



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea in Ingegneria Gestionale

Classe L-8

A.A. 2023/2024

Sessione di Laurea Marzo 2024

Tesi di Laurea Triennale

**Simulazione della Propagazione di un
Incendio**

Relatore:
Fulvio Corno

Candidato:
Matteo Tomatis 281159

Sommario

1. Proposta di Progetto	3
1. Descrizione del problema proposto	3
2. Descrizione della rilevanza gestionale del problema	3
3. Descrizione dei data-set per la valutazione.....	3
4. Descrizione preliminare degli algoritmi coinvolti.....	3
5. Descrizione preliminare delle funzionalità previste per l'applicazione software.....	3
2. Descrizione del Problema Affrontato.....	5
1. Una Panoramica Generale.....	5
2. Le Maggiori Sfide Affrontate	5
3. Descrizione del Data-Set.....	7
4. Descrizione ad Alto Livello delle Strutture Dati e degli Algoritmi.....	8
1. Organizzazione Interna.....	8
2. Algoritmi Principali	9
5. Diagramma delle Classi Principali	12
1. Package "Model"	12
2. Package "Db"	12
3. Package "Controller"	13
6. L'Applicazione.....	14
7. Risultati Sperimentali.....	16
8. Valutazioni e Conclusioni	17
Nota sulle Fonti e le Immagini	17

1. Proposta di Progetto

1. Descrizione del problema proposto

Lo scopo del programma è quello di utilizzare i dati raccolti sugli incendi in Sardegna nel decennio 2010-2019 per simulare nel modo più realistico possibile la propagazione delle fiamme in un'area boschiva fittizia nella quale le principali condizioni climatiche saranno liberamente determinabili dall'utente.

2. Descrizione della rilevanza gestionale del problema

Uno degli effetti più evidenti e dannosi del cambiamento climatico è stato quello di favorire la nascita

e la diffusione di incendi. L'attività di prevenzione e cura delle foreste può limitare i danni causati da questo tipo di evento e pertanto può essere di interesse generale conoscere come si modifichi la propagazione delle fiamme in relazione ad una moltitudine di parametri climatici e ambientali, tra i quali la densità di vegetazione.

3. Descrizione dei data-set per la valutazione

Per valutare l'influenza dei fattori ambientali nella propagazione degli incendi si farà riferimento ad un dataset contenente i dati storici sugli incendi in Sardegna nel decennio 2009-2019 raccolti grazie ai satelliti del Copernicus Climate Change Service (C3S) e il Copernicus Atmosphere Monitoring Service (CAMS).

L'area interessata si estende per più di 50.000 km² e i dati sono stati raggruppati per mese ed anno. Il dataset può essere trovato qui: <https://www.kaggle.com/datasets/christianmolliere/wildfires-and-climate-in-sardinia>.

Per ogni record, oltre alla misura in termini sia assoluti che percentuali di terra bruciata, vengono riportati numerosi fattori climatici tra i quali: le temperature medie, le precipitazioni, la velocità media del vento, la quantità di acqua immagazzinata dal terreno, e la percentuale di terreno ricoperta da vegetazione fitta o bassa.

4. Descrizione preliminare degli algoritmi coinvolti

Il software sarà realizzato in linguaggio Java con l'ausilio dell'interfaccia grafica in JavaFX, inoltre saranno usati i pattern MVC e DAO.

Tra le funzioni coinvolte, le principali riguardano: l'estrazione dei dati dal database SQL, l'elaborazione dei dati mediante la tecnica della Regressione Lineare Multipla al fine di individuare il collegamento tra l'espandersi degli incendi e i dati ambientali, una funzione capace di creare un grafo (rappresentante la "foresta") le cui celle saranno connesse con una probabilità p determinata incrociando i risultati della regressione con i dati inseriti dall'utente, ed infine un algoritmo che simuli il propagarsi di un incendio tra celle connesse.

Infine, saranno implementate diverse funzioni per gestire gli input dell'utente (in particolare riguardanti la validazione dei dati) e l'output visualizzato a schermo.

5. Descrizione preliminare delle funzionalità previste per l'applicazione software

Nell'interfaccia grafica iniziale l'utente potrà inserire i dati ambientali (temperatura, precipitazioni, velocità del vento) sui quali avviare la simulazione. Inoltre, sarà mostrata una griglia rappresentante una foresta le cui celle vengono inizialmente mostrate di colore verde (a simboleggiare la presenza di vegetazione). Alla pressione del tasto dedicato all'avvio della simulazione, il programma calcolerà la propagazione di un incendio partendo da una cella casuale

e mostrerà all'utente il risultato finale colorando in modo appropriato le celle in cui l'incendio è passato.

2. Descrizione del Problema Affrontato

1. Una Panoramica Generale

Lo scopo fondamentale del programma consiste nell'impiegare in maniera efficace i dati esaustivi e dettagliati raccolti in merito agli incendi che hanno colpito la regione della Sardegna nel periodo che va dal 2010 al 2019. Questo permetterà di condurre simulazioni altamente realistiche riguardo alla propagazione delle fiamme in un ambiente simulato, specificamente concepito come una zona boschiva virtuale. Questa simulazione avverrà in un contesto controllato, dove sarà possibile manipolare liberamente le principali condizioni climatiche, offrendo all'utente un'ampia gamma di variabili da considerare e modificare a proprio piacimento.

Attraverso l'utilizzo di questi dati storici, il programma ambisce a fornire una piattaforma di analisi avanzata che consenta di esplorare in modo approfondito gli effetti di diversi fattori ambientali sulla diffusione degli incendi forestali. L'obiettivo è quello di creare uno strumento d'analisi completo e versatile, che permetta alle autorità competenti di pianificare interventi preventivi e di gestione del territorio basati su dati concreti e informazioni precise.

In particolare, la capacità di manipolare le variabili climatiche all'interno della simulazione offre un'opportunità unica per esaminare gli impatti delle condizioni meteorologiche sulla diffusione delle fiamme e sulla velocità con cui un incendio può propagarsi in un ambiente boschivo. Ciò consente di valutare in modo dettagliato i rischi associati a diverse condizioni atmosferiche e di sviluppare strategie di gestione del territorio e di prevenzione degli incendi forestali più efficaci ed efficienti.

In definitiva, il programma mira a diventare uno strumento essenziale per le autorità e gli esperti del settore, offrendo loro la possibilità di analizzare e comprendere in modo approfondito la dinamica degli incendi boschivi e di adottare misure preventive e di gestione del territorio mirate a proteggere le risorse naturali e la sicurezza delle comunità locali.

2. Le Maggiori Sfide Affrontate

Durante la preparazione del progetto sono state rilevate alcune difficoltà poste principalmente dalla limitatezza dei dati a disposizione e dalla conoscenza di librerie specifiche di Java. Alcune di queste sfide hanno imposto un cambio di rotta nel modo di affrontare il problema, altre invece sono state risolte senza incidere sul risultato finale.

Il più grande inconveniente è stato causato da una mancanza di dati significativi sulla densità di vegetazione, che durante il periodo considerato non ha subito una variazione tale da poter essere presa in considerazione tra i parametri della regressione lineare. Di conseguenza, è stato deciso di togliere la possibilità all'utente di modificarla a proprio piacimento, fissandola invece ad un valore pari al valor medio delle rilevazioni contenute nel database.

Questo cambiamento ha comportato un ripensamento quasi totale del modo in cui la simulazione dovesse essere gestita, passando da un sistema in cui l'estensione dell'incendio venisse determinata da una combinazione tra i dati della regressione lineare e la densità di vegetazione, ad uno nella quale il fattore discriminante è la distanza dal punto di origine dell'incendio rispetto ad ogni altro punto (calcolato utilizzando il metodo della distanza euclidea). Al fine di migliorare le prestazioni della simulazione, la probabilità che un pixel potesse incendiarsi è stata quindi

determinata dalla probabilità che la distanza tra di esso e il punto di inizio dell'incendio fosse maggiore o uguale al raggio dell'area incendiata. Per calcolare tale probabilità è stata utilizzata una distribuzione gaussiana avente come media e deviazione standard i dati ottenuti dalla regressione lineare applicata agli input dell'utente.

La regressione lineare stessa ha rappresentato in sé una sfida, che ha comportato lo studio di una libreria esterna "Jama" studiata per l'utilizzo di matrici e la gestione di operazioni matriciali. La regressione lineare multipla è una tecnica statistica utilizzata per studiare le relazioni tra una variabile dipendente e due o più variabili indipendenti. Nel contesto del metodo della scomposizione QR, il processo di regressione viene scomposto in due fasi: la decomposizione della matrice dei regressori in una matrice ortogonale Q e una matrice triangolare superiore R, e la soluzione del sistema di equazioni lineari utilizzando questa decomposizione.

Nel dettaglio, la matrice dei regressori X viene decomposta come $X = QR$, dove Q è una matrice ortogonale (cioè $Q * Q^T = I$, dove I è la matrice identità) e R è una matrice triangolare superiore. Questa decomposizione semplifica il calcolo dei coefficienti di regressione e rende il processo più stabile numericamente, specialmente quando le variabili indipendenti sono fortemente correlate. Una volta ottenuta la decomposizione QR, è possibile risolvere il sistema di equazioni lineari tramite la sostituzione all'indietro, utilizzando la matrice R e il vettore dei risultati Y per calcolare i coefficienti di regressione.

In sintesi, il metodo della scomposizione QR offre un'alternativa efficiente e stabile per calcolare i coefficienti di regressione nella regressione lineare multipla, garantendo una migliore gestione della co-linearità tra le variabili indipendenti e una maggiore precisione nei risultati.

Nel contesto della gestione dell'interfaccia grafica, mi sono trovato di fronte alla necessità di gestire operazioni lunghe in background senza compromettere la reattività dell'interfaccia utente (UI). Per affrontare questa sfida, ho utilizzato le classi `Task` e `Thread`, che offrono strumenti efficaci per eseguire operazioni asincrone e multithreading in Java. La classe `Task` è un componente fondamentale di JavaFX, progettata per gestire operazioni che richiedono molto tempo senza bloccare l'interfaccia utente. Essa fornisce un'astrazione per eseguire operazioni lunghe in background e per aggiornare l'UI in modo sicuro quando il lavoro è completato. Nella applicazione, è stato utilizzato `Task` per eseguire l'elaborazione delle immagini in modo asincrono rispetto al thread dell'interfaccia utente. Il metodo `call()` all'interno di `Task` è stato implementato per eseguire il lavoro effettivo, garantendo che il thread UI rimanga reattivo durante l'elaborazione delle immagini. La classe `Thread`, d'altra parte, è una componente di base di Java utilizzata per creare e gestire thread. Un `thread` rappresenta un singolo thread di esecuzione nel programma Java e consente di eseguire operazioni in parallelo. Ho utilizzato `Thread` per avviare un nuovo thread separato per eseguire l'istanza di `Task`. Questo approccio mi ha consentito di eseguire l'elaborazione delle immagini in background senza influenzare negativamente l'interfaccia utente. Il metodo `start()` di `Thread` è stato invocato per avviare l'esecuzione del thread, che ha eseguito il codice all'interno del metodo `call()` di `Task`.

Questo approccio ha migliorato l'esperienza utente complessiva, permettendo all'utilizzatore dell'applicazione di osservare il progressivo progredire della simulazione.

3. Descrizione del Data-Set

Questo dataset raccoglie dati sull'occorrenza di incendi nella regione della Sardegna, Italia. Il periodo temporale considerato va dal 2009 al 2019. Inoltre, sono fornite alcune variabili climatiche selezionate della regione, tra cui precipitazioni, velocità del vento e temperatura.

La Sardegna è la seconda isola più grande del Mar Mediterraneo e, dal XVIII secolo, fa parte dell'Italia. Tuttavia, la sua regione si è sviluppata indipendentemente dalla penisola italiana, dando origine a una cultura e una lingua distinte, influenzate anche dai loro vicini catalani nell'attuale Spagna. L'isola eredita il tipico clima mediterraneo: inverni miti, estati calde con la maggior parte delle precipitazioni che si verificano durante i mesi invernali. La sua posizione geografica è 40°2' N, 9°4' E e copre un'area di circa 270 km per 190 km.

La flora della Sardegna è dominata dalla macchia mediterranea, caratterizzata da arbusti sempreverdi, e è completata da foreste mediterranee e cactacee. Alcuni delle sue coltivazioni agricole più importanti sono gli ulivi e le querce da sughero. Come molte regioni del sud Europa, il nostro clima in rapido cambiamento sta aumentando la frequenza di temperature estive record e spostando il clima di queste regioni semi-aride verso un rischio maggiore di desertificazione. In generale, la macchia stessa è abbastanza adattata agli incendi, consentendo un recupero entro pochi anni dopo un incendio. Tuttavia, se gli incendi sono troppo frequenti, il recupero completo è meno probabile e può disturbare l'ecosistema per periodi più lunghi.

Ai fini della simulazione sono stati presi in esame solo gli attributi ritenuti rilevanti:

- Burned_area: l'area totale incendiata in m2.
- Standard_error: l'errore nella misurazione dell'area incendiata in m2.
- Number_of_patches: il numero di incendi rilevati.
- Fraction_of_burnable_area: la frazione di area infiammabile su quella totale.
- T2m: la temperatura media in °C a 2 metri dal suolo.
- Tp: la quantità di precipitazioni totali in mm.
- U10: componente della velocità del vento in m/s.
- V10: componente della velocità del vento in m/s.

4. Descrizione ad Alto Livello delle Strutture Dati e degli Algoritmi

1. Organizzazione Interna

L'applicazione è stata implementata in linguaggio Java, seguendo il design pattern MVC (Model-View-Controller) e il pattern DAO (Data-Access-Object) per separare le diverse parti del codice e migliorarne la gestione complessiva.

Pattern MVC (Model-View-Controller)

Il pattern MVC è un design pattern architetturale che separa un'applicazione in tre componenti principali: Model, View e Controller.

- **Model:** Questo componente rappresenta la logica dell'applicazione e i dati associati. Il Model gestisce la logica di business, l'accesso ai dati e qualsiasi elaborazione dei dati necessaria per l'applicazione. È responsabile della gestione dello stato dell'applicazione e delle operazioni sui dati, senza preoccuparsi dell'interfaccia utente o della visualizzazione dei dati.
- **View:** La View è responsabile della rappresentazione grafica dell'interfaccia utente e della visualizzazione dei dati provenienti dal Model. La View riceve informazioni dal Controller e li visualizza all'utente in modo appropriato. Tuttavia, la View non deve contenere alcuna logica di business o accesso ai dati direttamente.
- **Controller:** Il Controller funge da intermediario tra la View e il Model. Si occupa di gestire le interazioni dell'utente con l'interfaccia utente e di inviare i comandi appropriati al Model per l'elaborazione. Il Controller riceve gli input dall'utente attraverso la View, li interpreta e li traduce in azioni da eseguire sul Model.

Il pattern MVC favorisce la separazione delle responsabilità, consentendo una migliore organizzazione del codice e una maggiore modularità dell'applicazione. Ciò facilita la manutenzione del codice, la riusabilità delle componenti e la scalabilità dell'applicazione.

Pattern DAO (Data-Access-Object)

Il pattern DAO è un design pattern architetturale che separa la logica di accesso ai dati dal resto dell'applicazione.

Data Access Object (DAO): Il DAO è un'interfaccia o una classe astratta che fornisce metodi per accedere ai dati da una sorgente dati, come un database o un file. Il DAO definisce operazioni CRUD (Create, Read, Update, Delete) per l'accesso ai dati e nasconde i dettagli dell'implementazione del database dal resto dell'applicazione.

Il pattern DAO consente di disaccoppiare la logica di business e l'accesso ai dati, consentendo una maggiore flessibilità nell'implementazione dei cambiamenti nel database o nella sorgente dati. Inoltre, semplifica il testing dell'applicazione, in quanto è possibile sostituire facilmente l'implementazione del DAO con un mock durante i test unitari.

2. Algoritmi Principali

Costruzione della Mappa e del Grafo

Questa funzione, denominata `getMap`, genera e restituisce un'immagine rappresentante una mappa di una area fittizia avente una densità di vegetazione paragonabile a quella del territorio della Sardegna. Inizialmente, vengono definite le dimensioni dell'immagine, la dimensione dei quadrati che costituiranno la mappa e calcolati il numero di righe e colonne in base a queste dimensioni. Viene quindi creata un'istanza di `WritableImage` con le dimensioni specificate e un `PixelWriter` per scrivere i pixel sull'immagine.

Successivamente, viene inizializzato un'ArrayList bidimensionale `pixelMap` per tenere traccia dei pixel della mappa. Si procede quindi a iterare attraverso tutte le righe e colonne della mappa. Per ciascun pixel, viene generato casualmente un numero compreso tra 0 e 1 per rappresentare la densità della vegetazione in quella posizione. Se il valore generato è maggiore della densità della vegetazione desiderata, il pixel viene impostato come territorio non a rischio incendio(false) e colorato di bianco, altrimenti viene considerato vegetazione (true) e colorato di verde chiaro o scuro a seconda della posizione.

Durante il processo di iterazione, vengono creati oggetti `Pixel` per rappresentare ciascun pixel e vengono aggiunti alla lista `pixelList`, che rappresenta una riga di pixel della mappa. Questa lista viene quindi aggiunta all'ArrayList bidimensionale `pixelMap`.

Infine, dopo aver generato la mappa dei pixel, viene chiamata la funzione `buildGraph` per costruire un grafo basato sulla disposizione dei pixel sulla mappa. Infine, l'immagine viene restituita.

Questa funzione, denominata `buildGraph`, è responsabile della costruzione del grafo che rappresenta la disposizione dei pixel sulla mappa generata precedentemente.

Inizialmente, viene istanziato un grafo diretto di default utilizzando la libreria JGraphT. Successivamente, vengono aggiunti i vertici del grafo, rappresentati dai pixel vegetati della mappa. Questo viene fatto iterando attraverso ogni riga di pixel della mappa e aggiungendo i pixel vegetati come vertici del grafo.

Successivamente, vengono aggiunte le connessioni tra i vertici del grafo. Questo viene fatto esaminando ogni coppia di pixel vegetati adiacenti nella mappa. Se due pixel adiacenti sono entrambi vegetati, viene aggiunta un'arco diretto dal pixel corrente al pixel adiacente. Questo processo viene eseguito iterando attraverso ogni riga di pixel nella mappa, ad eccezione dell'ultima riga.

Infine, viene gestito il collegamento tra l'ultima riga di pixel della mappa, assicurandosi che i pixel vegetati siano connessi tra loro, se presenti.

In sintesi, questa funzione genera un grafo non direzionato che rappresenta la disposizione dei pixel vegetati sulla mappa, con archi che collegano i pixel adiacenti in modo da modellare la connettività spaziale tra di essi.

La Regressione Lineare Multipla

La creazione di un modello di regressione lineare multipla rappresenta un passo cruciale nell'analisi e nella previsione dell'estensione media di incendi boschivi. La classe `LinearRegression` fornisce un'implementazione robusta di questo modello, che si basa sulla decomposizione QR per trovare i coefficienti della regressione. Questo metodo offre una soluzione efficiente e stabile per la stima dei parametri del modello.

Il metodo `findCoefficients` svolge un ruolo centrale nella classe `LinearRegression`. Esso è responsabile della risoluzione della regressione lineare multipla, prendendo in input una matrice di variabili indipendenti e un array di risultati. Il processo di risoluzione coinvolge diverse fasi:

- **Costruzione delle Matrici:** Il metodo inizia costruendo le matrici delle variabili indipendenti e dei risultati, utilizzando i dati forniti come input. Questo passaggio è fondamentale per preparare i dati per l'analisi e la successiva decomposizione QR.
- **Decomposizione QR:** Una volta costruite le matrici, il metodo utilizza la decomposizione QR per trovare la soluzione dei minimi quadrati. Questo approccio è preferibile in quanto offre una maggiore stabilità numerica rispetto ad altri metodi di risoluzione.
- **Calcolo dei Coefficienti:** Utilizzando i risultati della decomposizione QR, il metodo calcola i coefficienti della regressione, che rappresentano i pesi associati alle variabili indipendenti nel modello lineare.
- **Restituzione dei Coefficienti:** Infine, il metodo restituisce la matrice dei coefficienti calcolati, pronta per essere utilizzata per effettuare previsioni su nuovi dati.

Il metodo `buildVariables` svolge una funzione ausiliaria cruciale per la preparazione dei dati. Esso converte un array bidimensionale di dati in una matrice, che rappresenta le variabili indipendenti del modello. Questo processo è essenziale per organizzare correttamente i dati e prepararli per l'analisi tramite il metodo `findCoefficients`.

Analogamente al metodo `buildVariables`, il metodo `buildResults` si occupa di convertire un array unidimensionale di dati in una matrice, che rappresenta i risultati della regressione. Questo passaggio è necessario per strutturare correttamente i dati di output e prepararli per l'analisi tramite il metodo `findCoefficients`.

In sintesi, la classe `LinearRegression` fornisce un'implementazione completa e efficiente della regressione lineare multipla, utilizzando la decomposizione QR per trovare i coefficienti del modello. I metodi ausiliari `buildVariables` e `buildResults` giocano un ruolo fondamentale nella preparazione e organizzazione dei dati, consentendo una corretta analisi e previsione dei risultati.

Il programma sfrutta questa classe per ottenere una previsione dell'estensione media di un singolo incendio e il relativo errore standard combinando i coefficienti ottenuti dai dati del db con i dati forniti in input dall'utente (temperatura, velocità del vento e precipitazioni).

Simulazione dell'Incendio

La funzione `spreadFire()` costituisce il nucleo della simulazione degli incendi boschivi, predisponendo le variabili necessarie e avviando il processo di diffusione del fuoco all'interno della scena simulata. Questa funzione si articola nei seguenti passaggi chiave:

- **Inizializzazione delle Variabili:** La funzione inizia inizializzando due variabili cruciali per la simulazione. La prima variabile tiene traccia dell'area bruciata durante la simulazione, mentre la seconda consiste in una coda di priorità, che sarà utilizzata per scandire ogni passaggio della simulazione.
- **Distribuzione Normale dell'Area Incendiata:** Successivamente, viene inizializzata una distribuzione normale che tiene conto dell'area incendiata totale, considerando la densità di vegetazione. Questo passaggio è essenziale per simulare la diffusione realistica del fuoco all'interno dell'ambiente simulato.
- **Selezione Casuale del Pixel di Partenza:** La funzione seleziona casualmente un pixel di partenza all'interno della scena simulata. Questo pixel sarà il punto di partenza per la diffusione del fuoco all'interno dell'ambiente.
- **Calcolo delle Distanze Euclidee e Inserimento nella Coda di Priorità:** Vengono quindi calcolate le distanze euclidee da tutti i nodi rispetto al nodo di partenza e queste distanze vengono inserite nella coda di priorità. Questo passaggio è fondamentale per determinare l'ordine di propagazione del fuoco all'interno della scena.

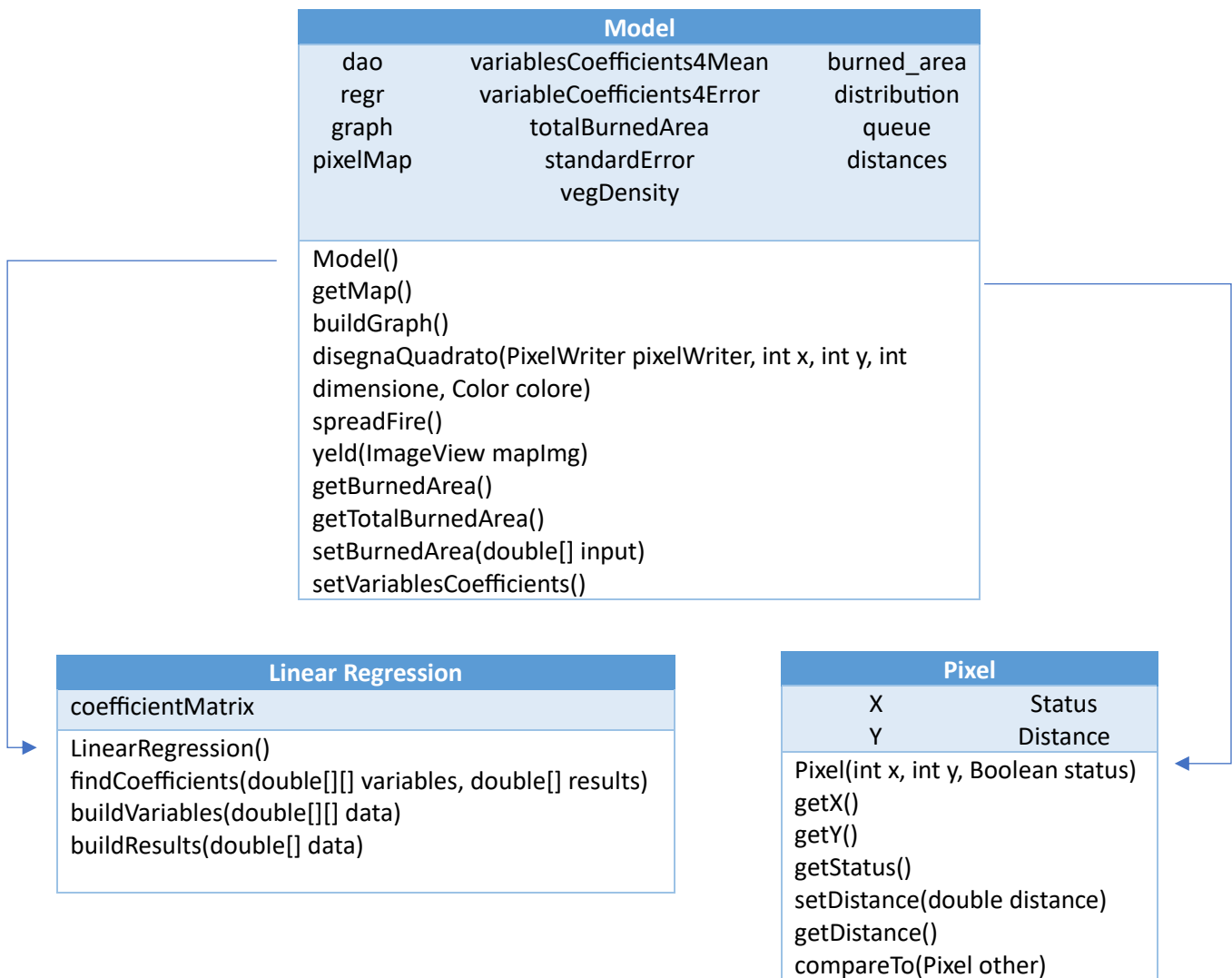
La funzione `yield(ImageView mapImg)` rappresenta il cuore pulsante della simulazione degli incendi boschivi, poiché modifica l'immagine pixel per pixel fino a quando non viene restituito null, indicando la fine della simulazione. Questa funzione si compone dei seguenti passaggi:

- **Ottenimento dell'Immagine e Conversione:** La funzione ottiene l'immagine da un oggetto `ImageView`, la converte in un'immagine scrivibile e ottiene un oggetto `PixelWriter` per poter modificare i pixel dell'immagine.
- **Scansione della Coda e Calcolo della Probabilità di Bruciatura:** Se la coda di priorità non è vuota, la funzione estrae un pixel dalla coda e calcola una probabilità di bruciatura basata sulla distribuzione normale. Questa probabilità rappresenta la probabilità che l'area dell'incendio partendo dal pixel iniziale coinvolga il pixel corrente.
- **Aggiornamento dell'Area Bruciata e Cambio del Colore del Pixel:** Se la probabilità soddisfa una condizione casuale, la funzione aggiorna l'area bruciata e cambia il colore del pixel corrispondente nell'immagine, simboleggiando così la diffusione del fuoco.
- **Restituzione dell'Immagine Modificata:** La funzione restituisce l'immagine modificata.
- **Fine della Simulazione:** Se la coda di priorità è vuota, la funzione restituisce null, indicando al controller che la simulazione è terminata.

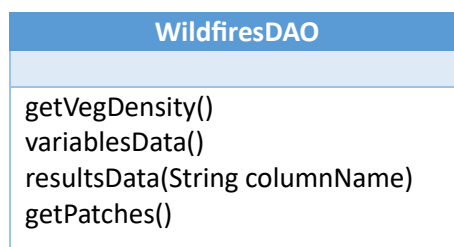
In sintesi, le funzioni `spreadFire()` e `yield()` collaborano sinergicamente per simulare la diffusione realistica degli incendi boschivi all'interno della scena simulata, permettendo di valutare e prevenire in modo efficace il propagarsi del fuoco.

5. Diagramma delle Classi Principali

1. Package "Model"



2. Package "Db"



3. Package "Controller"

FXMLControllerOne		
model	location	temperatureInput
resources	rainInput	windImg
rainImg	startButton	windInput
	templmg	
startSim() mostraAlertErrore(String titolo, String messaggio)		

FXMLControllerTwo	
model	fireButton
backImageView	quitbutton
resources	mapImg
location	text
goBack(ActionEvent event) startFire(ActionEvent event) updateImageView(WritableImage updatedImage) copyImage(Image originalImage)	

6. L'Applicazione

L'applicazione si articola in due finestre. Nella prima (Figura 1) viene mostrato il titolo del programma, le istruzioni per l'uso e sono presenti i tre campi di input in cui l'utente potrà inserire i parametri climatici sui quali calcolare la simulazione. Infine è presente un bottone che se premuto rimanda alla seconda pagina, nella quale potrà essere visualizzata la simulazione.

Nel caso in cui uno o più campi di input dovessero essere vuoti o invalidi nel momento in cui il bottone viene premuto, verrà visualizzato un pop-up che segnalerà l'errore e il programma non proseguirà alla pagina successiva fino alla risoluzione del problema.



The screenshot shows a window titled "Simulazione Incendio". The main heading is "Simulazione della Propagazione di un Incendio". Below it, instructions state: "ISTRUZIONI PER L'USO: inserire nei campi sottostanti i valori desiderati per ciascun fattore climatico. Si tenga conto che per valori estremi (outliers) il programma potrebbe produrre un output paradossale o privo di significato." There are three input fields: "Temperatura" with a thermometer icon and a unit of °C, "Velocità del Vento" with a wind icon and a unit of m/s, and "Precipitazioni" with a cloud and rain icon and a unit of mm. At the bottom is a button labeled "Avvia Simulazione".

Figura 1: Pagina Iniziale

Nella seconda finestra (Figura 2) viene mostrata graficamente la mappa l'area fittizia creata dal programma, nella quale i pixel di colore verde rappresentano una porzione di area coperta da vegetazione e a rischio incendio. In questa schermata sono presenti due pulsanti: a sinistra "Torna Indietro" permette di resettare tutti i parametri della simulazione e di reinserire i dati, visualizzando nuovamente la pagina iniziale, mentre a destra "Mostra Incendio" avvia la simulazione.

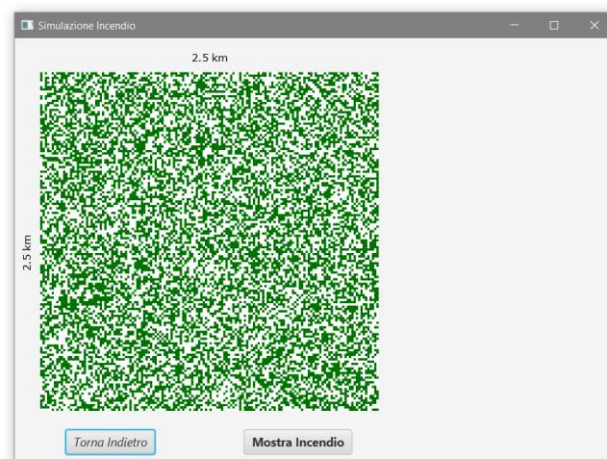


Figura 2: Seconda Pagina

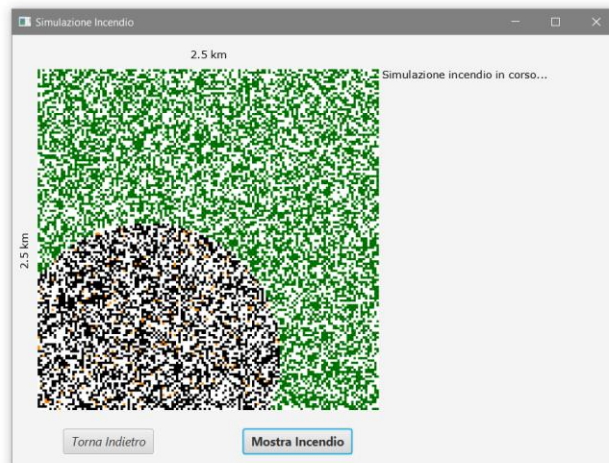


Figura 3: Pagina 2 durante la simulazione

Durante la simulazione l'utente potrà progressivamente osservare il propagarsi dell'incendio e apparirà una scritta che informerà l'utente dell'avvio della simulazione (Figura 3). Tale scritta cambierà soltanto a simulazione ultimata.

Infine, a simulazione ultimata, appariranno delle informazioni riguardanti l'area totale a rischio incendio, l'area effettivamente interessata dalla simulazione ed infine l'area teorica predetta dalla regressione lineare (Figura 4).

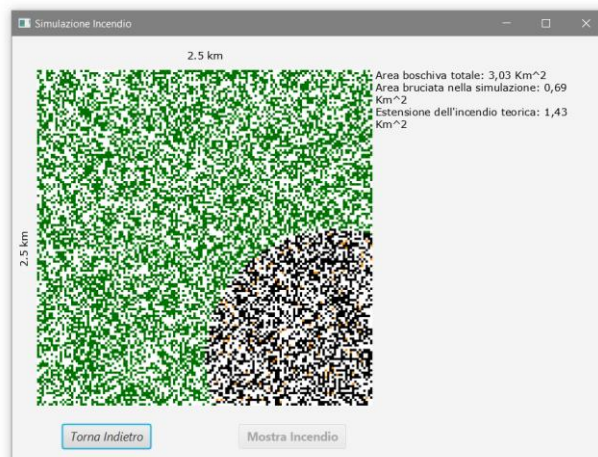


Figura 4: Pagina 2 a simulazione ultimata

[Qui](#) può essere visto un video youtube che spiega il funzionamento dell'applicazione.

7. Risultati Sperimentali

Di seguito vengono proposti i risultati di tre esperimenti effettuati in tre scenari molto diversi, in ordine di gravità decrescente.

Test 1

Simulazione della Propagazione di un Incendio

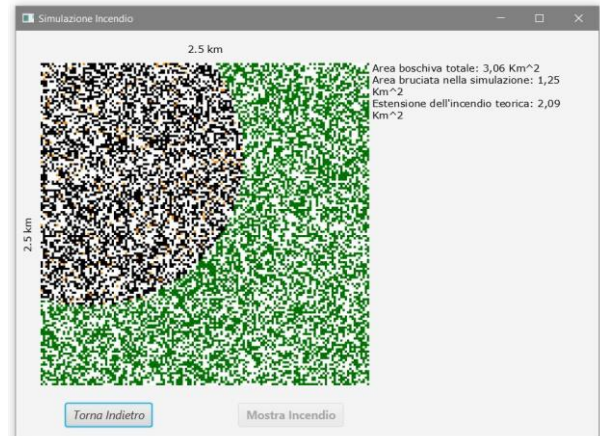
ISTRUZIONI PER L'USO: inserire nei campi sottostanti i valori desiderati per ciascun fattore climatico. Si tenga conto che per valori estremi (outliers) il programma potrebbe produrre un output paradossale o privo di significato.

Temperatura
40 °C

Velocità del Vento
2 m/s

Precipitazioni
5 mm

[Avvia Simulazione](#)



Test 2

Simulazione della Propagazione di un Incendio

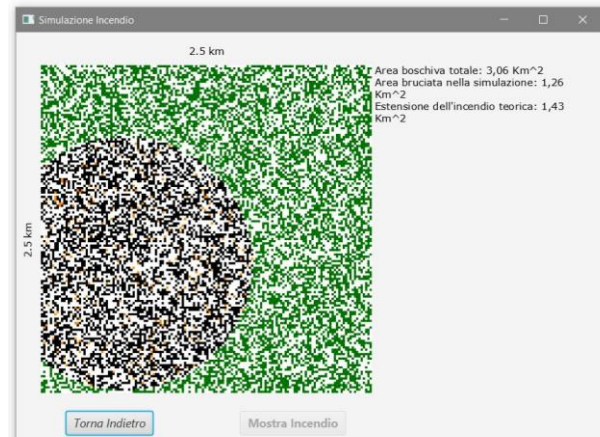
ISTRUZIONI PER L'USO: inserire nei campi sottostanti i valori desiderati per ciascun fattore climatico. Si tenga conto che per valori estremi (outliers) il programma potrebbe produrre un output paradossale o privo di significato.

Temperatura
30 °C

Velocità del Vento
1 m/s

Precipitazioni
10 mm

[Avvia Simulazione](#)



Test 3

Simulazione della Propagazione di un Incendio

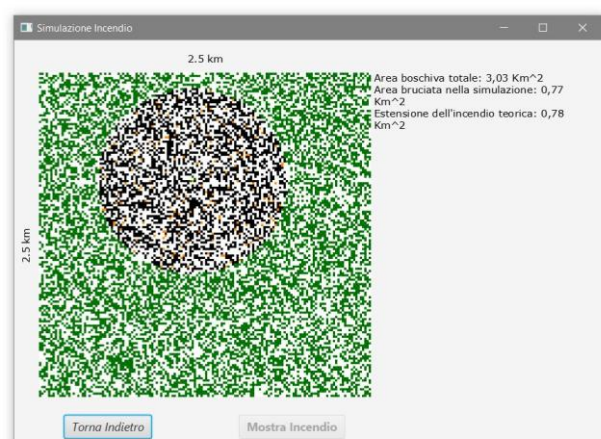
ISTRUZIONI PER L'USO: inserire nei campi sottostanti i valori desiderati per ciascun fattore climatico. Si tenga conto che per valori estremi (outliers) il programma potrebbe produrre un output paradossale o privo di significato.

Temperatura
20 °C

Velocità del Vento
1 m/s

Precipitazioni
15 mm

[Avvia Simulazione](#)



8. Valutazioni e Conclusioni

I risultati mostrati nel capitolo precedente permettono di trarre alcune conclusioni. Innanzitutto le previsioni teoriche del programma si dimostrano essere in accordo con le attese e fortemente dipendenti dagli input inseriti. E in condizioni ottimali i risultati della simulazione si avvicinano molto a quelli teorici, con scarti dell'ordine del centesimo. Tuttavia in condizioni particolarmente favorevoli alla propagazione dell'incendio, nelle quali la probabilità che l'area delimitata della simulazione non sia sufficientemente ampia da essere in grado di contenere tutto l'incendio, specialmente se il punto di partenza scelto casualmente dal programma è in prossimità dei bordi, i risultati ottenuti discostano molto da quelli teorici.

Inoltre, nonostante i risultati promettenti, il modello presenta alcune limitazioni che potrebbero essere oggetto di futuri sviluppi. Ad esempio, potrebbe essere interessante considerare l'inclusione di ulteriori variabili predittive o esplorare l'utilizzo di tecniche di regressione più avanzate per migliorare le prestazioni del modello.

Un altro fattore da considerare è la limitata variabilità dei dati presenti nel database, per cui l'attendibilità del programma diminuisce per valori di input per una o più variabili considerabili come "outliers", cioè al di fuori dell'intervallo tra i massimi e i minimi.

Nel complesso, i tempi impiegati dal programma si sono rivelati più che soddisfacenti, con tempi di attesa quasi nulli. In particolare, per quanto riguarda la simulazione della propagazione delle fiamme si è rivelato estremamente efficace interrompere l'algoritmo in anticipo quando la probabilità di incendio scende al di sotto di valori molto bassi (nell'ordine di 10^{-5}).

In conclusione, l'implementazione della regressione lineare multipla ha fornito un modello efficace per la previsione dell'estensione media degli incendi boschivi in base ai dati a disposizione e l'algoritmo di simulazione è in grado di ottenere ottimi risultati nelle giuste condizioni. L'analisi dei risultati ha evidenziato la relazione tra le variabili analizzate e l'estensione degli incendi, fornendo informazioni preziose per la gestione e la prevenzione degli incendi boschivi.

Nota sulle Fonti e le Immagini

L'idea di realizzare questa tesi nasce dalla lettura del testo "*Network Science*" di Albert-László Barabási, disponibile gratuitamente in formato web [qui](#). Il suo completamento non sarebbe stato possibile senza la [Multiple Linear Regression](#) di Robert Sedgewick e Kevin Wayne (Princeton University). Le principali librerie esterne utilizzate sono state: [JavaFX](#), [SQLite](#), [JgraphT](#) e [Jama](#).

Il database e le immagini utilizzate sono tutti rilasciati sotto licenza Creative Commons.



Quest'opera è distribuita con Licenza [Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale](https://creativecommons.org/licenses/by-nc-sa/4.0/)