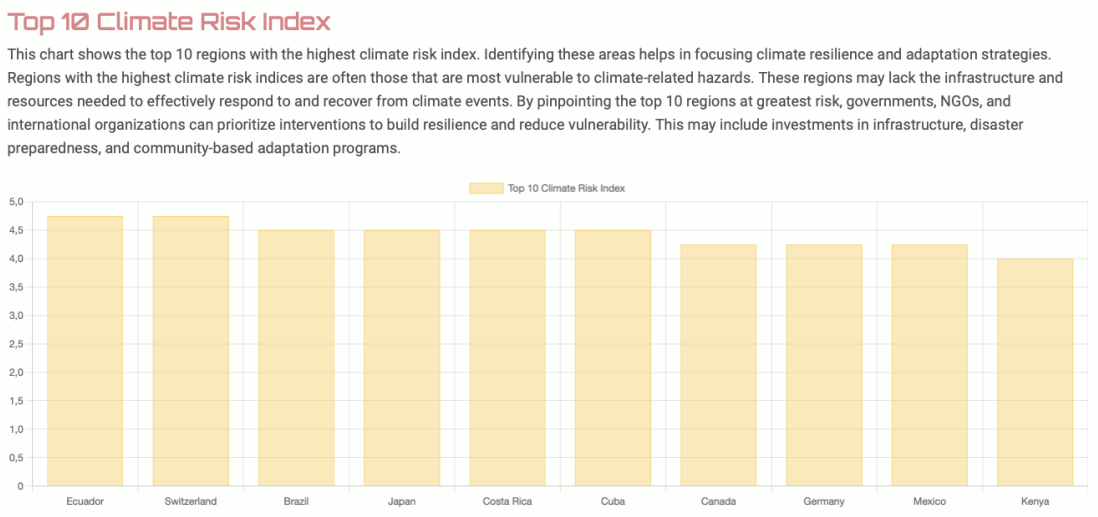

Figura 5.7: Descrizione dell'immagine.

5.24 Aumento Percentuale Maggiore delle Emissioni di CO2

La query seguente è progettata per identificare i paesi con il più grande aumento percentuale delle emissioni di CO2 tra gli anni 2000 e 2020. Utilizza una pipeline di aggregazione su MongoDB per filtrare, raggruppare, calcolare le emissioni e determinare la variazione percentuale.

Listing 5.35: Funzione per ottenere i dati sull'aumento percentuale maggiore delle emissioni di CO2

```
@app.route('/api/countries_largest_percentage_increase_co2_emissions',
    methods=['GET'])
def get_countries_largest_percentage_increase_co2_emissions():
    try:
        pipeline = [
            {"$match": {"Year": {"$in": [2000, 2020]}}},
            {"$group": {"_id": {"Country": "$Country Name", "Year":
                "$Year"}, "CO2_Emissions": {"$sum": "$Value"}}},
            {"$group": {"_id": "$_id.Country",
                        "Emissions": {"$push": {"year": "$_id.Year",
                            "value": "$CO2_Emissions"}}}},
            {"$project": {
                "Country": "$_id",
                "Percentage_Change": {
                    "$cond": {
                        "if": {"$eq": [{"$arrayElemAt":
                            ["$Emissions.value", 0]}, 0]},
                        "then": None,
                        "else": {
                            "$multiply": [
                                {"$divide": [{"$subtract":
                                    [{"$arrayElemAt": ["$Emissions.value",
                                    1]},
```

```
                                                    {"$arrayElemAt":
                                                        ["$Emissions.value",
                                                        0]}]},
                                            {"$arrayElemAt":
                                                ["$Emissions.value", 0]}]},
                                100
                            ]
                        }
                    }
                }
            }}},
            {"$sort": {"Percentage_Change": -1}},
            {"$limit": 10}
        ]
        data = list(mongo.db.co2_emissions.aggregate(pipeline))
        data = convert_and_filter(data)
        return jsonify(data)
    except Exception as e:
        logging.error(f"Error fetching countries with largest
            percentage increase in CO2 emissions: {e}")
        return jsonify({"error": "Internal server error"}), 500
```

Utilizza $match per filtrare i dati solo per gli anni 2000 e 2020, $group per raggruppare i dati in base al nome del paese e all'anno, calcolando la somma delle emissioni di CO2 (CO2_Emissions) per ciascun paese e anno e un secondo $group per aggregare i dati a livello di paese, creando un array di emissioni per ciascun paese con i dati degli anni 2000 e 2020. Inoltre utilizza $project per calcolare la variazione percentuale delle emissioni di CO2 tra il 2000 e il 2020. Se le emissioni del 2000 sono zero, la variazione percentuale viene impostata su *None*. $sort viene utilizzato per ordinare i paesi in base alla variazione percentuale in ordine decrescente (Percentage_Change: -1), mentre $limit per restringere i risultati ai primi 10 paesi con il maggiore aumento percentuale delle emissioni di CO_2.

Il risultato di questa query sarà una lista JSON contenente i paesi con il più grande aumento percentuale delle emissioni di CO_2. Ogni oggetto JSON nel risultato conterrà il nome del paese e la variazione percentuale delle emissioni di CO_2.

Questa funzione fornisce una panoramica dei paesi che hanno avuto il maggiore aumento percentuale delle emissioni di CO_2, evidenziando i cambiamenti significativi nelle emissioni a livello globale.

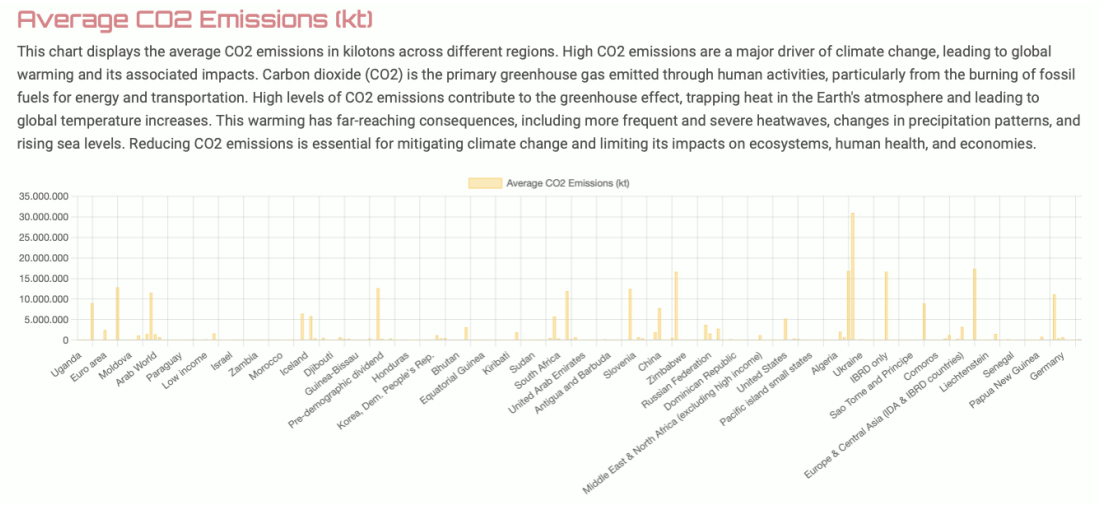

Figura 5.8: Descrizione dell'immagine.

5.25 Variazione delle Emissioni di CO_2

La query seguente è progettata per determinare la variazione percentuale delle emissioni di CO_2 per ciascun paese tra gli anni 2000 e 2020. Utilizza una pipeline di aggregazione su MongoDB per filtrare, raggruppare e calcolare le emissioni e la loro variazione percentuale.

Listing 5.36: Funzione per ottenere la variazione percentuale delle emissioni di CO2

```
@app.route('/api/co2_emissions_change')
def co2_emissions_change():
    pipeline = [
        {"$match": {"Year": {"$in": [2000, 2020]}}},
```

```
        {"$group": {"_id": {"Country": "$Country Name", "Year":
            "$Year"}, "CO2_Emissions": {"$sum": "$Value"}}},
        {"$group": {"_id": "$_id.Country", "Emissions": {"$push":
            {"year": "$_id.Year", "value": "$CO2_Emissions"}}}},
        {"$project": {
            "Country": "$_id",
            "Percentage_Change": {
                "$cond": {
                    "if": {"$eq": [{"$arrayElemAt": ["$Emissions.value",
                        0]}, 0]},
                    "then": None,
                    "else": {
                        "$multiply": [
                            {"$divide": [{"$subtract": [{"$arrayElemAt":
                                ["$Emissions.value", 1]},
                                                        {"$arrayElemAt":
                                                            ["$Emissions.value",
                                                            0]}]},
                                        {"$arrayElemAt":
                                            ["$Emissions.value", 0]}]},
                            100
                        ]
                    }
                }
            }
        }}
    ]
    data = list(mongo.db.co2_emissions.aggregate(pipeline))
    data = convert_and_filter(data)
    return jsonify(data)
```

Utilizza $match per filtrare i dati solo per gli anni 2000 e 2020, $group per raggruppare i dati in base al nome del paese e all'anno, calcolando la somma

delle emissioni di CO_2 (CO2_Emissions) per ciascun paese e anno e un secondo $group per aggregare i dati a livello di paese, creando un array di emissioni per ciascun paese con i dati degli anni 2000 e 2020. Inoltre utilizza $project per calcolare la variazione percentuale delle emissioni di CO_2 tra il 2000 e il 2020. Se le emissioni del 2000 sono zero, la variazione percentuale viene impostata su *None*. La pipeline viene eseguita sulla collezione co2_emissions e i risultati vengono convertiti e filtrati.

Il risultato di questa query sarà una lista JSON contenente i paesi con la variazione percentuale delle emissioni di CO_2. Ogni oggetto JSON nel risultato conterrà il nome del paese e la variazione percentuale delle emissioni di CO_2.

Questa funzione fornisce una panoramica delle variazioni percentuali delle emissioni di CO_2 per i paesi selezionati, permettendo di identificare le nazioni con i cambiamenti più significativi nelle emissioni di CO_2 tra il 2000 e il 2020.

Capitolo 6

Tecnologie Utilizzate

Questo capitolo descrive le tecnologie e gli strumenti utilizzati nel progetto *SOS Earth*, spiegando i motivi della loro scelta e i vantaggi che offrono.

6.1 Data Warehouse

Il data warehouse, costruito utilizzando MongoDB, è stato scelto come backend del progetto per diverse ragioni fondamentali. Un data warehouse è un sistema utilizzato per la raccolta, la gestione e l'analisi di grandi quantità di dati provenienti da diverse fonti. I principali vantaggi dell'utilizzo di un data warehouse in questo progetto includono:

- **Integrazione dei Dati**: Consente di integrare dati provenienti da diverse fonti, fornendo una visione unificata e coerente delle informazioni. Questo è particolarmente importante per un progetto che analizza vari indicatori ambientali provenienti da diverse origini.

- **Efficienza di Query**: I data warehouse sono ottimizzati per eseguire query complesse in modo rapido ed efficiente. Questo è cruciale per fornire risposte immediate alle richieste del frontend, migliorando l'esperienza utente.

- **Storico dei Dati**: Permettono di mantenere uno storico dei dati, rendendo possibile l'analisi delle tendenze nel tempo. Questo è utile per la sezione

delle serie storiche del progetto, dove gli utenti possono esplorare i dati ambientali nel corso degli anni.

- **Scalabilità**: Possono gestire grandi volumi di dati e supportare un numero elevato di utenti simultanei, rendendo il sistema scalabile per future espansioni del progetto.
- **Qualità dei Dati**: Il processo ETL (Extract, Transform, Load) integrato nel data warehouse garantisce che i dati siano puliti, trasformati e caricati in un formato coerente e di alta qualità, migliorando l'accuratezza delle analisi.

6.2 Linguaggi e Librerie

Una varietà di linguaggi di programmazione e librerie sono stati utilizzati per sviluppare *SOS Earth*, ciascuno scelto per le sue specifiche capacità e vantaggi. Di seguito vengono descritti i principali linguaggi e librerie utilizzati nel progetto.

6.2.1 Python per il Processo ETL

Python è stato scelto per il processo ETL grazie alla sua semplicità, flessibilità e vasta gamma di librerie per la manipolazione dei dati. Le principali librerie utilizzate includono:

- **Pandas**: Utilizzata per la manipolazione e l'analisi dei dati, offre potenti strumenti per l'elaborazione dei dati, inclusi filtri, aggregazioni e trasformazioni, rendendo il processo ETL efficiente e intuitivo.
- **NumPy**: Utilizzata per operazioni numeriche ad alte prestazioni, è essenziale per il calcolo efficiente dei dati, offrendo supporto per array multidimensionali e una vasta gamma di funzioni matematiche.

6.2.2 Flask per le API REST

Flask è stato utilizzato per creare le API REST. Flask è un micro framework per Python che facilita lo sviluppo di applicazioni web e API. Le principali motivazioni per la scelta di Flask includono:

- **Semplicità e Flessibilità**: Flask è leggero e facile da configurare, permettendo di costruire applicazioni web e API senza una struttura rigida. Questo rende lo sviluppo rapido e flessibile.
- **Modularità**: La natura modulare di Flask consente di estendere le funzionalità dell'applicazione utilizzando diverse estensioni, che possono essere aggiunte o rimosse secondo le necessità del progetto.
- **Integrazione con altre Librerie**: Flask si integra facilmente con altre librerie Python utilizzate nel progetto, come Pandas e NumPy, e con strumenti di front-end come JavaScript e Bootstrap, permettendo di creare un'architettura coesa.

6.2.3 JavaScript per l'Interattività

JavaScript è stato scelto per aggiungere interattività alle pagine web, essendo il linguaggio principale per lo sviluppo frontend. Offre numerose librerie e framework che permettono di creare interfacce utente dinamiche e responsive.

- **Chart.js**: Utilizzato per creare grafici dinamici e interattivi è una libreria semplice da usare ma potente, che supporta vari tipi di grafici, inclusi grafici a barre, linee, torte e scatter plot. È ideale per visualizzare dati in modo chiaro e accattivante.
- **Bootstrap**: Utilizzato per lo sviluppo di design reattivi e moderni, offre una vasta gamma di componenti predefiniti e un sistema di grid layout che facilita la creazione di pagine web responsive, garantendo un'esperienza utente ottimale su dispositivi diversi.

- **amCharts5.js (am5.j)**: Una libreria di grafici ricca di funzionalità che supporta vari tipi di grafici e mappe, con un'ampia gamma di opzioni di personalizzazione. È particolarmente utile per creare dashboard complesse e visualizzazioni di dati dettagliate.

6.3 Motivazioni delle Scelte Tecnologiche

Le tecnologie e gli strumenti sopra descritti sono stati scelti per le loro capacità di supportare gli obiettivi del progetto in modo efficiente e scalabile. La combinazione di un data warehouse robusto, potenti strumenti di elaborazione dei dati in Python, e librerie di visualizzazione interattive in JavaScript, offre una piattaforma solida per l'analisi e la visualizzazione dei dati ambientali.

La scelta di Flask per costruire le API REST si è rivelata vantaggiosa grazie alla sua semplicità, flessibilità e capacità di integrarsi facilmente con altre tecnologie utilizzate nel progetto. Queste caratteristiche hanno permesso di sviluppare un backend robusto e scalabile, capace di gestire richieste complesse e fornire dati in modo efficiente al frontend interattivo.

Capitolo 7

Conclusioni e Lavori Futuri

In questo capitolo vengono presentate le conclusioni del progetto *SOS Earth* e vengono discusse le possibilità di miglioramento e le direzioni future per l'espansione del progetto.

7.1 Conclusioni

Il progetto *SOS Earth* ha dimostrato l'importanza di avere un sistema integrato per il monitoraggio e l'analisi dei dati ambientali. Attraverso l'uso di un data warehouse robusto e tecnologie avanzate per l'elaborazione e la visualizzazione dei dati, il progetto ha raggiunto diversi obiettivi chiave:

- **Centralizzazione dei Dati**: Il progetto ha centralizzato dati provenienti da diverse fonti, offrendo una visione unificata e coerente delle informazioni ambientali.
- **Accessibilità dei Dati**: Grazie all'uso di API REST e di strumenti di visualizzazione interattiva, gli utenti possono accedere facilmente ai dati e comprenderne l'importanza.
- **Analisi delle Tendenze**: Il sistema ha permesso di analizzare le tendenze storiche degli indicatori ambientali, fornendo una base solida per la pianificazione e l'implementazione di politiche ambientali efficaci.

- **Sensibilizzazione e Educazione**: *SOS Earth* ha aumentato la consapevolezza del pubblico riguardo ai problemi ambientali e ha offerto strumenti educativi preziosi.

Questi risultati evidenziano il successo del progetto e il suo potenziale impatto positivo sull'ambiente e sulla società.

7.2 Possibili Miglioramenti

Nonostante i risultati positivi, ci sono sempre margini di miglioramento. Ecco alcuni suggerimenti per rendere il progetto ancora più efficace:

- **Miglioramento della Qualità dei Dati**: Implementare tecniche avanzate di pulizia e normalizzazione dei dati per garantire una qualità ancora maggiore delle informazioni raccolte.
- **Integrazione di Nuovi Dataset**: Espandere il data warehouse includendo nuovi dataset rilevanti, come dati meteorologici in tempo reale o dati sull'inquinamento idrico.
- **Ottimizzazione delle Prestazioni**: Migliorare l'efficienza delle query e delle operazioni di elaborazione dati per ridurre ulteriormente i tempi di risposta del sistema.
- **Espansione delle Funzionalità di Visualizzazione**: Integrare nuove tipologie di grafici e visualizzazioni, come mappe termiche e grafici 3D, per offrire una rappresentazione più dettagliata e intuitiva dei dati.
- **Miglioramento dell'Interfaccia Utente**: Continuare a perfezionare l'interfaccia utente per renderla ancora più intuitiva e user-friendly, basandosi sui feedback degli utenti.

7.3 Sviluppi Futuri

Per garantire la crescita e l'efficacia continuativa del progetto, si potrebbero considerare i seguenti sviluppi futuri:

- **Previsioni e Modelli Predittivi**: Sviluppare algoritmi di machine learning per prevedere le tendenze future degli indicatori ambientali e identificare potenziali rischi.
- **Espansione Geografica**: Estendere il progetto per includere dati ambientali di nuove regioni geografiche, ampliando così la portata e l'impatto del progetto a livello globale.
- **Collaborazione con Organizzazioni Ambientali**: Stabilire collaborazioni con organizzazioni ambientali internazionali per condividere dati e risorse, e per promuovere iniziative congiunte.
- **Applicazioni Mobili**: Sviluppare applicazioni mobili per rendere i dati e le visualizzazioni accessibili anche su dispositivi mobili, migliorando l'accessibilità e l'usabilità del sistema.
- **Gamification e Coinvolgimento degli Utenti**: Implementare elementi di gamification per incentivare l'interazione degli utenti e promuovere comportamenti sostenibili attraverso sfide e premi.