Università degli studi di Bari Aldo Moro

DIPARTIMENTO DI INFORMATICA
Corso di Laurea Triennale in Informatica

CASO DI STUDIO DEL CORSO IN INGEGNERIA DELLA CONOSCENZA

"Sistema di Monitoraggio Ambientale e Controllo della Qualità dell'Aria basato sull'Inteligenza artificiali"

Autori :

Zagharitafreshi Marjan 704820 m.zagharitafreshi@studenti.uniba.it

Robles Vincenza 728324 v.robles1@studenti.uniba.it

Anno Accademico 2023-2024

Indice

Obiettivo:

Implementare un sistema intelligente di monitoraggio ambientale per valutare la qualità dell'aria e gli inquinanti atmosferici in un'area specifica come città di Bari, fornendo avvisi tempestivi alle autorità competenti e al pubblico.

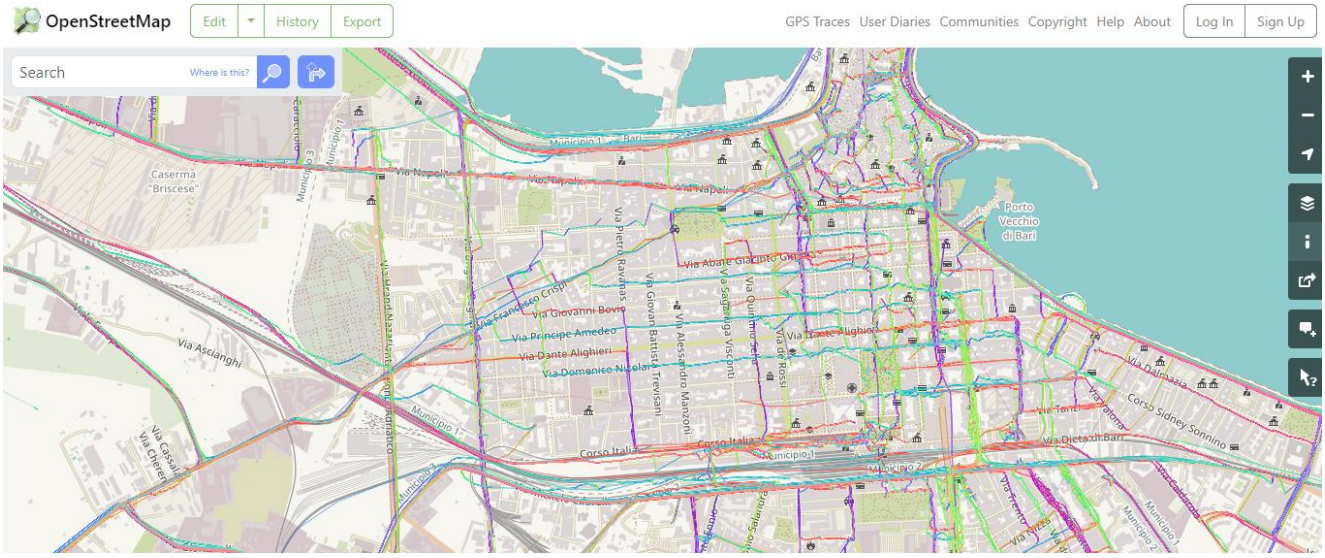
Introduzione:

Negli ultimi decenni, la crescente industrializzazione e l'urbanizzazione hanno portato a un aumento significativo dell'inquinamento atmosferico in molte aree del mondo. L'esposizione a inquinanti come PM2.5, PM10 , NO2 , CO ha gravi implicazioni sulla salute umana e sull'ambiente. Pertanto, è diventato sempre più cruciale implementare sistemi di monitoraggio ambientale avanzati per valutare la qualità dell'aria in modo accurato e fornire avvisi tempestivi per proteggere la salute pubblica. In questo contesto, questo progetto propone lo sviluppo di un Sistema di Monitoraggio Ambientale e Controllo della Qualità dell'Aria basato sull'Ingegneria della Conoscenza, che sfrutta dati in tempo reale da sensori ambientali, modelli atmosferici e analisi storiche per valutare e gestire l'inquinamento atmosferico in un'area specifica.

Ontologie

OpenStreetMap (OSM) è un progetto open source che permette agli utenti di creare e condividere dati geografici, come informazioni su strade, edifici e punti di interesse. I dati di OSM sono disponibili in vari formati, incluso XML, un formato di testo ampiamente utilizzato per la trasmissione e rappresentazione dei dati su Internet.

Nel contesto di un Sistema di Monitoraggio Ambientale e Controllo della Qualità dell'Aria, i dati di OSM in formato XML possono essere particolarmente utili. Questi dati possono essere convertiti in Prolog, un linguaggio di programmazione logica utilizzato per rappresentare conoscenza e interrogare basi di dati. La conversione dei dati XML di OSM in un formato compatibile con Prolog consente di creare una serie di clausole e classi che descrivono le relazioni tra gli elementi della mappa, facilitando così l'integrazione con sistemi di monitoraggio ambientale.



Un file XML di OpenStreetMap si compone di diversi elementi che descrivono gli oggetti sulla mappa, tra cui:

- <node>: rappresenta un punto geografico sulla mappa, definito da attributi come latitudine e longitudine.

Utilizzo Sensori Ambientali: Ogni sensore di qualità dell'aria può essere rappresentato come un <node>, con latitudine e longitudine che indicano la sua posizione esatta.

Punti di Riferimento: Puoi mappare punti chiave come parchi, scuole o industrie, che possono influenzare i livelli di qualità dell'aria.

- <way>: è una sequenza di nodi che rappresenta strade, sentieri, fiumi o altre linee.

Utilizzo Percorsi di Trasporto: Mappare strade o percorsi di trasporto può aiutare a comprendere le fonti di inquinamento legate al traffico.

Zone di Monitoraggio: Le strade possono essere utilizzate per delineare aree specifiche di monitoraggio e analizzare l'impatto delle emissioni stradali sulla qualità dell'aria.

- <relation>: definisce una relazione tra vari elementi (nodi, vie, altre relazioni).

Utilizzo Zone di Influenza: Può essere utilizzato per definire relazioni tra sensori e zone specifiche, come la relazione tra una strada trafficata e la qualità dell'aria nelle aree circostanti.

Reti di Monitoraggio: Può rappresentare reti complesse di sensori o la correlazione tra diverse fonti di dati (es. sensori e stazioni meteorologiche).

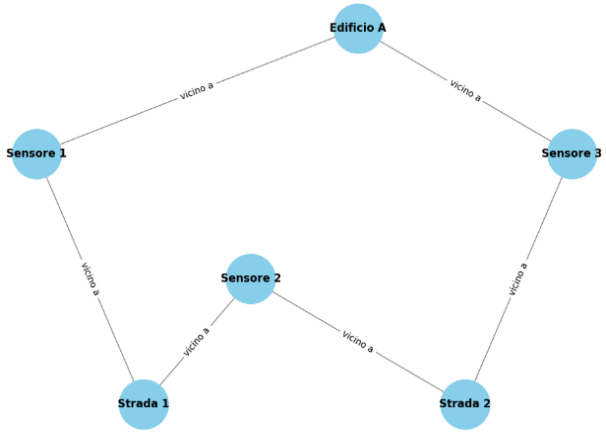
- <tag>: contiene metadati, come il nome di una strada, il tipo di edificio o altro.

Utilizzo Attributi dei Sensori: Puoi usare i <tag> per aggiungere informazioni come il tipo di sensore, il tipo di inquinante misurato (es. PM2.5, NO2), la data di installazione, ecc.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.9.3 (4148004 spike-08.openstreetmap.org)" copyright="OpenStreetMap and contributors" attr
 <bounds minlat="41.1148000" minlon="16.8401000" maxlat="41.1264000" maxlon="16.8663000"/>
 <node id="68528585" visible="true" version="47" changeset="152112135" timestamp="2024-06-01T09:23:04Z" user="Buraddo" uid="2287
  <tag k="capital" v="4"/>
  <tag k="gfoss_id" v="6166"/>
  <tag k="name" v="Bari"/>
  <tag k="name:ar" v="باري"/>
  <tag k="name:be" v="Бары"/>
  <tag k="name:el" v="Μπάρι"/>
  <tag k="name:en" v="Bari"/>
  <tag k="name:fa" v="باری"/>
  <tag k="name:ja" v="バーリ"/>
  <tag k="name:ks" v="باری"/>
  <tag k="name:mk" v="Бари"/>
  <tag k="name:ro" v="Bari"/>
  <tag k="name:ru" v="Бари"/>
  <tag k="name:sr" v="Бари"/>
  <tag k="name:uk" v="Барі"/>
  <tag k="name:ur" v="باری"/>
  <tag k="name:zh" v="巴里"/>
  <tag k="name:zh-Hant" v="巴里"/>
  <tag k="place" v="city"/>
  <tag k="population" v="326344"/>
  <tag k="rank" v="20"/>
  <tag k="source" v="geodati.gfoss.it"/>
  <tag k="wikidata" v="Q3519"/>
 </node>
 <node id="139164298" visible="true" version="8" changeset="39173761" timestamp="2016-05-08T01:38:09Z" user="mcheckimport" uid="
 <node id="139164301" visible="true" version="7" changeset="32978720" timestamp="2015-07-30T12:29:50Z" user="gfigliuolo97" uid="
 <node id="139164303" visible="true" version="6" changeset="8253723" timestamp="2011-05-26T14:45:42Z" user="_mrp_" uid="218322"
 <node id="139164304" visible="true" version="5" changeset="7009795" timestamp="2011-01-18T12:20:00Z" user="_mrp_" uid="218322"
```

Classificazione delle Aree: I tag possono aiutare a classificare aree in base all'uso del suolo, come residenziale, industriale o commerciale, per analizzare l'impatto sull'ambiente.

Questi dati sono organizzati in una struttura a grafo, con ogni elemento identificato da un ID univoco e connesso ad altri elementi tramite relazioni. Questo schema facilita l'accesso e l'estrazione di informazioni utili per il monitoraggio ambientale e il controllo della qualità dell'aria. Ad esempio, è possibile utilizzare i dati di OSM per localizzare sensori di qualità dell'aria rispetto a strade e edifici, ottimizzando la raccolta e l'analisi dei dati ambientali.



I nodi blu rappresentano sensori di qualità dell'aria e altri elementi come strade e edifici.
Gli archi (le linee che collegano i nodi) rappresentano le relazioni di vicinanza tra i diversi elementi.
Le etichette sugli archi indicano la relazione "vicino a" tra un sensore e altri elementi della mappa, come strade o edifici.

Knowledge Base (KB)

Una Knowledge Base (KB)è un insieme di conoscenze organizzate in modo da poter essere utilizzate da un programma o sistema per rispondere a domande o prendere decisioni. una KB memorizza informazioni sui vari inquinanti, le misurazioni rilevate dai sensori, le condizioni ambientali e le regole per l'analisi e il controllo della qualità dell'aria, è importante definire chiaramente le classi e le loro sottoclassi, insieme alla descrizione di ciascuna di esse. Di seguito trovi la descrizione dettagliata delle classi e delle sottoclassi usate nella KB.

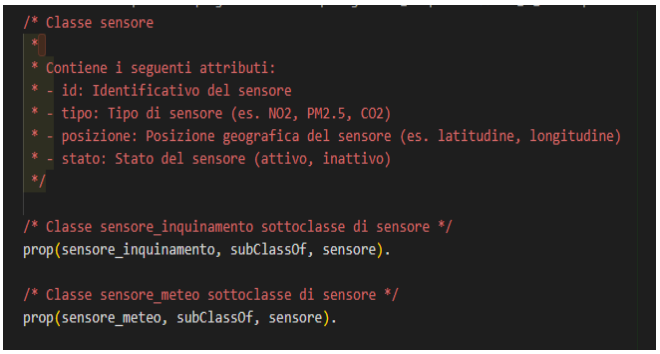
Fatti e regole della KB

Per strutturare una base di conoscenza (KB) in Prolog per un "Sistema di Monitoraggio Ambientale e Controllo della Qualità dell'Aria basato sull'Intelligenza Artificiale", è utile organizzare le informazioni in classi e sottoclassi. Questo approccio aiuta a modellare il dominio in modo più organizzato e semantico, facilitando la creazione di fatti e regole.

I fatti rappresentano dati concreti e informazioni specifiche che il sistema conosce. Sono affermazioni che descrivono lo stato del sistema, come le misurazioni dei sensori, le soglie di inquinanti e le condizioni ambientali in determinati luoghi e tempi.

- **Classe Sensore** : Rappresenta un dispositivo che raccoglie dati ambientali, come i livelli di inquinamento atmosferico,Attributi:

id_sensore: Identificativo univoco del sensore.
tipo: Tipo di sensore (es. PM10, NO2, CO2).
posizione: Coordinate geografiche del sensore (latitudine e longitudine).
stato: Stato operativo del sensore (attivo, inattivo).
Sottoclasse: Sensore Inquinamento,Attributi:
soglia_inquinante: Valore soglia per l'inquinante monitorato.
frequenza_campionamento: Frequenza con cui vengono raccolti i dati.



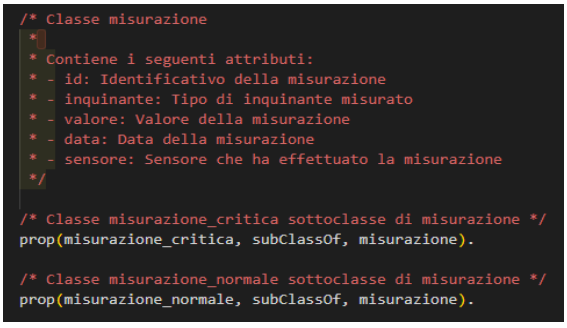
```
/* Classe sensore
 *
 * Contiene i seguenti attributi:
 * - id: Identificativo del sensore
 * - tipo: Tipo di sensore (es. NO2, PM2.5, CO2)
 * - posizione: Posizione geografica del sensore (es. latitudine, longitudine)
 * - stato: Stato del sensore (attivo, inattivo)
 */

/* Classe sensore_inquinamento sottoclasse di sensore */
prop(sensore_inquinamento, subClassOf, sensore).

/* Classe sensore_meteo sottoclasse di sensore */
prop(sensore_meteo, subClassOf, sensore).
```

- **Classe Misurazione**: Rappresenta una registrazione dei dati raccolti da un sensore in un dato.,Attributi:

id_misurazione: Identificativo univoco della misurazione.
inquinante: Tipo di inquinante misurato.
valore: Valore misurato dell'inquinante.
data: Data della misurazione.
sensore: Riferimento al sensore che ha effettuato la misurazione.
Sottoclasse: Misurazione Critica,Attributi:
livello_soglia_superato: Booleano che indica se il valore misurato supera la soglia di sicurezza.



```
/* Classe misurazione
 *
 * Contiene i seguenti attributi:
 * - id: Identificativo della misurazione
 * - inquinante: Tipo di inquinante misurato
 * - valore: Valore della misurazione
 * - data: Data della misurazione
 * - sensore: Sensore che ha effettuato la misurazione
 */

/* Classe misurazione_critica sottoclasse di misurazione */
prop(misurazione_critica, subClassOf, misurazione).

/* Classe misurazione_normale sottoclasse di misurazione */
prop(misurazione_normale, subClassOf, misurazione).
```

- **Classe Condizione Ambientale**: Rappresenta le condizioni ambientali rilevate in un determinato luogo e tempo,Attributi:

```
/* Classe condizione_ambientale
 *
 * Contiene i seguenti attributi:
 * - temperatura: Temperatura dell'aria (in gradi Celsius)
 * - umidita: Percentuale di umidità relativa
 * - velocita_vento: Velocità del vento (in km/h)
 * - direzione_vento: Direzione del vento
 * - pressione: Pressione atmosferica (in hPa)
 * - data: Data della rilevazione
 * - luogo: Luogo della rilevazione
 */

/* Classe condizione_critica sottoclasse di condizione_ambientale */
prop(condizione_critica, subClassOf, condizione_ambientale).

/* Classe condizione_normale sottoclasse di condizione_ambientale */
prop(condizione_normale, subClassOf, condizione_ambientale).
```

temperatura: Temperatura rilevata.
umidita: Percentuale di umidità.
velocita_vento: Velocità del vento.
direzione_vento: Direzione da cui proviene il vento.
pressione: Pressione atmosferica.
luogo: Nome del luogo in cui sono state rilevate le condizioni.
data: Data della registrazione.
Sottoclasse: Condizione Critica,Attributi:
allerta_vento_forte: Booleano che indica se è stato rilevato vento forte.
allerta_inquinamento: Booleano che segnala condizioni che possono peggiorare la qualità dell'aria.

Classe allerta: rappresenta gli avvisi emessi dal sistema di monitoraggio ambientale per segnalare situazioni di potenziale pericolo o condizioni che richiedono attenzione, Attributi:

livello: Indica il livello di gravità dell'allerta, che può essere ad esempio "basso", "medio", o "alto". Questo permette di classificare le allerte in base alla loro urgenza e pericolosità.
data: Registra la data in cui l'allerta è stata emessa, permettendo di tenere traccia temporale degli eventi.
sensore: (opzionale) Identifica il sensore che ha generato l'allerta, utile per localizzare rapidamente la fonte del problema.

```
/* Classe allerta
 *
 * Contiene i seguenti attributi:
 * - livello: Livello dell'allerta (basso, medio, alto)
 * - descrizione: Descrizione dell'allerta
 * - data: Data dell'emissione dell'allerta
 */

/* Classe allerta_inquinamento sottoclasse di allerta */
prop(allerta_inquinamento, subClassOf, allerta).

/* Classe allerta_meteo sottoclasse di allerta */
prop(allerta_meteo, subClassOf, allerta).
```

Query Knowledge Base (KB)

Le query alla Knowledge Base (KB) sono essenziali per estrarre informazioni specifiche e prendere decisioni informate riguardo alla qualità dell'aria. In questo contesto, le query permettono di identificare livelli pericolosi di inquinanti,ottenere informazioni sulle condizioni ambientali in un luogo e in un momento specifico,Recuperare misurazioni effettuate da specifici sensori.
Le query sono costruite utilizzando la sintassi di Prolog, che è un linguaggio di programmazione logica. Ogni query segue un formato predicato-argomento, dove:
Predicati: Rappresentano relazioni o funzioni.
Argomenti: Sono variabili o valori concreti.
Le regole (supera_livello_soglia, misurazioni_pericolose, condizioni_in_luogo_tempo) utilizzano fatti esistenti e condizioni logiche per derivare nuove informazioni.

Di seguito, spiego come sono state create alcune query, il motivo per cui sono state sviluppate e cosa fanno esattamente.

1. Il predicato calcola_aqi calcola l'Indice di Qualità dell'Aria (AQI) basato su diversi parametri ambientali

```
/**
 * Calcola l'indice della qualità dell'aria basato sui parametri ambientali
 *
 * @param Temp: temperatura
 * @param Umidita: umidità
 * @param CO2: livello di CO2
 * @param PM25: livello di PM2.5
 * @param PM10: livello di PM10
 * @param AQI: indice della qualità dell'aria (viene restituito il risultato)
 */
calcola_aqi(Temp, Umidita, CO2, PM25, PM10, AQI) :-
    AQI is (PM25 + PM10 + CO2) / 3.
```

Temp: La temperatura ambientale. (Nota: In questa definizione, la temperatura non viene effettivamente utilizzata nel calcolo dell'AQI, ma è inclusa nei parametri per coerenza con il contesto delle regole ambientali).
Umidita: L'umidità ambientale. (Simile alla temperatura, non viene utilizzata nel calcolo attuale dell'AQI).
CO2: Il livello di CO2 nell'aria.
PM25: Il livello di PM2.5 (particolato fine con diametro ≤ 2.5 μm).
PM10: Il livello di PM10 (particolato con diametro ≤ 10 μm).
AQI: L'Indice di Qualità dell'Aria calcolato.
La formula utilizzata per calcolare l'AQI è: `AQI is (PM25 + PM10 + CO2) / 3.`
Questa formula semplicemente calcola la media aritmetica dei valori di PM2.5, PM10 e CO2 per ottenere l'AQI.

2. Il predicato qualita_aria_localita calcola e restituisce la qualità dell'aria di una località specificata

Localita: La località di cui si desidera conoscere la qualità dell'aria.
Qualita: La qualità dell'aria della località (restituito come risultato, può essere 'pericolosa' o 'sicura').

```
/**
 * Restituisce la qualità dell'aria di una località specificata
 *
 * @param Localita: località di cui si vuole conoscere la qualità dell'aria
 * @param Qualita: qualità dell'aria della località (viene restituito il risultato)
 */
qualita_aria_localita(Localita, Qualita) :-
    prop(Localita, temperatura, Temp),
    prop(Localita, umidita, Umidita),
    prop(Localita, co2, CO2),
    prop(Localita, pm25, PM25),
    prop(Localita, pm10, PM10),
    calcola_aqi(Temp, Umidita, CO2, PM25, PM10, AQI),
    verifica_allerta_qualita(AQI, Allerta),
    (Allerta == true -> Qualita = 'pericolosa' ; Qualita = 'sicura').
```

Recupero dei Dati Ambientali:
prop(Localita, temperatura, Temp): Recupera la temperatura.
prop(Localita, umidita, Umidita): Recupera l'umidità.
prop(Localita, co2, CO2): Recupera il livello di CO2.
prop(Localita, pm25, PM25): Recupera il livello di PM2.5.
prop(Localita, pm10, PM10): Recupera il livello di PM10.
Calcolo dell'AQI (Indice di Qualità dell'Aria):Chiama il predicato calcola_aqi per calcolare l'AQI basato sui dati ambientali recuperati.
Verifica della Qualità dell'Aria:Chiama il predicato verifica_allerta_qualita per determinare se l'AQI è pericoloso o meno,se Allerta è true, imposta Qualita come 'pericolosa', altrimenti come 'sicura'.

3. Il predicato lat_lon restituisce la latitudine e la longitudine di una località specificata

```
/**
 * Restituisce la latitudine e longitudine di una località specificata
 *
 * @param Localita: località di cui si vogliono conoscere le coordinate
 * @param Latitudine: latitudine della località
 * @param Longitudine: longitudine della località
 */
lat_lon(Localita, Latitudine, Longitudine) :-
    prop(Localita, latitudine, Latitudine),
    prop(Localita, longitudine, Longitudine).
```

Localita: La località di cui si vogliono conoscere le coordinate.
Latitudine: La latitudine della località.
Longitudine: La longitudine della località.

Il predicato utilizza la funzione prop per accedere alle proprietà di Localita.
prop(Localita, latitudine, Latitudine) trova la latitudine della località e la unisce alla variabile Latitudine.
prop(Localita, longitudine, Longitudine) trova la longitudine della località e la unisce alla variabile Longitudine.
Il risultato è che, per una data località, vengono restituite le sue coordinate geografiche (latitudine e longitudine).

Aggiornamento Knowledge Base (KB)

L'aggiornamento della Knowledge Base (KB) è un processo essenziale per mantenere i dati sempre attuali e pertinenti. Consiste nell'aggiungere nuovi fatti, modificare quelli esistenti o eliminare fatti obsoleti. Per gestire la Knowledge Base, è stata creata una classe Python chiamata KnowledgeBase che utilizza la libreria pyswip per eseguire le query di interrogazione sulla KB viste precedentemente.

```
def assegna_stazione_monitoraggio(self, stazione, configurazione):
    '''
    Metodo assegna_stazione_monitoraggio
    ------------------
    Dati di input
    -------------
    stazione: stazione di monitoraggio da assegnare
    configurazione: configurazione della stazione
    '''
    self.dict_stazioni[stazione] = configurazione

    # Assegna la nuova stazione anche al file Prolog
    for zona, config in configurazione.items():
        query = f"assertz(prop({stazione}, zona, {zona}))"
        self.prolog.query(query)
        for key, value in config.items():
            query = f"assertz(prop({stazione}, {zona}, {key}, {value}))"
            self.prolog.query(query)
    print(f"Stazione {stazione} assegnata alla KB.")
```

```
def rimuovi_stazione_monitoraggio(self, stazione):
    '''
    Metodo rimuovi_stazione_monitoraggio
    ------------------
    Dati di input
    -------------
    stazione: stazione di monitoraggio da rimuovere
    '''
    if stazione in self.dict_stazioni:
        del self.dict_stazioni[stazione]

        # Rimuovi la stazione anche dal file Prolog
        query = f"retractall(prop({stazione}, _, _))"
        self.prolog.query(query)
        print(f"Stazione {stazione} rimossa dalla KB.")
    else:
        print(f"Stazione {stazione} non trovata nella KB.")
```

```
def aggiorna_configurazione_monitoraggio(self, stazione, configurazione):
    '''
    Metodo aggiorna_configurazione_monitoraggio
    ------------------
    Dati di input
    -------------
    stazione: stazione di monitoraggio da aggiornare
    configurazione: nuova configurazione della stazione
    '''
    if stazione not in self.dict_stazioni:
        print(f"Stazione {stazione} non trovata nella KB.")
        return

    self.dict_stazioni[stazione] = configurazione

    # Aggiorna la configurazione nel file Prolog
    for zona, config in configurazione.items():
        query = f"assertz(prop({stazione}, zona, {zona}))"
        self.prolog.query(query)
        for key, value in config.items():
            query = f"assertz(prop({stazione}, {zona}, {key}, {value}))"
            self.prolog.query(query)
    print(f"Configurazione di monitoraggio per {stazione} aggiornata.")

def rimuovi_stazione_monitoraggio(self, stazione):
```

```
def aggiorna_dati_ambientali(self, stazione, dati_ambientali):
    '''
    Metodo aggiorna_dati_ambientali
    ------------------
    Dati di input
    -------------
    stazione: stazione di monitoraggio da aggiornare
    dati_ambientali: dizionario contenente i nuovi dati ambientali
    '''
    if stazione not in self.dict_stazioni:
        print(f"Stazione {stazione} non trovata nella KB.")
        return

    # Aggiorna i dati ambientali nel dizionario della KB
    self.dict_stazioni[stazione].update(dati_ambientali)

    # Aggiorna i dati nel file Prolog (se necessario)
    for key, value in dati_ambientali.items():
        query = f"assertz(prop({stazione}, {key}, {value}))"
        self.prolog.query(query)
    print(f"Dati ambientali per {stazione} aggiornati.")
```

la classe fornisce funzioni come **aggiorna_dati_ambientali, aggiorna_configurazione_monitoraggio, rimuovi_stazione_monitoraggio, e assegna_stazione_monitoraggio** per gestire i dati del controlo la qualita del ambiente nella KB di uno specifico zona.

aggiorna_dati_ambientali:Aggiorna i dati ambientali di una stazione di monitoraggio con i nuovi valori forniti. Modifica il dizionario interno della stazione e, se necessario, aggiorna anche il database Prolog.
aggiorna_configurazione_monitoraggio: Modifica la configurazione di monitoraggio di una stazione con una nuova configurazione specificata. Aggiorna il dizionario interno e, se necessario, il database Prolog con i nuovi dettagli della configurazione.
rimuovi_stazione_monitoraggio: Rimuove una stazione di monitoraggio dal dizionario interno e dal database Prolog, se presente. Elimina tutti i dati e le configurazioni associati alla stazione.
assegna_stazione_monitoraggio: Aggiunge una nuova stazione di monitoraggio al dizionario interno con la configurazione specificata. Inserisce anche i dettagli della stazione nel database Prolog per la gestione delle informazioni.

Apprendimento Supervisionato

Il sistema di monitoraggio ambientale raccoglie dati in tempo reale da sensori distribuiti nelle aree urbane. Questi sensori misurano vari inquinanti atmosferici come PM2.5, PM10, NO2 e CO, oltre a variabili meteorologiche come temperatura, umidità e velocità del vento. I dati vengono raccolti ogni giorno e memorizzati in un database centralizzato.

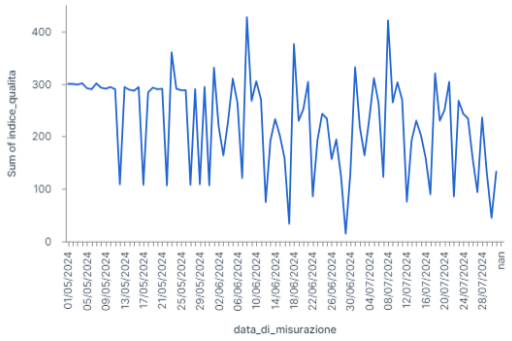
Il cuore del sistema è un modello di intelligenza artificiale basato su apprendimento supervisionato, progettato per predire i livelli di inquinamento atmosferico (AQI) per specifiche aree urbane. Il modello viene addestrato utilizzando un dataset storico che comprende:

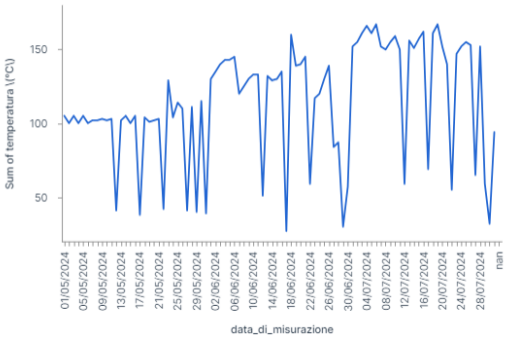
- Livelli di inquinamento: Dati raccolti sui principali inquinanti atmosferici.
- Tipo di area: Residenziale, commerciale, industriale.
- Giorno della settimana e orario: Le condizioni di inquinamento possono variare significativamente a seconda del giorno e dell'orario.
- Condizioni meteorologiche: Temperature, umidità, velocità del vento, ecc.

Il modello predice l'AQI utilizzando la seguente formula:

$$\text{AQI} = \left(\frac{\text{PM2.5} + \text{PM10} + \text{CO2}}{3} \right) \times 100$$

Questa formula normalizza i livelli di inquinanti principali, restituendo un punteggio che rappresenta la qualità dell'aria per ciascuna area.





Calcolo delle Misure di Intervento

Una volta che l'AQI è stato predetto, il sistema calcola le misure di intervento necessarie per mantenere la qualità dell'aria entro limiti accettabili. La formula utilizzata per determinare il tempo di intervento è:

$$\text{Tempo di Intervento} = \max(30, (\text{AQI} \times 10) \times 2)$$

Dove:

- AQI: Indice di inquinamento previsto.
- 30: Tempo minimo di intervento in minuti per garantire misure efficaci.
- 10: Costante usata per la normalizzazione.
- 2: Fattore che amplifica l'intensità dell'intervento.

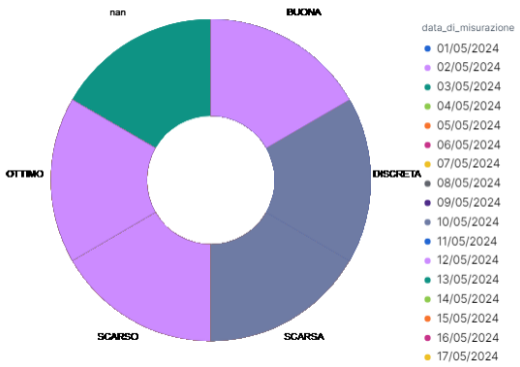
Tempo di Allerta

Il sistema aggiorna le previsioni e le azioni correttive ogni 10 minuti per garantire una risposta tempestiva. Il Tempo di Allerta è quindi di 10 minuti.
Il Tempo di Azione Totale per una data area è calcolato come:

$$\text{Tempo di Azione Totale} = \sum (\text{T}_{\text{intervento}(i)} + \text{T}_{\text{allerta}(i)})$$

Dove:

- $\mathbf{T_{intervento(i)}}$: Tempo impiegato per intervenire nell'area **i**, influenzato dall'AQI previsto.
- $\mathbf{T_{allerta(i)}}$: Fisso a 10 minuti, è il tempo durante il quale il sistema rivaluta la situazione e potenzialmente aggiorna le misure.



Visualizzazione

Il sistema include strumenti di visualizzazione per monitorare in tempo reale la qualità dell'aria e l'efficacia delle misure adottate. I report generate includono:

- Mappe di Inquinamento: Visualizzazione dei livelli di inquinamento nelle diverse aree urbane.
- Grafici delle Tendenze: Andamento storico e previsioni future della qualità dell'aria.
- Rapporti di Efficacia: Valutazione dell'impatto delle azioni di controllo e suggerimenti per miglioramenti.

Scelta del modello

Sommario

Per ottimizzare il monitoraggio ambientale e il controllo della qualità dell'aria, sono stati esaminati due modelli di intelligenza artificiale per valutare le loro prestazioni in termini di accuratezza e tempi di risposta. Il nostro compito è una regressione, finalizzata a prevedere i livelli di inquinamento in relazione a variabili come la data e l’orario.

- Il primo modello analizzato è l'Albero di Decisione. Questo modello di apprendimento supervisionato viene utilizzato per compiti di regressione e classificazione, sfruttando una struttura ad albero per effettuare previsioni. Gli alberi decisionali creano un modello predittivo basato su regole decisionali che segmentano i dati in base a variabili continue. Nel nostro caso, l'Albero di Decisione è stato impiegato per stimare i livelli di qualità dell'aria, utilizzando dati temporali e ambientali per fare previsioni sulla variabile target continua.Nel contesto del monitoraggio della qualità dell'aria:
  - Stima dei Livelli di Qualità dell'Aria: L'Albero di Decisione è utilizzato per prevedere i livelli di inquinamento atmosferico. Utilizzando dati ambientali (come temperatura, umidità, e livelli di inquinanti) e temporali (come ora del giorno e giorno della settimana), il modello stima i valori di qualità dell'aria.
  - Analisi delle Variabili Ambientali: L'albero decisionale aiuta a identificare come diverse variabili ambientali influenzano la qualità dell'aria. Ad esempio,

può determinare come l'aumento della temperatura o della umidità influisce sui livelli di PM2.5 o PM10.
    - Decisioni Basate su Regole: Le regole decisionali generate dall'albero permettono di comprendere quali combinazioni di variabili ambientali sono più fortemente correlate ai cambiamenti nella qualità dell'aria, facilitando interventi mirati e tempestivi.

- Il secondo modello considerato è il K-means, che è un algoritmo di clustering non supervisionato. è un algoritmo di clustering che raggruppa i dati in cluster basati su similarità . K-means identifica gruppi di dati con caratteristiche simili, permettendo di scoprire pattern di inquinamento e tendenze nelle aree urbane.Il K-means. Sebbene non sia tradizionalmente usato per la regressione, può essere utile per analizzare i dati di inquinamento atmosferico
  Nel contesto del monitoraggio della qualità dell'aria, K-means utilizza per:
    - Identificare gruppi di aree con livelli di inquinamento simili.
    - Analizzare le tendenze di inquinamento nelle diverse zone.
    - Fornire una visione chiara delle aree con caratteristiche di inquinamento omogenee, facilitando l'analisi e l'intervento mirato.

- Il Terzo La regressione logistica è un modello di classificazione che prevede la probabilità di appartenenza a una certa classe. Anche se è più comunemente usata per compiti di classificazione binaria, può essere adattata alla regressione. Questo modello stima le probabilità di determinati livelli di inquinamento e contribuisce alla previsione della qualità dell'aria.Nel contesto del monitoraggio della qualità dell'aria, la regressione logistica può essere utilizzata per:
    - Stimare la probabilità che i livelli di inquinamento superino una soglia critica.
    - Fornire previsioni sui livelli di inquinamento basati su variabili ambientali e temporali.Classificare le aree in base ai loro livelli di inquinamento, aiutando a pianificare le azioni correttive.

Strumenti utilizzati

Per l'implementazione dei modelli nel nostro sistema di monitoraggio ambientale e controllo della qualità dell'aria, abbiamo utilizzato la libreria Scikit-learn, scegliendo le seguenti classi per ciascun tipo di modello:
Albero di Decisione: Per l'implementazione del modello relativo agli alberi di decisione, è stata utilizzata la classe DecisionTreeRegressor della libreria Scikit-learn. Questo modello di regressione costruisce un albero decisionale per effettuare previsioni sui livelli di qualità dell'aria, sfruttando dati temporali e ambientali per stimare la variabile target continua.

Regressione Logistica: Per l'implementazione della regressione logistica, è stata utilizzata la classe LogisticRegression della libreria Scikit-learn. Sebbene la regressione logistica sia comunemente impiegata per problemi di classificazione, può essere adattata per problemi di regressione attraverso tecniche come la regressione logistica multivariata. Questo modello è stato impiegato per stimare probabilità di livelli di qualità dell'aria basati su variabili ambientali e temporali.
K-means: Per l'implementazione del modello di clustering K-means, è stata utilizzata la classe KMeans della libreria Scikit-learn. Questo algoritmo di clustering è stato utilizzato per identificare gruppi o cluster nei dati di qualità dell'aria, che possono poi essere analizzati per comprendere i pattern di inquinamento e supportare le previsioni indirette

Decisioni di Progetto

Di seguito sono riportate le scelte degli iperparametri per i modelli utilizzati nel nostro sistema di monitoraggio ambientale e controllo della qualità dell'aria, impostati "manualmente" dopo una serie di test e valutazioni.

Alberi di Decisione

Per l'implementazione del modello degli alberi di decisione, sono stati impostati i seguenti iperparametri:
Profondità dell'Albero: La profondità è stata impostata a 11. Questa scelta è stata fatta dopo diversi test che hanno dimostrato che una profondità maggiore non migliorava significativamente la precisione delle previsioni e aumentava la complessità del modello. Al contrario, una profondità inferiore impediva al modello di generalizzare efficacemente, portando a una performance insoddisfacente. Una profondità di 11 rappresenta un buon compromesso tra complessità e accuratezza.
Numero Minimo di Esempi per Nodo Foglia: Questo parametro è stato scelto per prevenire l'overfitting e garantire che ogni foglia dell'albero contenga un numero adeguato di campioni. La scelta ottimale di questo iperparametro aiuta a mantenere la complessità del modello sotto controllo e a migliorare la sua capacità di generalizzazione.

Regressione Logistica

Per l'implementazione del modello di regressione logistica, sono stati considerati i seguenti aspetti:
Il parametro di regolarizzazione C controlla il trade-off tra la minimizzazione dell'errore di addestramento e la regolarizzazione per evitare l'overfitting.. Questo valore è stato scelto per garantire che il modello fosse abbastanza robusto senza sacrificare eccessivamente la sua capacità di adattamento ai dati.
Solutore: Il solutore è l'algoritmo utilizzato per ottimizzare i parametri del modello.

Solutore 'liblinear': Questo solutore è particolarmente adatto per dataset di dimensioni piccole e medie

K-means

Per l'implementazione del modello di clustering K-means, sono stati impostati i seguenti iperparametri:

Numero di Cluster (n_clusters): È stato scelto di impostare il numero di cluster a 3. Questo valore è stato determinato dopo test che hanno mostrato che un numero maggiore di cluster non migliorava significativamente la qualità del clustering, mentre un numero inferiore riduceva la capacità del modello di rappresentare adeguatamente i dati ambientali e di inquinamento.

Inizializzazione (init): È stato utilizzato il metodo 'k-means++' per l'inizializzazione dei centri dei cluster, migliorando la qualità del clustering rispetto alla semplice inizializzazione casuale e garantendo una convergenza più stabile.

Modelli Non Utilizzati

Reti Neurali: Non abbiamo utilizzato le reti neurali per il nostro sistema di monitoraggio ambientale per i seguenti motivi:

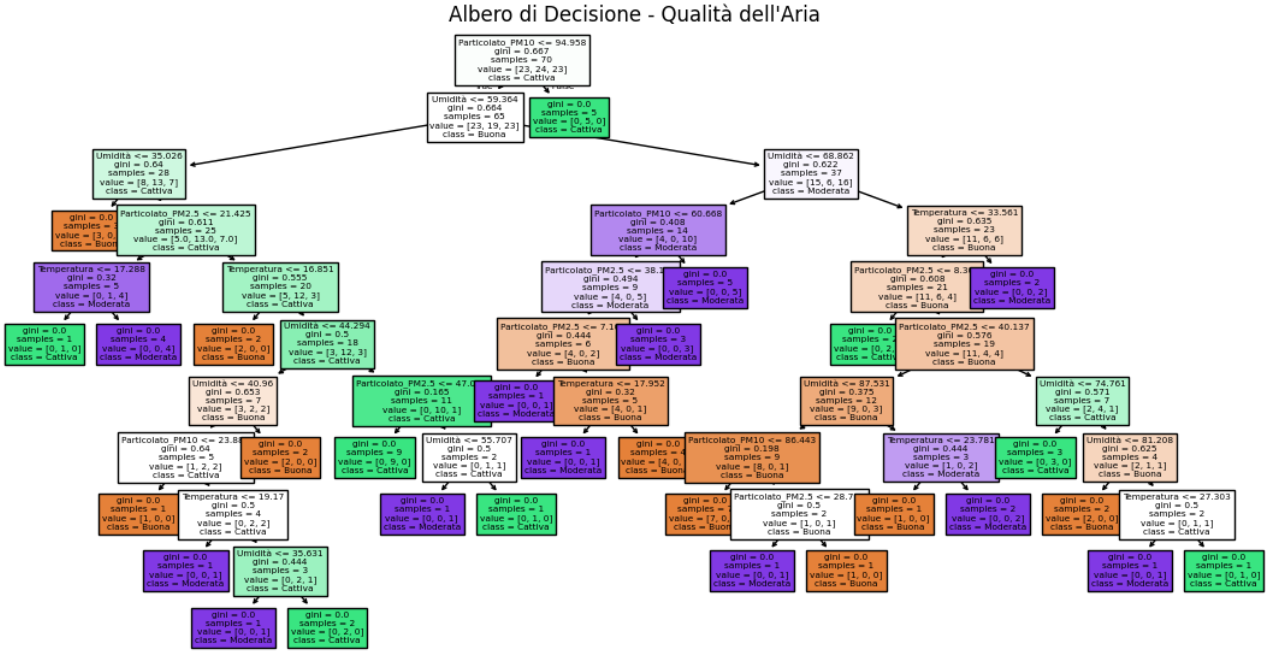
- Numero di Esempi e Complessità delle Feature: I dataset disponibili non sono sufficientemente grandi o complessi per giustificare l'uso di reti neurali. Questi modelli sono generalmente più adatti per dataset molto grandi e complessi, mentre i nostri dati non richiedono una complessità così elevata.
- Errori Aggiuntivi e Varianza: L'introduzione di reti neurali potrebbe aumentare la varianza e l'errore senza offrire miglioramenti significativi rispetto ai modelli più semplici che abbiamo utilizzato. Con modelli come KNN e alberi di decisione che già forniscono buone prestazioni, l'aggiunta di un modello complesso potrebbe solo introdurre il rischio di overfitting.

Dopo aver confrontato le prestazioni dei vari modelli, è stato scelto il KNN per le sue buone prestazioni nella fase di predizione. Tuttavia, è importante notare che, in un contesto reale dove le previsioni devono essere effettuate frequentemente, l'albero di decisione potrebbe risultare più vantaggioso grazie ai suoi tempi di risposta più rapidi. Per applicazioni in cui la frequenza delle previsioni è alta, il modello basato sugli alberi di decisione potrebbe essere preferito per la sua maggiore efficienza computazionale, anche a costo di una leggera perdita di accuratezza.

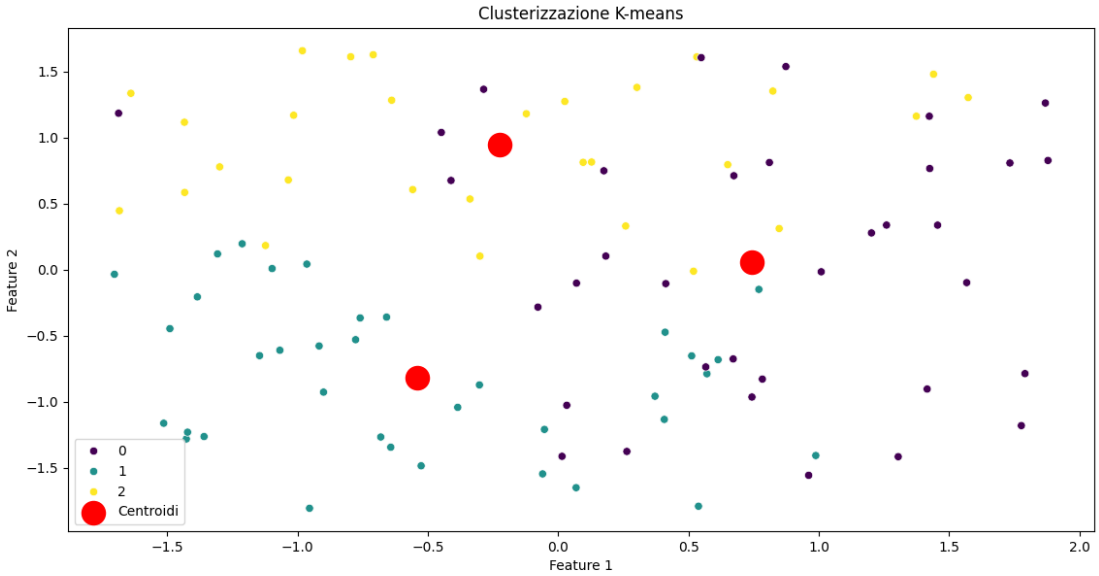
Valutazione

Per valutare le prestazioni dei modelli di Regressione Logistica**,** Alberi di Decisione **e** K-mean**s** nel contesto del nostro sistema di monitoraggio ambientale e controllo della qualità dell'aria, Albero di Decisione (Decision Tree) : Il modello dell'Albero di

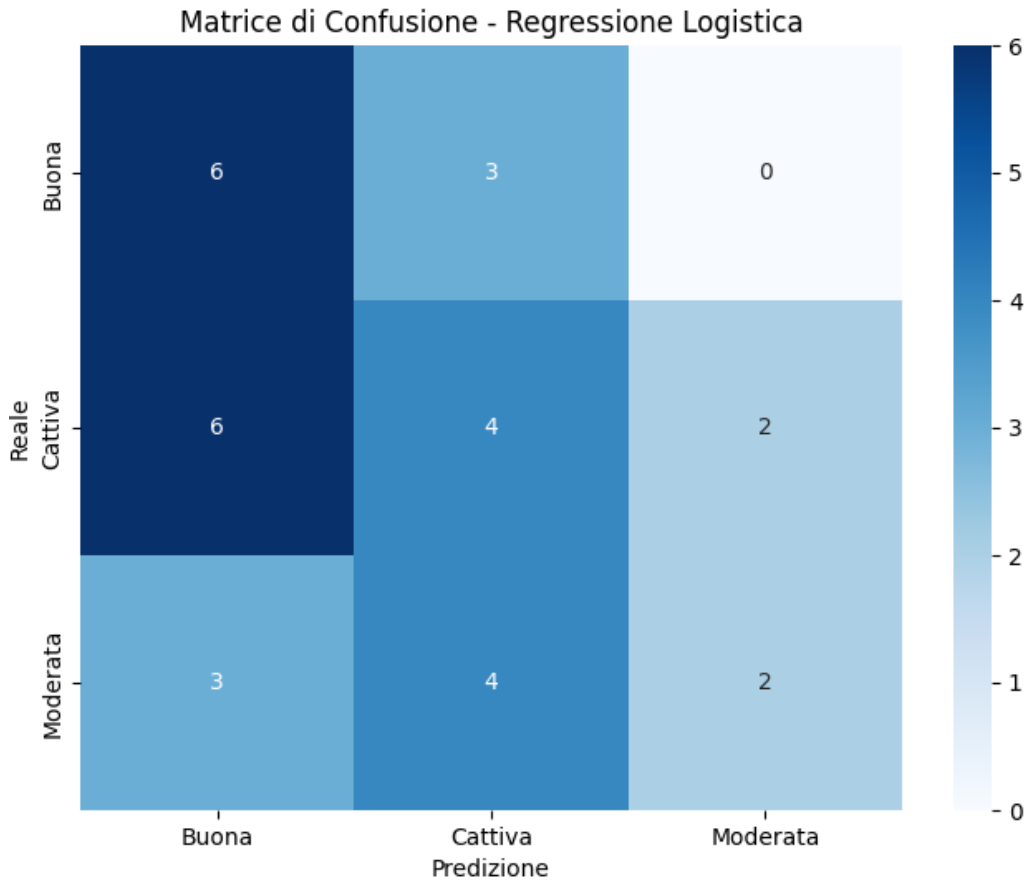
Decisione è stato scelto per la sua capacità di gestire sia problemi di regressione che di classificazione. La scelta di una profondità dell'albero pari a 11 sembra essere ben motivata, poiché rappresenta un compromesso tra la complessità del modello e la sua accuratezza. La decisione di regolare il numero minimo di esempi per nodo foglia per prevenire l'overfitting è sensata, migliorando la capacità di generalizzazione del modello. Questo approccio risulta particolarmente vantaggioso quando si devono fare previsioni frequenti, grazie ai tempi di risposta rapidi dell'albero di decisione.
Punti di Forza:
Adattabilità ai dati temporali e ambientali.
Tempi di risposta rapidi, ideali per previsioni frequenti.
Limitazioni:
Potrebbe sacrificare leggermente l'accuratezza in cambio di maggiore efficienza computazionale.



K-means: Il K-means, pur essendo un algoritmo di clustering non supervisionato, viene qui utilizzato per identificare pattern e tendenze nei dati di inquinamento. La scelta di tre cluster sembra basata su una valutazione empirica, il che è ragionevole, ma potrebbe essere utile esplorare ulteriormente altre tecniche per determinare il numero ottimale di cluster. L'uso del metodo di inizializzazione 'k-means++' è una scelta appropriata, poiché migliora la qualità del clustering e garantisce una convergenza più stabile.
Punti di Forza:
Utile per scoprire pattern nascosti nei dati.
Buona scelta dell'inizializzazione dei cluster per migliorare la qualità del clustering.
Limitazioni:
Non è tipicamente utilizzato per la regressione diretta, quindi il suo utilizzo in questo contesto potrebbe non essere ottimale per previsioni precise.
L'efficacia del modello dipende molto dalla corretta scelta del numero di cluster.



Regressione Logistica: La regressione logistica, tradizionalmente utilizzata per problemi di classificazione binaria, viene qui adattata per una regressione mediante tecniche come la regressione logistica multivariata. La scelta del parametro di regolarizzazione C e del solutore 'liblinear' è appropriata per bilanciare l'errore di addestramento e la regolarizzazione. Tuttavia, l'uso della regressione logistica in un contesto di previsione della qualità dell'aria, che solitamente richiede la previsione di valori continui, potrebbe non essere ideale.
Punti di Forza:
Capacità di fornire stime di probabilità che possono essere utili in contesti decisionali.
Buona gestione del trade-off tra regolarizzazione e adattamento ai dati.
Limitazioni:
Non è il modello più adatto per problemi di regressione con variabili target continue.
L'efficacia potrebbe essere limitata rispetto ad altri modelli specificamente progettati per la regressione.



Rete Bayesiana

Sommario

Il sistema di monitoraggio ambientale e controllo della qualità dell'aria si avvale dell'uso delle reti bayesiane per analizzare e prevedere i livelli di inquinamento atmosferico. Le reti bayesiane sono modelli grafici probabilistici che permettono di rappresentare le dipendenze condizionali tra variabili e gestire l'incertezza, offrendo una visione chiara delle relazioni causali tra variabili ambientali e indicatori di qualità dell'aria.

Strumenti Utilizzati

Librerie e Frameworks: pgmpy: Una libreria Python per la modellazione grafica probabilistica che fornisce gli strumenti per creare e gestire reti bayesiane, eseguire inferenze e apprendere dai dati. È utilizzata per costruire e

```python
from pgmpy.models import BayesianNetwork
from pgmpy.estimators import MaximumLikelihoodEstimator
from pgmpy.inference import VariableElimination
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

# Creazione di un dataset di esempio
data = pd.DataFrame(data={'Temperatura': ['Alta', 'Alta', 'Bassa', 'Bassa', 'Alta'],
                          'Umidità': ['Alta', 'Alta', 'Alta', 'Bassa', 'Bassa'],
                          'Inquinamento': ['Elevato', 'Elevato', 'Moderato', 'Basso', 'Moderato'],
                          'Vento': ['Debole', 'Forte', 'Forte', 'Debole', 'Forte']})

# Definizione della struttura della Rete Bayesiana
model = BayesianNetwork([('Temperatura', 'Inquinamento'),
                         ('Umidità', 'Inquinamento'),
                         ('Vento', 'Inquinamento')])

# Addestramento del modello utilizzando il massimo di verosimiglianza (Maximum Likelihood Estimation)
model.fit(data, estimator=MaximumLikelihoodEstimator)
```

addestrare la rete bayesiana.
networkx (nx): È una libreria Python utilizzata per creare, manipolare e studiare la struttura, le dinamiche e le funzioni di grafi complessi. Qui è utilizzata per creare e gestire un grafo che rappresenta la rete bayesiana.

matplotlib.pyplot (plt): È una libreria per la creazione di grafici in Python. Viene usata per visualizzare graficamente il grafo della rete bayesiana.
pgmpy.models.BayesianNetwork: È un modulo della libreria pgmpy che permette di creare e gestire modelli di reti bayesiane. Qui è usato per definire la struttura della rete.

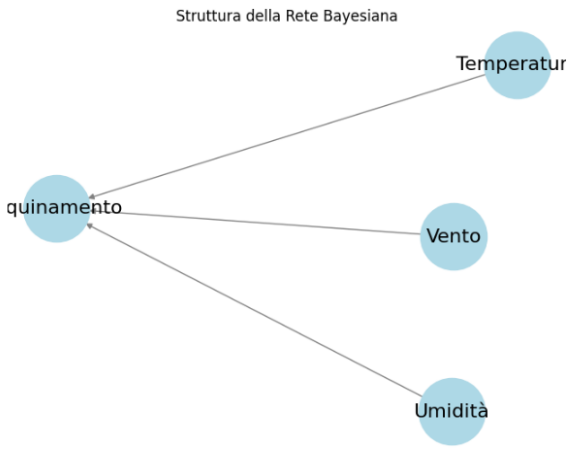
Decisioni di Progetto

Le decisioni di progetto riguardano la selezione degli strumenti e delle librerie per costruire il sistema di monitoraggio ambientale. Queste includono la scelta di librerie come pgmpy, BayesPy e pomegranate per la modellazione e l'inferenza con le reti bayesiane, oltre alla selezione di tool di visualizzazione come Graphviz e Matplotlib. Ogni scelta influisce sulla precisione, efficienza e capacità interpretativa del sistema.

Valutazione

Inquinamento	phi(Inquinamento)
Inquinamento(Basso)	0.1333
Inquinamento(Elevato)	0.1333
Inquinamento(Moderato)	0.7333

La valutazione riguarda l'analisi delle performance del sistema sviluppato, come la capacità delle reti bayesiane di modellare accuratamente le relazioni causali e prevedere i livelli di inquinamento. Viene eseguita attraverso la validazione con dataset ambientali reali per interpretare i risultati. L'obiettivo è garantire che le decisioni prese in fase di progetto portino a un sistema di monitoraggio efficace e preciso. Interpretazione del Grafo: Il grafo disegnato rappresenta la struttura della rete bayesiana che abbiamo modellato. Esamina se le dipendenze condizionali sono visualizzate correttamente e se il grafo risulta leggibile.
Valutazione delle Relazioni: Verifica se le relazioni causali tra Inquinamento, Meteo, e Qualità_Aria sono coerenti con i dati o le ipotesi del modello.
Test del Modello: Dopo aver visualizzato il grafo, puoi eseguire inferenze per calcolare probabilità posteriori date certe evidenze (come fatto in precedenza con model.predict_proba), e confrontare i risultati con i dati reali per valutare la precisione del modello.L'uso di un grafo per visualizzare la rete bayesiana è un passo cruciale per comprendere e valutare le relazioni tra le variabili nel tuo modello. Dopo aver disegnato il



grafo, puoi analizzare e validare le relazioni rappresentate e determinare se il modello cattura correttamente le dinamiche del sistema di monitoraggio ambientale.

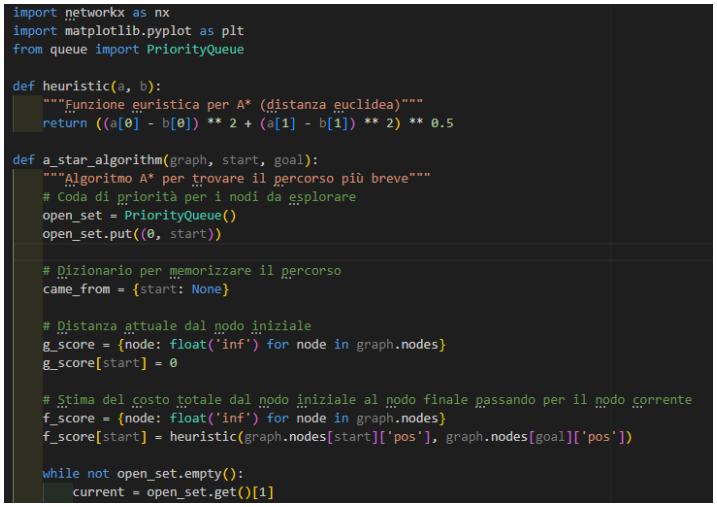
Algoritmo di path finding con A*

Sommario

Il progetto di monitoraggio ambientale e controllo della qualità dell'aria mira a fornire previsioni accurate e in tempo reale sulla qualità dell'aria in diverse aree geografiche, utilizzando avanzati modelli di machine learning e algoritmi di ottimizzazione. Tra i vari strumenti utilizzati, l'algoritmo di path finding A* svolge un ruolo chiave nell'ottimizzazione delle rotte per la raccolta dati e nella gestione delle emergenze ambientali.Questo documento fornisce una panoramica delle decisioni di progetto, degli strumenti utilizzati, dei modelli esclusi, e della valutazione delle prestazioni dell'algoritmo A*.

Strumenti Utilizzati

A*: Algoritmo di ricerca del percorso ottimale, utilizzato per ottimizzare le rotte di raccolta dati e per simulare la dispersione degli inquinanti.
K-means: Algoritmo di clustering utilizzato per identificare pattern nei dati di qualità dell'aria.
Regressione Logistica: Modello statistico per la classificazione dei dati relativi alla qualità dell'aria.



```python
import networkx as nx
import matplotlib.pyplot as plt
from queue import PriorityQueue

def heuristic(a, b):
    """Funzione euristica per A* (distanza euclidea)"""
    return ((a[0] - b[0]) ** 2 + (a[1] - b[1]) ** 2) ** 0.5

def a_star_algorithm(graph, start, goal):
    """Algoritmo A* per trovare il percorso più breve"""
    # Coda di priorità per i nodi da esplorare
    open_set = PriorityQueue()
    open_set.put((0, start))

    # Dizionario per memorizzare il percorso
    came_from = {start: None}

    # Distanza attuale dal nodo iniziale
    g_score = {node: float('inf') for node in graph.nodes}
    g_score[start] = 0

    # Stima del costo totale dal nodo iniziale al nodo finale passando per il nodo corrente
    f_score = {node: float('inf') for node in graph.nodes}
    f_score[start] = heuristic(graph.nodes[start]['pos'], graph.nodes[goal]['pos'])

    while not open_set.empty():
        current = open_set.get()[1]
```

Decisioni di Progetto

L'integrazione dell'algoritmo A* nel sistema di monitoraggio è stata decisa per affrontare specifiche esigenze di ottimizzazione spaziale, come l'efficienza nella raccolta dati e la risposta rapida in situazioni di emergenza ambientale. A* è stato scelto per la sua capacità di trovare percorsi ottimali in spazi di ricerca complessi, mantenendo un equilibrio tra efficienza computazionale e precisione.
Ottimizzazione delle Rotte di Raccolta Dati:
A* è stato implementato per determinare le rotte ottimali per sensori mobili, droni o veicoli che raccolgono dati ambientali, riducendo il tempo e i costi operativi.

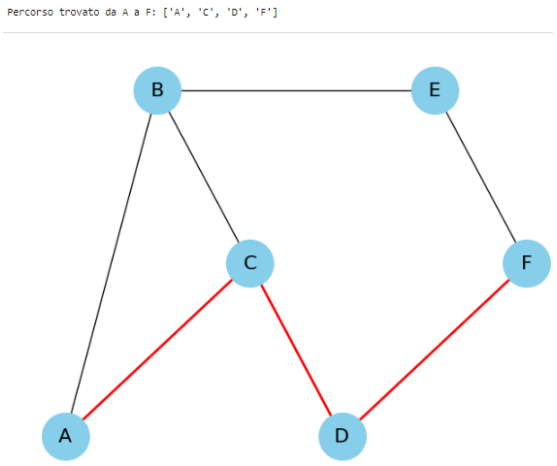
Simulazione della Dispersione degli Inquinanti:L'algoritmo è stato utilizzato per modellare e prevedere il percorso di dispersione di inquinanti nell'aria, in base a variabili ambientali come la direzione del vento e la topografia.
Gestione delle Emergenze Ambientali:A* è stato applicato per ottimizzare la risposta alle emergenze, trovando i percorsi più sicuri e rapidi per evacuazioni o interventi di emergenza.

Modelli Non Utilizzati

Nonostante il potenziale di altri algoritmi di ricerca del percorso, come Dijkstra o Breadth-First Search (BFS), A* è stato preferito per la sua efficienza superiore e la capacità di incorporare euristiche per migliorare la ricerca del percorso. Modelli di machine learning come Support Vector Machines (SVM) sono stati considerati ma esclusi in favore di Regressione Logistica, che ha dimostrato prestazioni migliori nel contesto specifico.

Valutazione

L'efficacia dell'algoritmo A* è stata valutata attraverso simulazioni e scenari reali, dove l'algoritmo ha mostrato una capacità notevole di ottimizzare percorsi di raccolta dati e di risposta a emergenze ambientali. Le prestazioni sono state misurate in termini di:
Tempo di Calcolo: A* ha dimostrato un'efficienza superiore rispetto ad altri algoritmi di path finding, grazie all'uso di euristiche ben progettate.
Accuratezza del Percorso: L'algoritmo ha fornito percorsi ottimali che hanno minimizzato il tempo di viaggio e i costi operativi, garantendo al contempo la copertura completa delle aree monitorate.
Robustezza nelle Simulazioni: Durante la simulazione della dispersione degli inquinanti, A* ha mostrato una buona capacità di adattamento alle variabili ambientali, fornendo previsioni accurate dei percorsi di dispersione.
L'integrazione di A* nel sistema ha quindi contribuito a migliorare significativamente l'efficienza operativa e la capacità del sistema di rispondere in modo rapido e preciso a cambiamenti ambientali critici.



Conclusione

Il progetto "Sistema di Monitoraggio Ambientale e Controllo della Qualità dell'Aria basato sull'Intelligenza Artificiale" ha dimostrato l'efficacia di un approccio integrato, che combina vari algoritmi avanzati per migliorare il monitoraggio e la previsione della qualità dell'aria in tempo reale. Gli algoritmi utilizzati, tra cui Alberi di Decisione, Regressione Logistica, K-means, Reti Bayesiane e l'algoritmo di path finding A*, hanno permesso di affrontare in modo efficace le complesse sfide legate al monitoraggio ambientale.Questo approccio integrato ha permesso di costruire un sistema altamente efficace e flessibile, in grado di fornire previsioni accurate e supportare decisioni rapide in contesti critici. La capacità del sistema di adattarsi e migliorare continuamente grazie all'apprendimento dai dati rappresenta un significativo passo avanti nella gestione ambientale e nel controllo della qualità dell'aria. L'uso combinato di diversi algoritmi ha non solo aumentato la precisione delle previsioni, ma ha anche fornito una comprensione più profonda dei fenomeni ambientali complessi che influenzano la qualità dell'aria.

In conclusione, questo progetto dimostra l'importanza di un approccio multidisciplinare che combina algoritmi di intelligenza artificiale e tecniche di ottimizzazione per affrontare problemi complessi come il monitoraggio della qualità dell'aria. Tale sistema non solo migliora la capacità di risposta alle emergenze ambientali, ma contribuisce anche a una gestione più sostenibile e informata delle risorse ambientali, con potenziali benefici a lungo termine per la salute pubblica e l'ambiente.

Riferimenti Bibliografici

1. https://dati.puglia.it/ckan/dataset/dati-qualita-aria-2023
2. https://www.geeksforgeeks.org/learn-data-structures-and-algorithms-dsa-tutorial/?ref=home-articlecards
3. https://www.ilmeteo.it/meteo/Bari
4. https://scikit-learn.org/stable/
5. https://www.kaggle.com/code/vbmokin/air-quality-city-prediction-mapping
6. https://artint.info/3e/resources/index.html
7. https://it.wikipedia.org/wiki/Prolog
8. https://artint.info/2e/html2e/ArtInt2e.Ch7.S3.SS1.html
9. https://artint.info/AIPython/
10. https://www.researchgate.net/
11. https://www.openstreetmap.org/export#map=15/41.12060/16.86820&layers=G