

CASO DI STUDIO MAP 2021-2022

K-Nearest Neighbors (**KNN**)

Progetto svolto da:

Liantonio Dario

Astarita Michele

Lopedota Francesco

Corso A

Consegna in data:

18/11/2022

SOMMARIO

1. Descrizione del progetto	3
2. Spiegazione dell'algoritmo	3
3. Funzionalità implementate con estensione	4
4. Guida di installazione JavaFx	5
5. Diagramma delle classi.	12
6. Guida di installazione BAT.	14
7. Guida di installazione JAR.	16
8. Guida utente con esempi di test.	17

1. Descrizione del progetto:

Il progetto realizzato, KNN, è compreso nel campo del Data Mining. Lo scopo del Data Mining è l'estrazione (semi) automatica di conoscenza nascosta in voluminose basi di dati al fine di renderla disponibile e direttamente utilizzabile.

L'area di applicazione per cui è stato svolto il progetto è la regressione, ovvero la predizione del valore di un attributo numerico associato a un esempio sulla base di valori osservati per altri attributi dell'esempio medesimo. Gli esempi usati sono descritti come vettori di coppie attributo-valore per i quali è noto l'attributo classe (target), il quale nella regressione è numerico.

2. Spiegazione dell'algoritmo:

Dato:

- un training set (X,Y) ;
- un esempio x denominato query per il quale il valore y sia sconosciuto;
- un intero $k > 0$;

Predice il valore sconosciuto di y associato ad x identificando i k esempi del training set più vicini ad x e restituendo la media dei valori y nei k vicini selezionati.

3. Funzionalità implementate con estensione:

Con l'implementazione dell'estensione, tutte le funzionalità presenti nella Sorgente di Sistema a linea di comando vengono riproposte attraverso un'**interfaccia grafica**. L'interfaccia grafica è stata realizzata attraverso l'utilizzo delle librerie **JavaFX**. Le varie scene sono state modellate attraverso dei file con estensione .fxml realizzati attraverso un software chiamato **SceneBuilder**.



Lato **Client**:

- abbiamo modificato la classe principale utilizzando l'ereditarietà, rendendola una sottoclasse che estende la superclasse **Application**.
- La connessione tra client e server viene gestita da un nuovo metodo **Connection()**, che ne permette l'iterazione, non più a linea di comando bensì attraverso l'interfaccia.
- Sono stati inseriti diversi metodi che permettono la **scalatura** delle varie scene eventualmente visualizzate anche a schermo intero.
- Sono state implementate diverse classi **controller** che gestiscono le esigenze delle relative scene e che le collegano l'una con l'altra.
- **scena 1**: breve introduzione alle funzionalità dell'applicazione
- **scena 2**: scelta origine del file per il caricamento del TrainingSet
- **scena 3**: inserimento del nome della tabella da file di testo
- **scena 4**: inserimento dell'esempio e del valore k
- **scena 5**: inserimento del nome della tabella da file binario
- **scena 6**: inserimento del nome della tabella da database
- **scena 7**: visualizzazione della predizione e possibilità di effettuarne una nuova

4. Guida di installazione JavaFx su VSC:

Prima dell'installazione, ci sono due requisiti necessari da rispettare, ovvero:

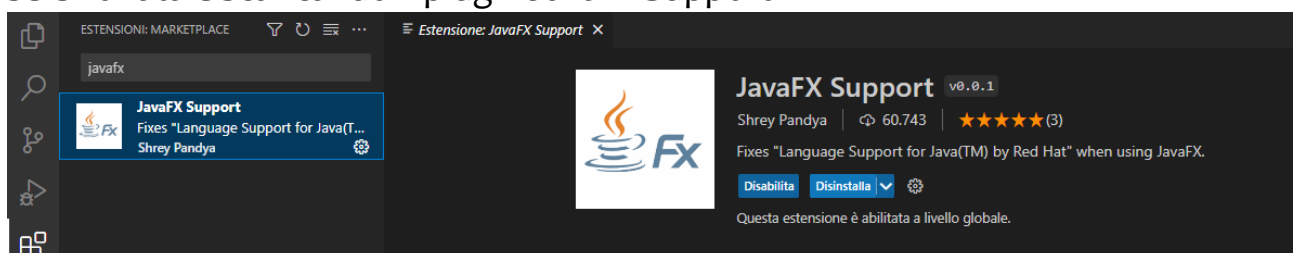
- Accertarsi che VSC sia aggiornato almeno alla versione 1.49.3, o superiori; ci si può accertare di ciò andando nel menù di guida e cliccando il tasto Informazioni su, che vi dirà la versione corrente di VSC;



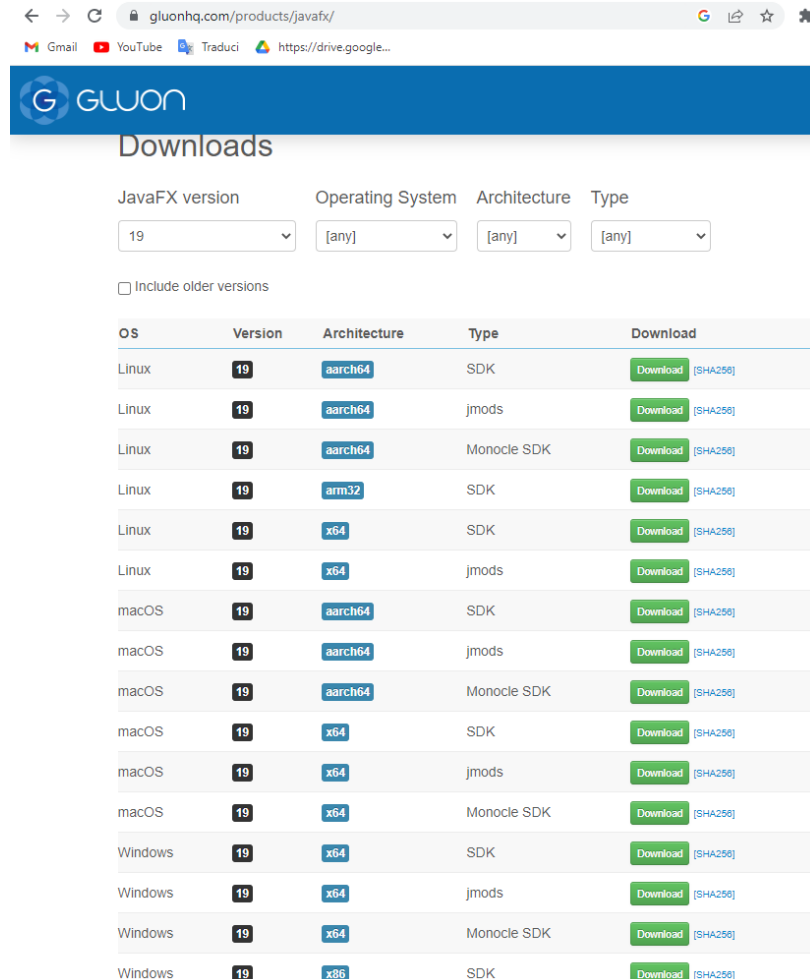
- Accertarsi di aver scaricato ed installato il JDK almeno la versione 11.

Dopo aver rispettato i requisiti, i passi per l'installazione di JavaFx sono i seguenti:

- Installare l'estensione JavaFx in Visual Studio Code, cliccando sull'icona selezionata e scaricando il plugin JavaFx Support:



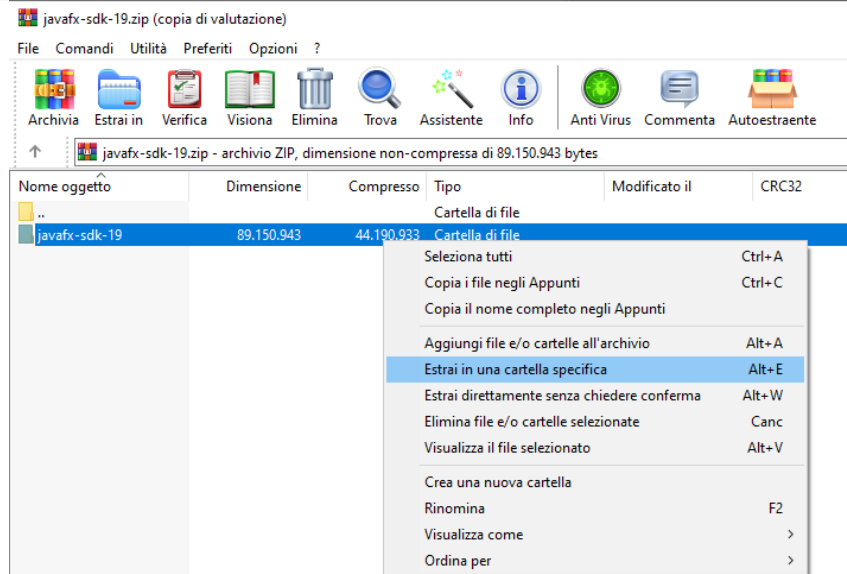
- Andare sul sito <https://gluonhq.com/products/javafx/> e scaricare la versione di JavaFx adatta al proprio sistema operativo, nel nostro caso Windows con architettura x64:



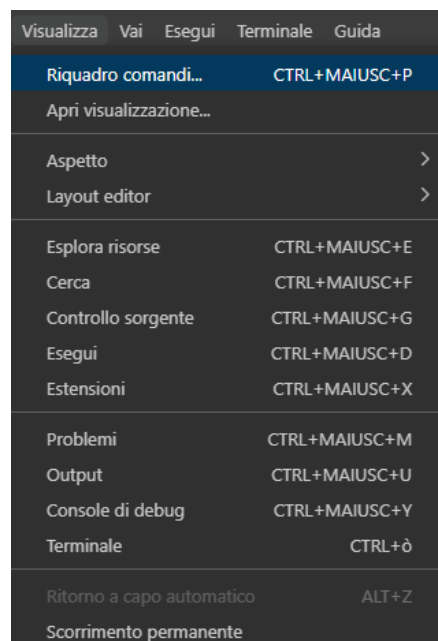
The screenshot shows the Gluon JavaFX Downloads page. At the top, there's a blue header with the Gluon logo. Below it, the word "Downloads" is centered. There are four dropdown menus for filtering: "JavaFX version" (set to 19), "Operating System" (set to [any]), "Architecture" (set to [any]), and "Type" (set to [any]). Below these is a checkbox for "Include older versions". A table lists the available downloads, with columns for OS, Version, Architecture, Type, and Download. The table shows 19 versions of JavaFX for various operating systems and architectures. The last row, which is highlighted, shows Windows, version 19, x64 architecture, SDK type, and a download link.

OS	Version	Architecture	Type	Download
Linux	19	aarch64	SDK	Download [SHA256]
Linux	19	aarch64	jmods	Download [SHA256]
Linux	19	aarch64	Monocle SDK	Download [SHA256]
Linux	19	arm32	SDK	Download [SHA256]
Linux	19	x64	SDK	Download [SHA256]
Linux	19	x64	jmods	Download [SHA256]
macOS	19	aarch64	SDK	Download [SHA256]
macOS	19	aarch64	jmods	Download [SHA256]
macOS	19	aarch64	Monocle SDK	Download [SHA256]
macOS	19	x64	SDK	Download [SHA256]
macOS	19	x64	jmods	Download [SHA256]
macOS	19	x64	Monocle SDK	Download [SHA256]
Windows	19	x64	SDK	Download [SHA256]
Windows	19	x64	jmods	Download [SHA256]
Windows	19	x64	Monocle SDK	Download [SHA256]
Windows	19	x86	SDK	Download [SHA256]

- Una volta scaricato, estrarre dove si preferisce la cartella contenuta nel file .zip:



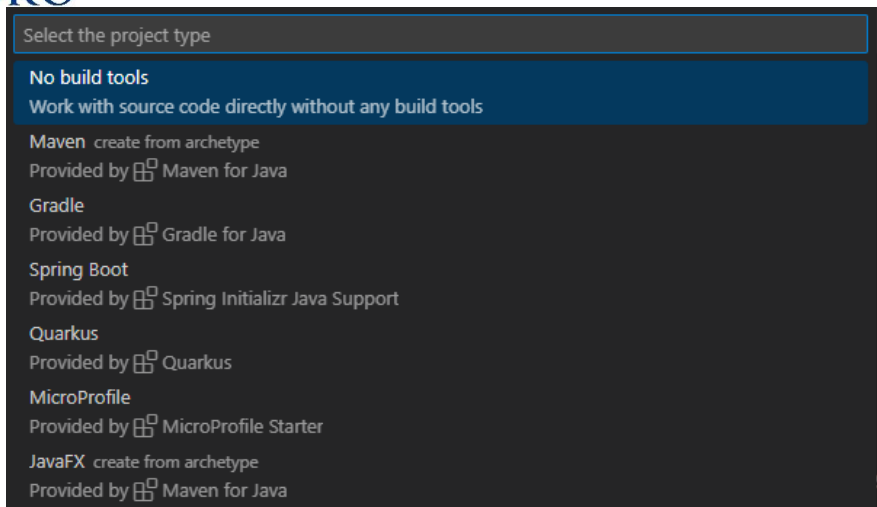
- Dopodichè, tornare in VSC e creare un progetto Java, andare sul riquadro comandi:



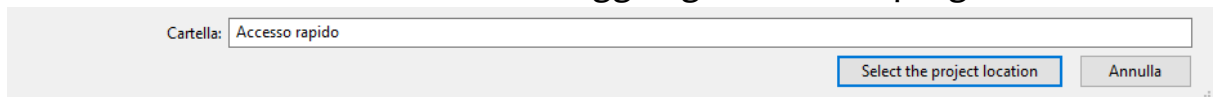
- Creare un nuovo progetto Java:



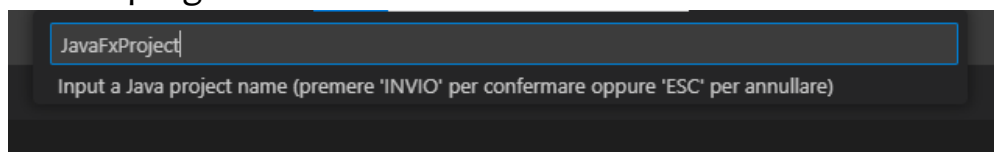
- Cliccare su No Build Tools:



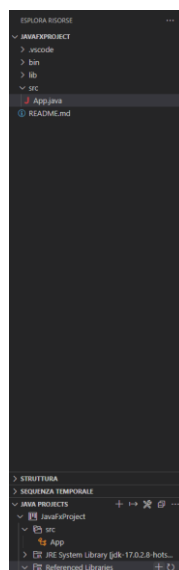
- Selezionare la cartella in cui si vuole aggiungere il nuovo progetto:



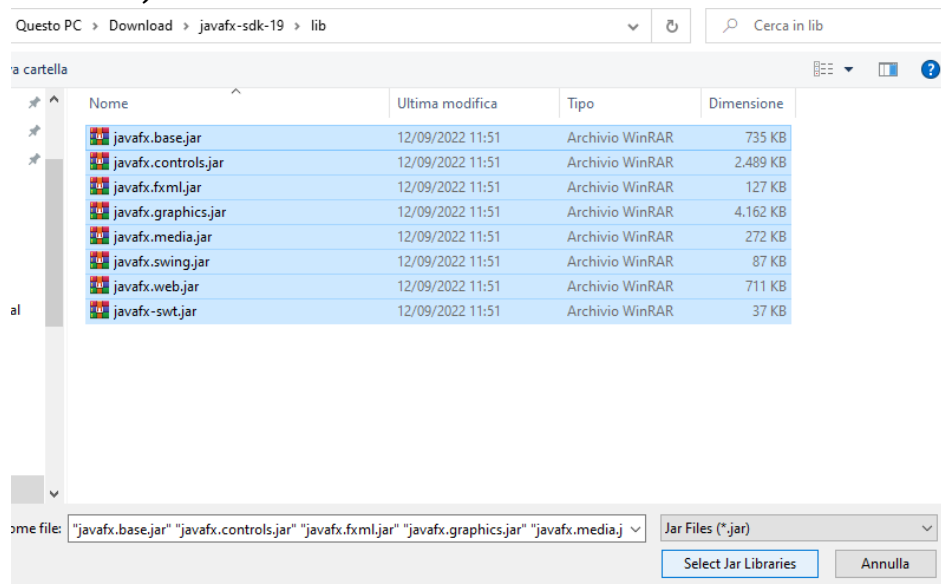
- In seguito, dare un nome al progetto che si sta creando e premere invio per confermare il progetto in VSC:



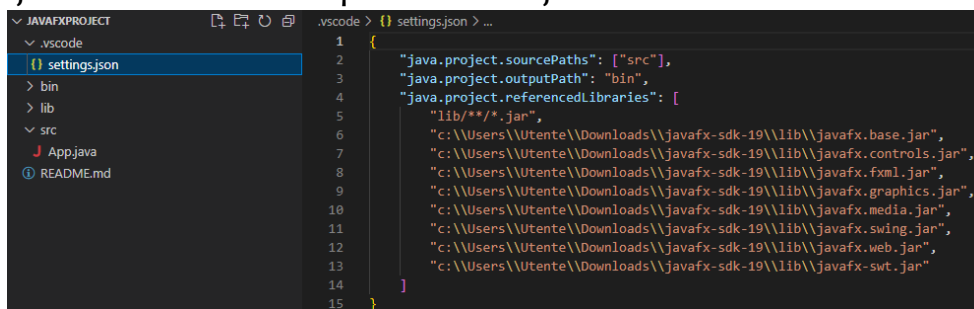
- Andare in fondo alla pagina sulla voce “Referenced Libraries” e cliccare sul tasto +:



- Si aprirà una finestra in cui bisognerà andare a selezionare la cartella precedentemente salvata, ed entrare nella sottodirectory lì presente nominata 'lib', e selezionare tutti i file presenti (lo si può fare con il comando Ctrl+A) e cliccare Select Jar Libraries:



- Così facendo, tutte le librerie di riferimento necessarie per creare un progetto JavaFx saranno aggiunte. Dopo questa operazione, nella sottodirectory del progetto ".vscode" sarà apparso un file con estensione .json con all'interno i path dei file .jar a cui si fa riferimento:



- Per proseguire, copiare da qualsiasi parte un qualsiasi codice con la sintassi JavaFx ed incollarlo nella classe App.java:

```
import javafx.application.Application;
import javafx.beans.property.IntegerProperty;
import javafx.beans.property.SimpleIntegerProperty;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.HBox;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class App extends Application {

    Stage window;
    Button button;

    Run | Debug
    public static void main(String[] args){
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        window = primaryStage;
        window.setTitle("thenewbutton");

        IntegerProperty x = new SimpleIntegerProperty(3);
        IntegerProperty y = new SimpleIntegerProperty();
    }
}
```

- Cliccando il tasto Run posto sopra alla dichiarazione del main, si avrà un errore di compilazione, che non permetterà al programma di essere eseguito:

```
Run | Debug
20     public static void main(String[] args){
21         launch(args);
22     }
23
24     @Override
25     public void start(Stage primaryStage) throws Exception {
26         window = primaryStage;
27         window.setTitle("thenewbutton");
28
29         IntegerProperty x = new SimpleIntegerProperty(3);
30         IntegerProperty y = new SimpleIntegerProperty();
31         //con il bind il valore di y resta legato a quello di x sempre nel programma,
32         y.bind(x.multiply(10));
    }
}
```

PROBLEMI OUTPUT CONSOLE DI DEBUG TERMINALE

Windows PowerShell
Copyright (C) Microsoft Corporation. Tutti i diritti riservati.

Prova la nuova PowerShell multiplatforma <https://aka.ms/pscore6>

PS C:\Users\Utente\Desktop\JavaFxProject> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe' -jar 'C:\Users\Utente\Desktop\JavaFxProject\build\classes\Main.class' 'App'

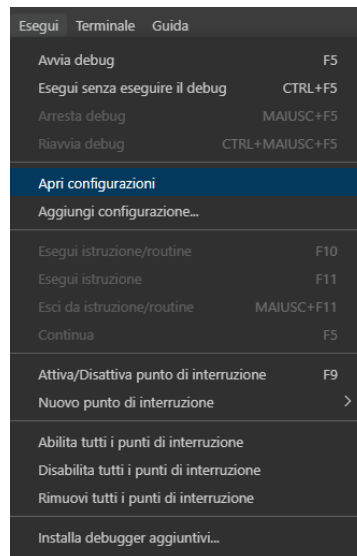
Hello, World!

PS C:\Users\Utente\Desktop\JavaFxProject> c:\> cd 'c:\Users\Utente\Desktop\JavaFxProject'; & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe' -jar 'C:\Users\Utente\Desktop\JavaFxProject\build\classes\Main.class' 'App'

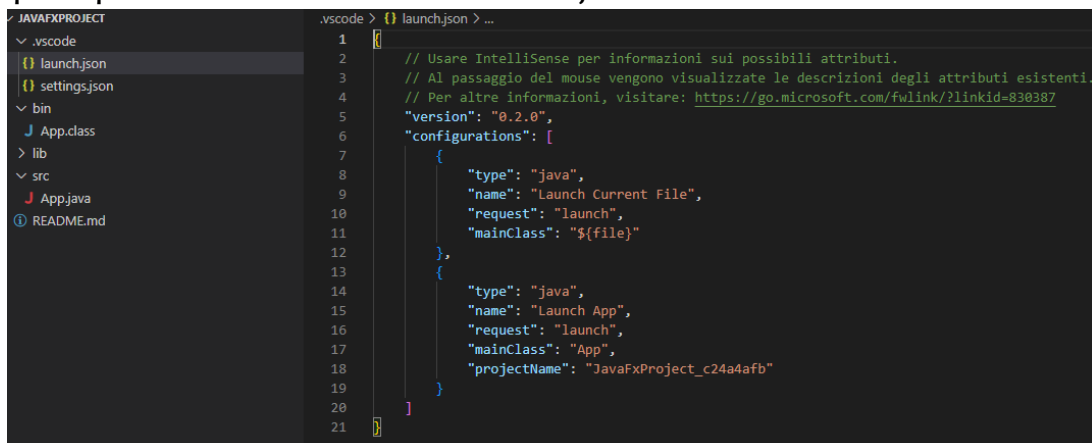
Errore: non sono presenti i componenti runtime di JavaFX necessari per eseguire questa applicazione

PS C:\Users\Utente\Desktop\JavaFxProject>

- Dunque cliccare su Esegui e successivamente su Apri Configurazioni:



- Si aprirà presto un file chiamato launch.json:



```

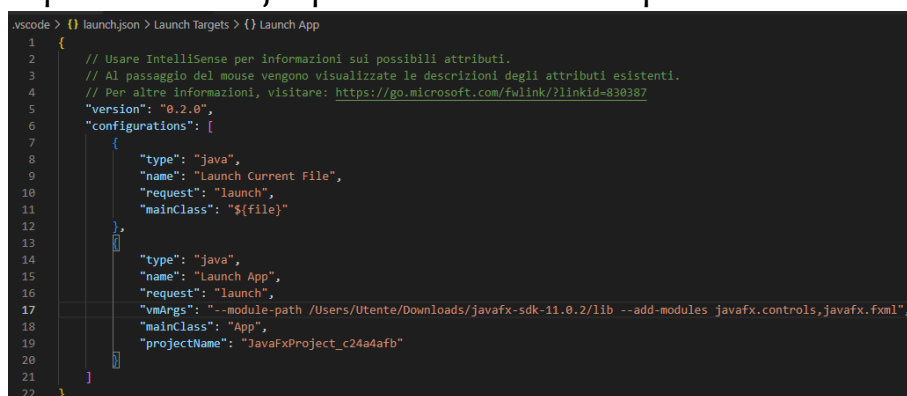
1  {
2    // Usare IntelliSense per informazioni sui possibili attributi.
3    // Al passaggio del mouse vengono visualizzate le descrizioni degli attributi esistenti.
4    // Per altre informazioni, visitare: https://go.microsoft.com/fwlink/?linkid=830387
5    "version": "0.2.0",
6    "configurations": [
7      {
8        "type": "java",
9        "name": "Launch Current File",
10       "request": "launch",
11       "mainClass": "${file}"
12     },
13     {
14       "type": "java",
15       "name": "Launch App",
16       "request": "launch",
17       "mainClass": "App",
18       "projectName": "JavaFxProject_c24a4afb"
19     }
20   ]
21 }

```

- Inserire questo testo

```
"vmArgs": "--module-path /Users/user/Downloads/javafx-sdk-11.0.2/lib --add-modules javafx.controls,javafx.fxml",
```

all'interno del file .json, impostando il path corretto in cui è salvato il file lib in cui sono presenti i file .jar precedentemente importati:



```

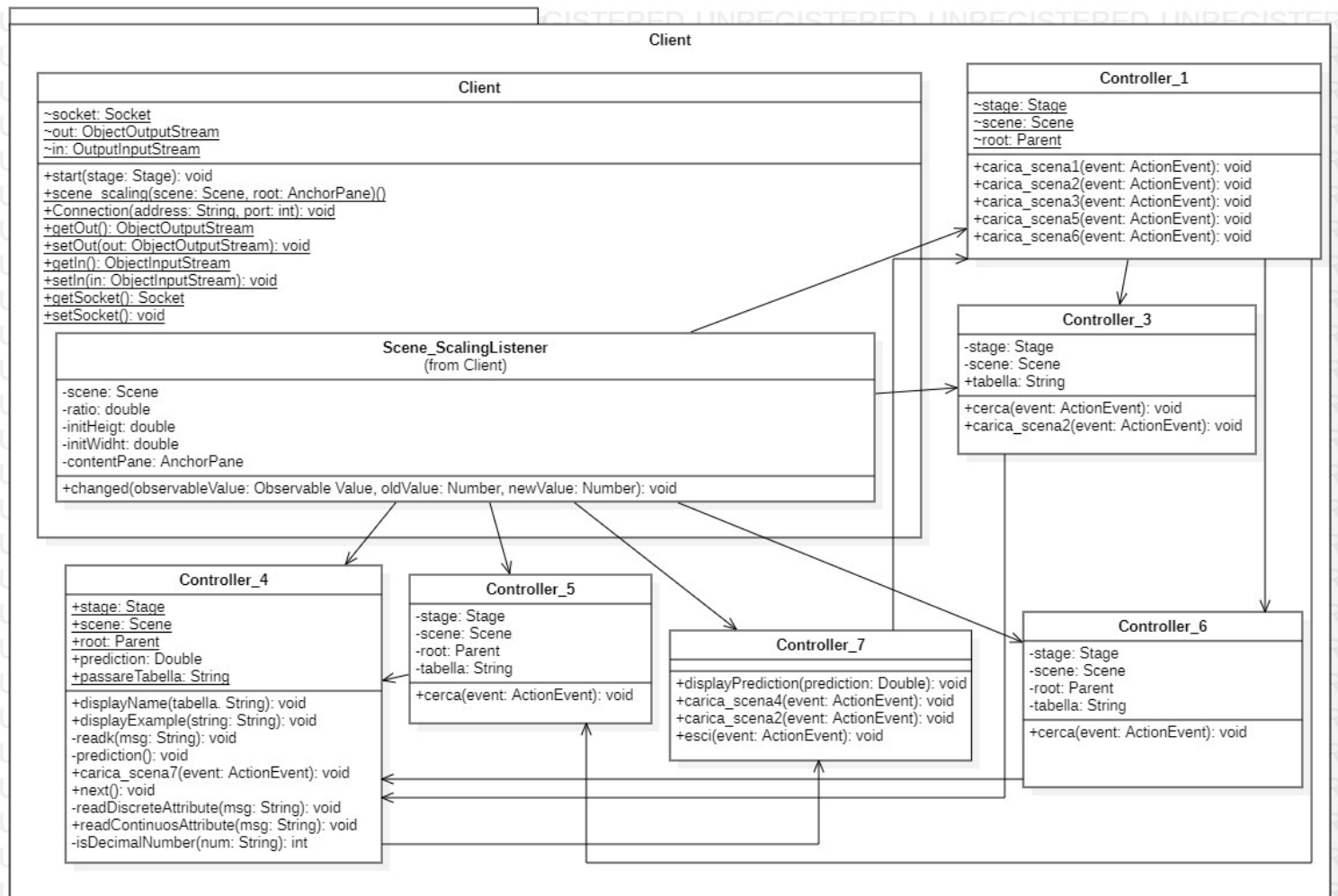
1  {
2    // Usare IntelliSense per informazioni sui possibili attributi.
3    // Al passaggio del mouse vengono visualizzate le descrizioni degli attributi esistenti.
4    // Per altre informazioni, visitare: https://go.microsoft.com/fwlink/?linkid=830387
5    "version": "0.2.0",
6    "configurations": [
7      {
8        "type": "java",
9        "name": "Launch Current File",
10       "request": "launch",
11       "mainClass": "${file}"
12     },
13     {
14       "type": "java",
15       "name": "Launch App",
16       "request": "launch",
17       "vmArgs": "--module-path /Users/Utente/Downloads/javafx-sdk-11.0.2/lib --add-modules javafx.controls,javafx.fxml",
18       "mainClass": "App",
19       "projectName": "JavaFxProject_c24a4afb"
20     }
21   ]
22 }

```

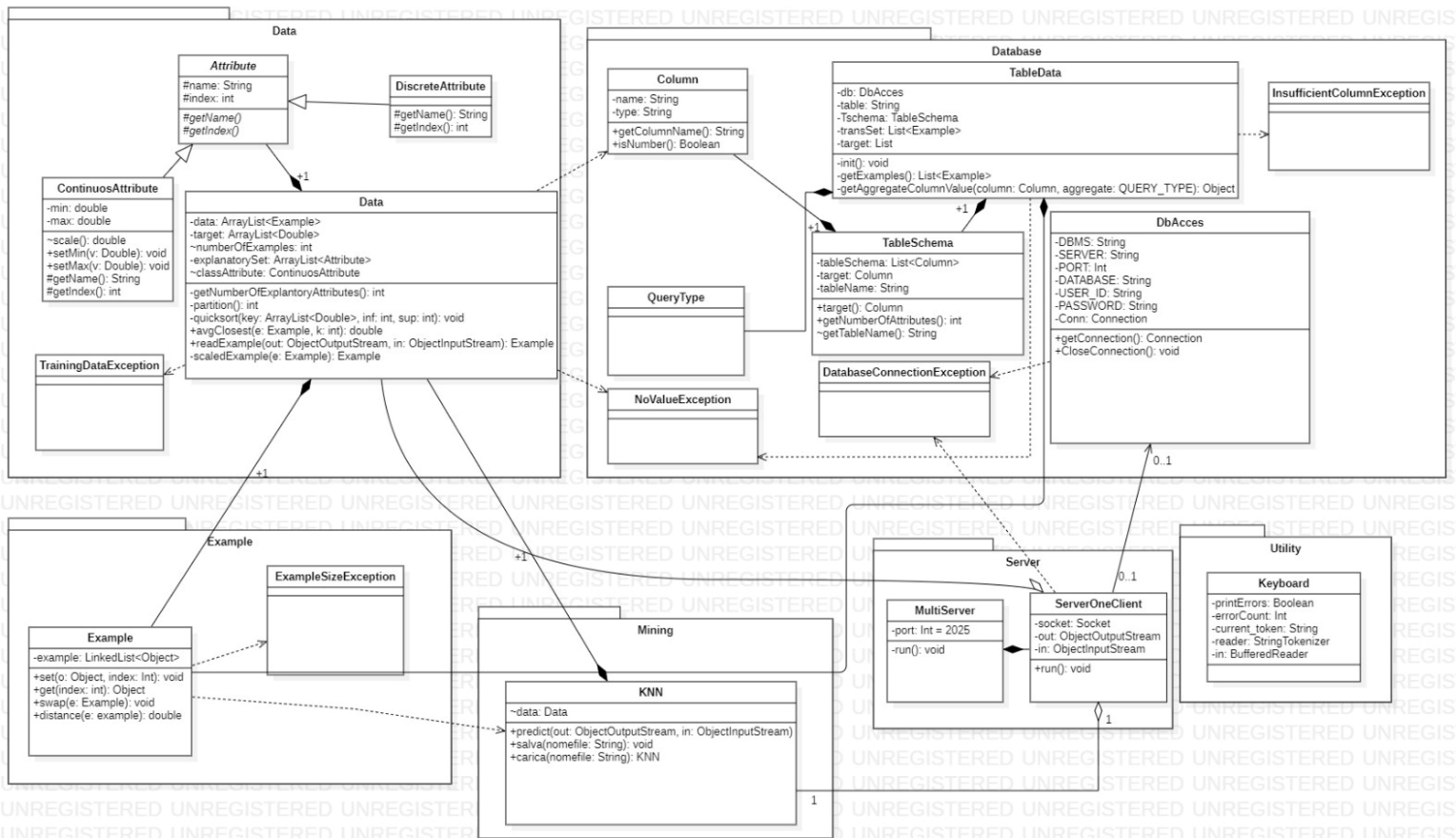
- Una volta fatto ciò riprovare il run del progetto e si vedrà subito che JavaFx è stato implementato correttamente e il progetto creato funziona allo stesso modo correttamente.

5. Diagramma delle classi:

- Diagramma delle classi Client:

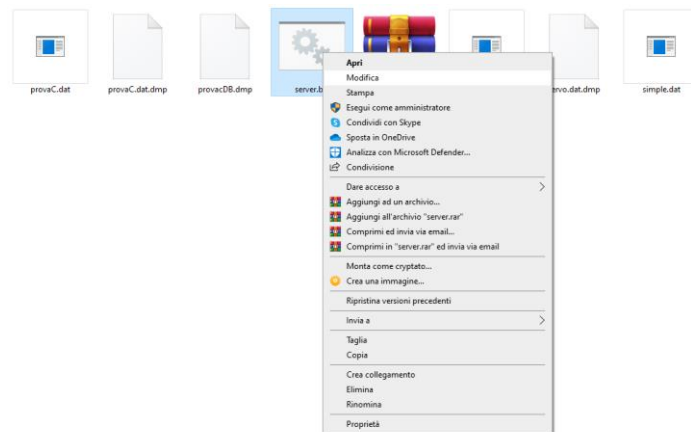


- Diagramma delle classi Server:



6. Guida di installazione BAT:

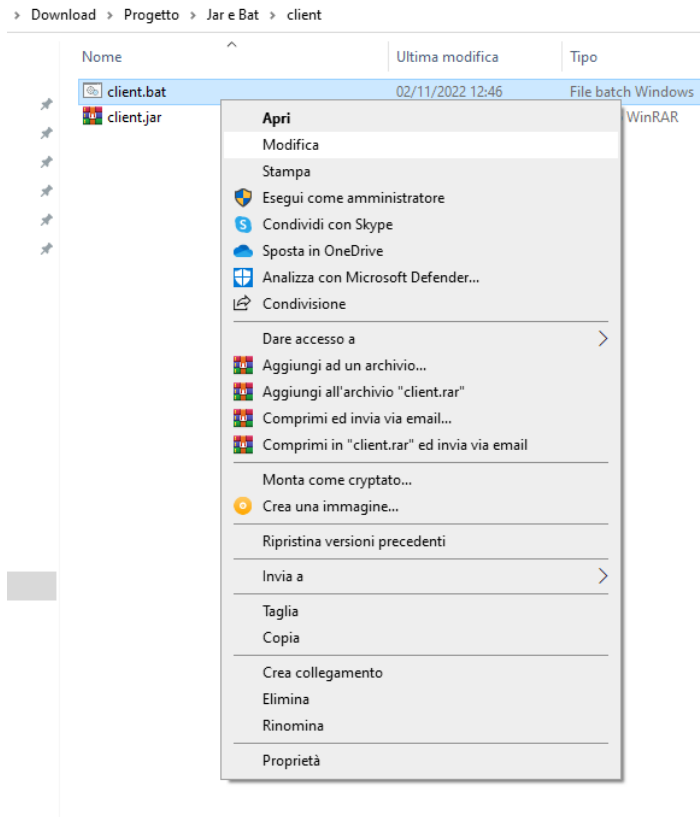
- Entrare nella cartella Jar e Bat.
- Immettere i binari (ovvero con estensione .dmp) e i file testuali (ovvero con estensione .dat) nella cartella server.
- Nella cartella server cliccare tasto destro sul file server.bat e cliccare su modifica.



- Dopo aver aperto il file .bat, immettere i percorsi scelti. Nella prima riga viene inserito il path relativo alla cartella server presente Jar e Bat a sua volta contenuta nella cartella Utente. Mentre nella seconda riga viene immesso il percorso relativo al file server.jar.

```
File Modifica Formato Visualizza ?
cd "C:\Users\dario\Desktop\Progetto\Jar e Bat\server"
java -jar C:\Users\Utente\Downloads\Progetto\Jar e Bat\server\server.jar
```

- In seguito, ritornare nella cartella Jar e Bat e cliccare tasto destro del mouse sul file client.bat e nuovamente premere su modifica.



- Anche qui immettere i percorsi scelti, in questo caso abbiamo il path relativo alla cartella client e in seguito quello del file client.jar.

```
@echo off
cd "C:\Users\dario\Desktop\Progetto\Progetto knn con estensione\Jar e Bat\client"
java --module-path "C:\Users\dario\Desktop\javafx-sdk-19\lib" --add-modules javafx.controls,javafx.fxml -jar client.jar
pause
```

- Infine, ricordarsi che per per l'avvio corretto dell'applicazione è necessario prima aprire il .bat del server e dopo quello del client.

```
C:\Users\dario\Desktop\Progetto\Progetto knn con estensione\Jar e Bat\server>java -jar server.jar
Server avviato sulla porta: 2025
```


7. Guida di installazione Jar:

PRIMA DI PROCEDERE CON L'INSTALLAZIONE BISOGNA ESSERE SICURI CHE NEL CODICE SIANO STATI IMMESSE LE CREDENZIALI CORRETTE PER L'ACCESSO AL DATABASE, INFATTI NEL CASO CI FOSSERO PROBLEMI, VERIFICARE CHE IL SOCKET ADDRESS (OVVERO PORTA E INDIRIZZO IP NEL SERVER) SIANO QUELLI CORRETTI.

- Aprire il prompt dei comandi ed eseguire il comando `java -jar` incollando il percorso del file `server.jar`, situato nella directory `server`.

```
C:\Users\dario\Desktop\Progetto\Progetto knn con estensione\Jar e Bat\server>java -jar server.jar  
Server avviato sulla porta: 2025
```

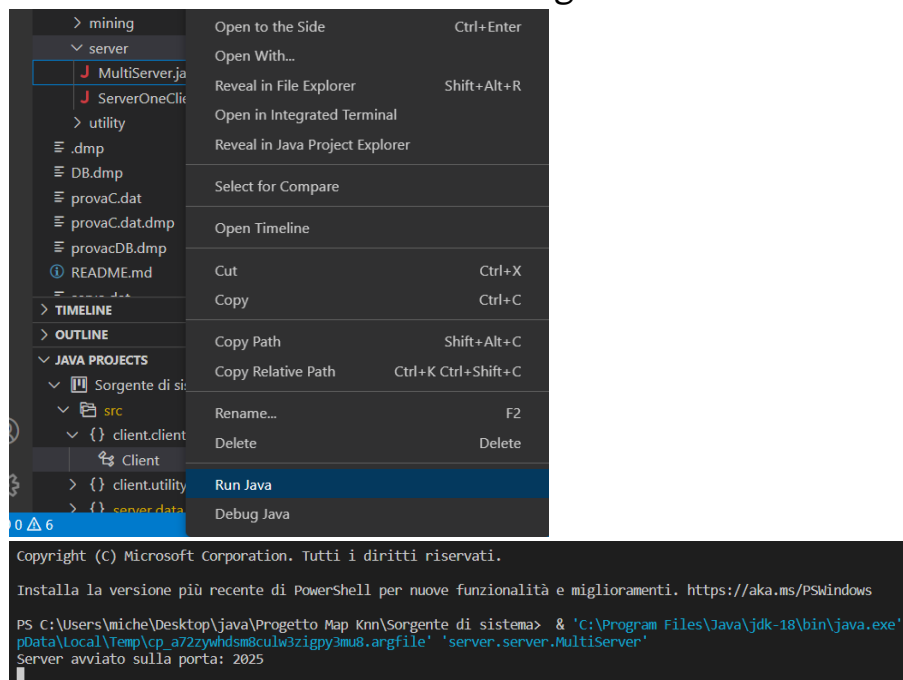
- Dopo di che aprire di nuovo il prompt ed eseguire lo stesso comando (`java -jar`) questa volta incollando il percorso del file `client.jar`, situato nella directory `client`.

```
C:\Users\dario\Desktop\Progetto\Progetto knn con estensione\Jar e Bat\client>java --module-path "C:\Users\dario\Desktop\javafx-sdk-19\lib" --add-modules javafx.controls,javafx.fxml -jar client.jar  
Socket[addr=/127.0.0.1,port=2025,localport=62229]
```


8. Guida utente con esempi di test

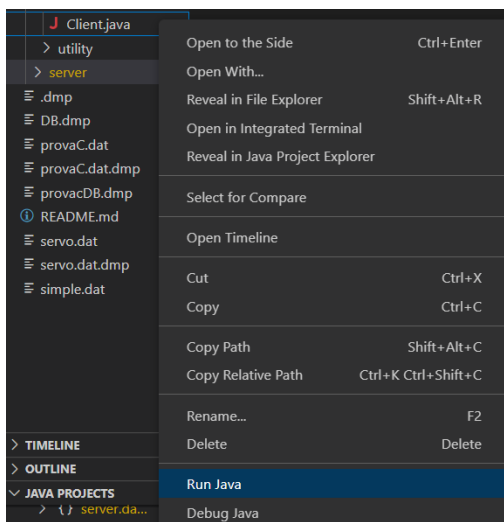
- Avviare il **server**:

Dopo aver importato correttamente tutte le librerie e procedere all'esecuzione del server come in figura

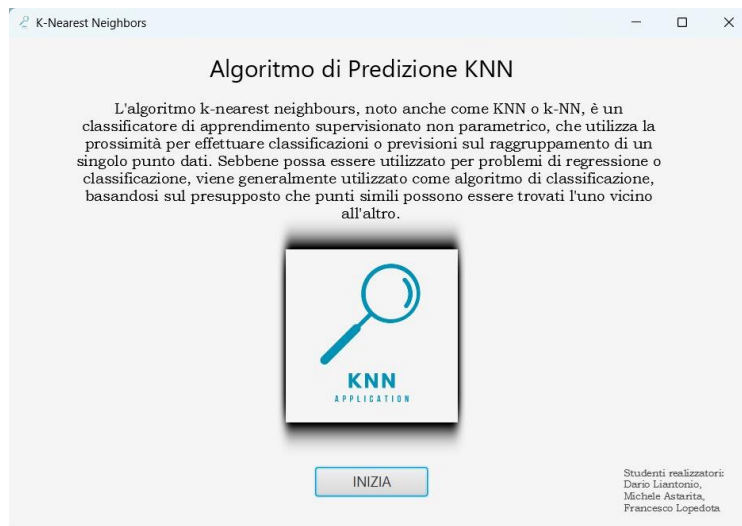


- Avviare il **client**:

Dopo aver importato correttamente tutte le librerie e procedere all'esecuzione del client come in figura



- Una volta stabilita la connessione tra client e server l'applicazione partirà mostrando l'**interfaccia grafica**.

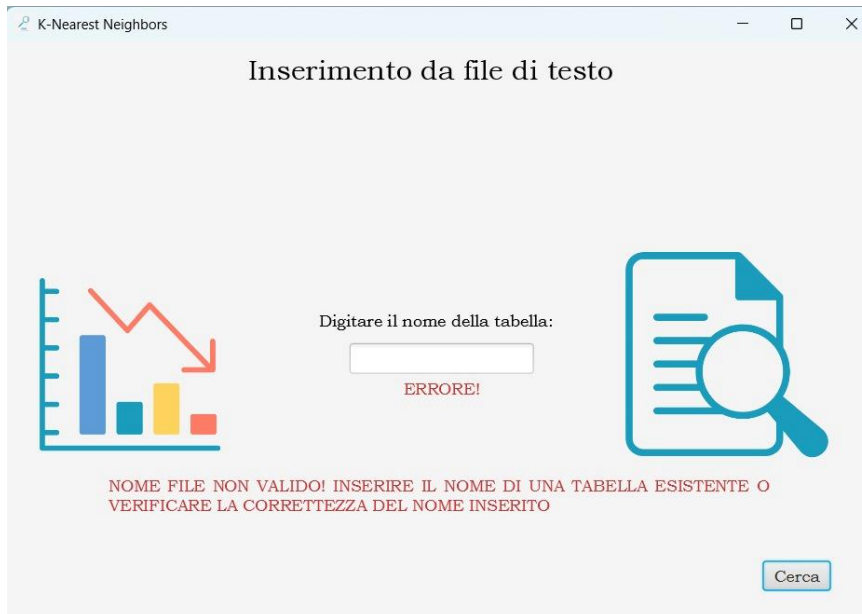


- Successivamente verrà chiesto all'utente di selezionare l'**origine del file** da cui prelevare il TrainingSet.



- Dopo aver selezionato l'origine del file, l'applicazione richiede l'inserimento del **nome della tabella**:

Test 1: inserimento nome tabella errato



K-Nearest Neighbors

Inserimento da file di testo

Digitare il nome della tabella:

ERRORE!

NOME FILE NON VALIDO! INSERIRE IL NOME DI UNA TABELLA ESISTENTE O VERIFICARE LA CORRETTEZZA DEL NOME INSERITO

Cerca

Test 2: inserimento nome tabella corretto, in questo caso l'applicazione andrà avanti nell'esecuzione

- Caricato correttamente il Knn sul server, verrà richiesto all'utente di inserire un **esempio** e il valore **k**



K-Nearest Neighbors

Caricamento dell'esempio su cui effettuare la predizione

Inserisci valore discreto ...

A Conferma

Inserisci valore continuo...

2 Conferma

Inserisci valore $k \geq 1$:

5 Conferma

Predizione

Test 1: Esempio inserito non rispetta la tipologia dell'attributo (discreto o continuo), o campo lasciato vuoto.

Inserisci valore discreto ...

Campo valore vuoto!

Test 2: Valore k inserito errato

Inserisci valore $k \geq 1$:

Valore non valido

Test 3: Valore k inserito corretto, in questo caso verrà visualizzata la predizione

- Successivamente verrà mostrata a schermo la predizione
- Infine, verrà chiesto all'utente di scegliere se effettuare un'altra predizione con lo stesso DataSet o eventualmente cambiare il file

