**YOUniversity**

Progetto di Ingegneria della Conoscenza

A.A. 2020/2021

**Componenti del gruppo:**

* *Anna Marzella*

Matricola: 689634 Mail: [a.marzella5@studenti.uniba.it](mailto:a.marzella5@studenti.uniba.it)

* *Martina Melero Cavallo*

Matricola: 699074 Mail: [m.melerocavallo@studenti.uniba.it](mailto:m.melerocavallo@studenti.uniba.it)

* *Giovanni Tauro*

Matricola: 699612 Mail: [g.tauro7@studenti.uniba.it](mailto:g.tauro7@studenti.uniba.it)

**Link del repository**: <https://github.com/mmc185/YOUniversity>

**Documentazione**

**Introduzione**

Il progetto consiste in un’applicazione chiamata *YOUniversity* che presenta delle funzionalità utili agli studenti, basandosi su argomenti studiati durante il corso.

Le feature che sono state inserite sono:

1. *Ricerca in un grafo:* il sistema calcola il percorso migliore, dato un punto di partenza scelto dallo studente, per raggiungere una delle destinazioni possibili.
2. *Predizione del voto*: il sistema cerca di predire, attraverso vari fattori, l’andamento dello studente in un certo esame.
3. *Interrogare una base di conoscenza:* il sistema permette allo studente di interrogare la KB per ottenere informazioni (ad esempio: elenco lezioni del lunedì, elenco lezioni di un determinato professore, edificio in cui si trova un’aula…).
4. **Ricerca in un grafo**

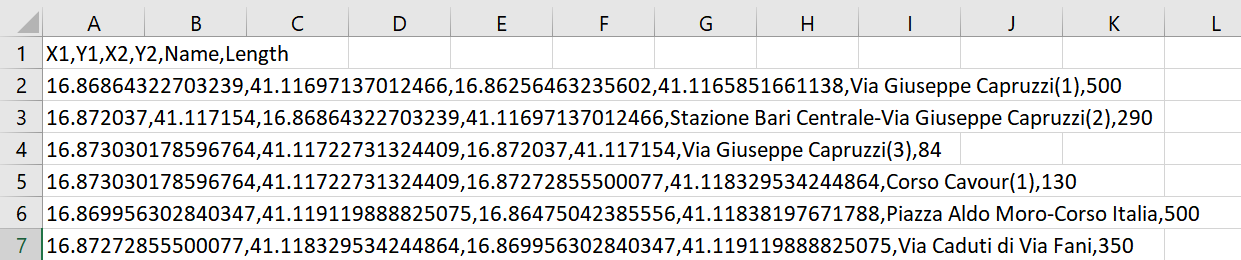
Si è scelto di implementare questa funzionalità affinché un ipotetico studente si possa orientare, attraverso l’uso di una mappa, per raggiungere una delle sedi universitarie.

Per la creazione della mappa, sono state selezionate la città di Bari e l’Università degli Studi di Bari “Aldo Moro”.

Le quattro sedi di destinazione sono:

* "Ateneo-Piazza Cesare Battisti(1)"
* "Politecnico/Campus-Via Edoardo Orabona"
* "Policlinico-Piazza Giulio Cesare"
* "Facoltà di Economia-Largo Abbazia Santa Scolastica"

Per la struttura del grafo, è stato creato appositamente un file *csv* contenente le informazioni riguardanti i luoghi e le vie di Bari.

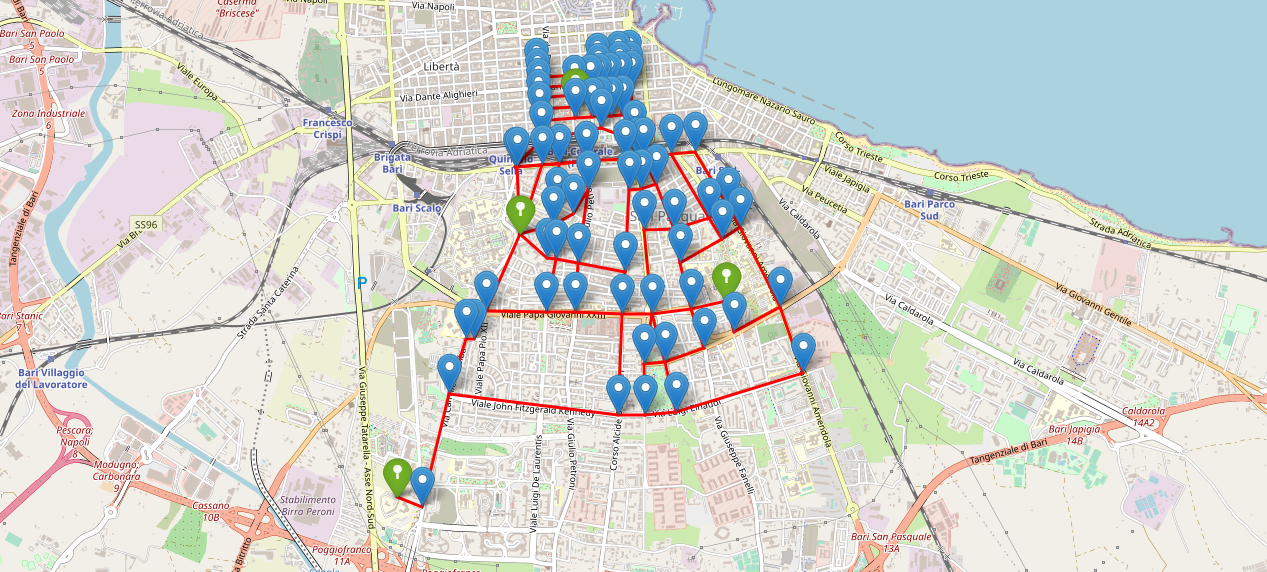


È stato utilizzato [*Google Maps*](https://www.google.it/maps) per prendere i riferimenti di inizio e fine di ognuna delle strade (talvolta suddivise in segmenti ridotti) che sono state mappate.   
Per ogni riga del dataset, sono state salvate:

* le coordinate (latitudine Y e longitudine X) del punto di partenza (X1, Y1), e del punto di arrivo (X2, Y2)
* nome della strada (Name)
* lunghezza della strada in metri (Length).

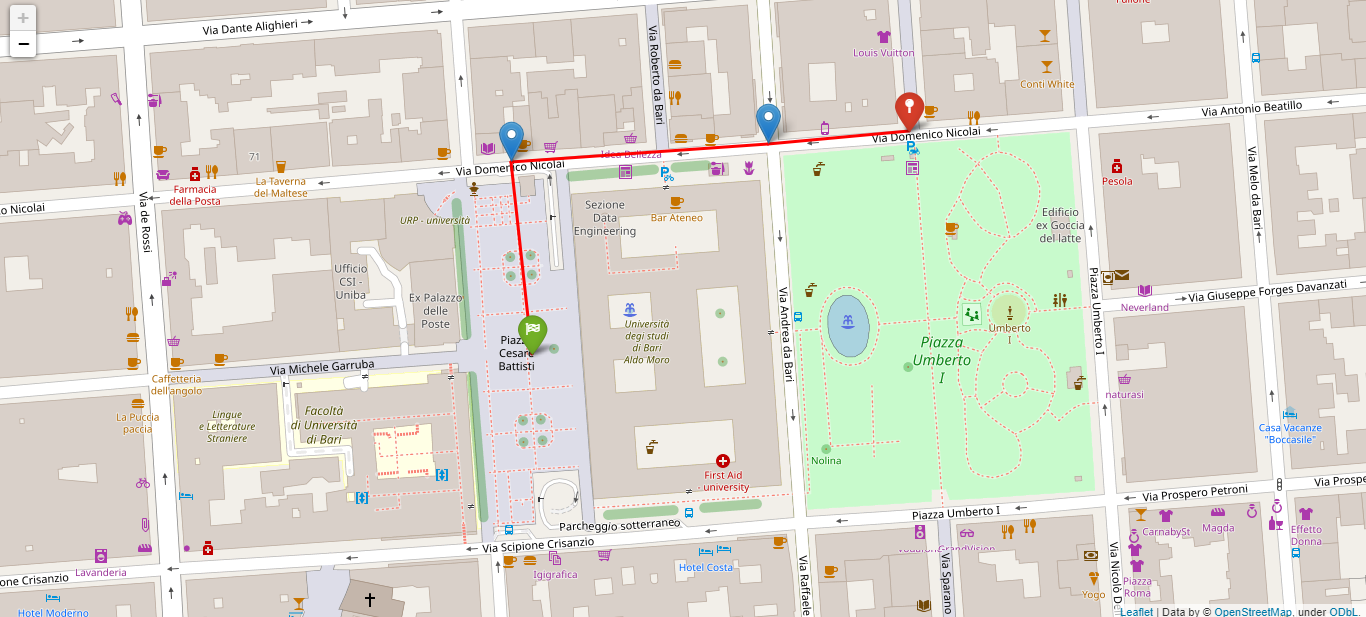
Dal [dataset](https://github.com/mmc185/YOUniversity/blob/main/resources/locations/Bari.csv) si è ricavato poi il grafo in cui ogni nodo rappresenta una coppia di coordinate (che identificano un unico luogo) e ogni arco una strada, con la sua lunghezza come peso.

Attraverso l’utilizzo della libreria folium, è stata applicata la parte di visualizzazione grafica sul servizio di [*OpenStreetMap*](http://www.openstreetmap.org).

**

Per la ricerca su grafo, è stato implementato l’algoritmo **A\*** con funzione euristica calcolata mediante la *distanza euclidea* (in linea d’aria tra due punti).

Il risultato dell’algoritmo viene poi visualizzato sulla mappa tracciando in rosso il percorso e indicando con un’icona rossa il punto di partenza, verde l’obiettivo e i nodi intermedi in blu.

**

Contenuto del package *Pathfinding*:

* *GeoLocationsUtilities*: file contenente utilities per la gestione dei luoghi
* *GraphUtilies*: file contenente classi di supporto al grafo
* *PathFinding*: file di connessione tra la struttura del grafo e la mappa
* *SearchProblemUtilities*: file contenente utilities per effettuare la ricerca

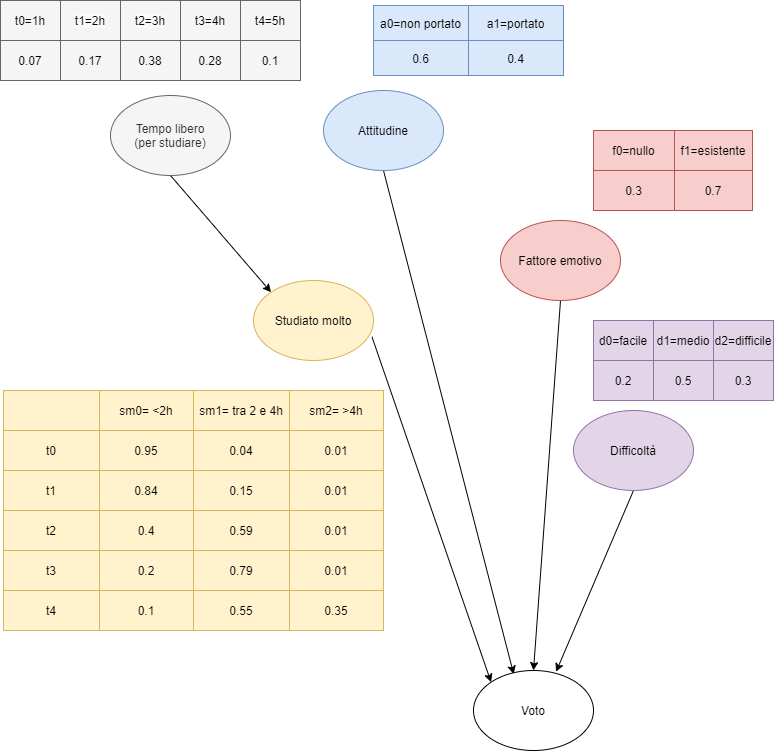
Librerie utilizzate:

* [Folium](https://python-visualization.github.io/folium/)

1. **Predizione del voto**

Per questa funzionalità si è creata una **Belief Network** (**Rete Bayesiana**) per predire il voto che lo studente può ottenere ad un determinato esame in base a dei fattori.

La rete bayesiana costruita è la seguente:



La visualizzazione testuale completa delle tabelle è presente nel file [*reteBayesiana.txt*](https://github.com/mmc185/YOUniversity/blob/main/doc/reteBayesiana.txt)

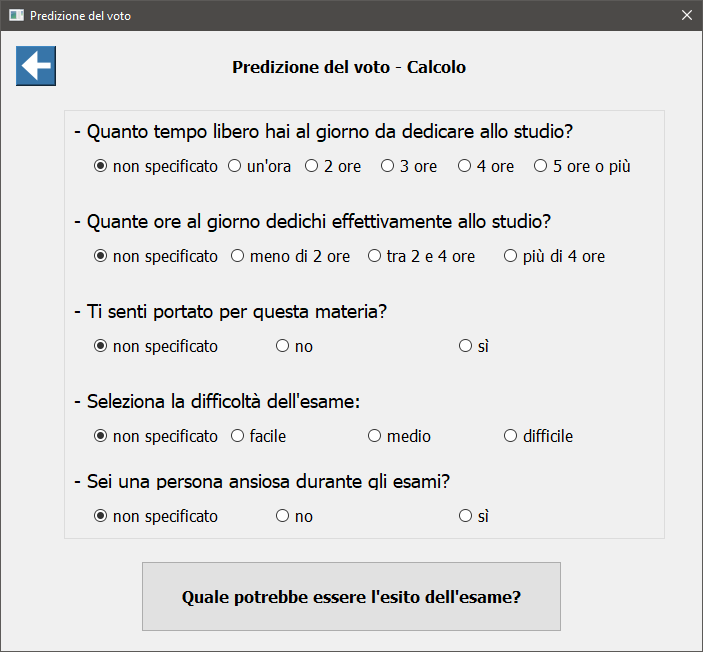
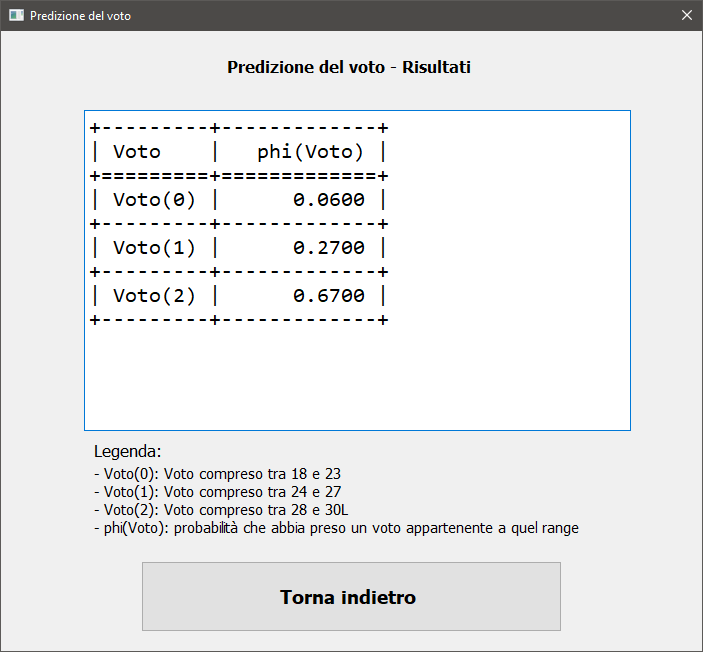
Come si evince dall’immagine, la probabilità di ottenere un voto appartenente a un certo range ( [18-23], [24-27], [28-30L] ) è influenzata dalle seguenti features:

* *difficoltà* dell’esame [facile, normale, difficile]
* *fattore emotivo* (ansia) [si, no]
* *attitudine* alla materia [si, no], per indicare se lo studente è portato o meno per quella materia
* *ore* di studio giornaliere [x<2h, 2h≤x≤4h, x>4h]
  + influenzate, a loro volta, anche dal *tempo libero* che lo studente potrebbe dedicare allo studio [1h, 2h ,3h, 4h, 5h]

In particolare, il tempo libero è stato pensato considerando il periodo di esami, in cui non vi sono lezioni e gli studenti hanno più tempo per studiare. Inoltre, si è scelto di osservare anche questa variabile poiché, ad esempio, potrebbero esserci studenti lavoratori che non possono dedicarsi completamente allo studio.

Allo studente è richiesto di indicare i valori di uno o più fattori e il sistema calcolerà la predizione del voto in base alle probabilità di essi.

La libreria che implementa la rete bayesiana è *pgmpy*, che permette di creare le tabelle di distribuzione di probabilità e fare inferenza sulla predizione del voto, basandosi sulla struttura della Belief Network.

Contenuto del package *bayesianNetwork*:

* *bayesianNetwork*: file che contiene la creazione della Belief Network e la funzione per fare inferenza.

Librerie utilizzate:

* [pgmpy](https://pgmpy.org/)

1. **Interrogare la base di conoscenza**

Un’altra funzionalità del sistema permette allo studente di interrogare la base di conoscenza al fine di ottenere varie informazioni, tra cui: orari delle lezioni, quali professori insegnano una certa materia o tutte le lezioni di un determinato giorno della settimana.

Per far ciò è stata usata la libreria *pytholog* che permetteva di popolare una base di conoscenza ed effettuare query su di essa.

Le query:

* Permettono di verificare se una clausola è conseguenza logica della base di conoscenza (in questo caso, il sistema risponderà “*Yes*”, altrimenti “*No*”).
* Possono comprendere variabili in modo che la risposta riporti tutti i valori che possono essere assunti dalla/e variabile/i.

*Esempio*: la query “*insegna(P,M)*” ridarà tutte le coppie ove *P* è un professore e *M* è la materia insegnata.

Gli assiomi utilizzati nella KB sono i seguenti (per la maggior parte, è riportato solo un esempio con relativa spiegazione) :

* insegna(fanizzi,icon) insegna(Professore, Materia)
* in(aula4,palazzo\_aule) in(Aula, Edificio)
* corso(di\_terlizzi,a1) corso(Professore, Anno\_corso)
* orario(di\_terlizzi,h9,lunedi) orario(Professore, Ora\_inizio, Giorno)
* lezione(M,H,G,P):-insegna(P,M),orario(P,H,G)   
  lezione(Materia, Ora\_inizio, Giorno, Professore):-insegna(Professore, Materia),orario(Professore, Ora\_inizio, Giorno)
* classe(a1,aula4) classe(Anno\_corso, Aula)
* luogo(C,X):-classe(C,A),in(A,X)   
  luogo(Anno\_corso, Palazzo):-classe(Anno\_corso, Aula),in(Aula, Palazzo)

Le informazioni salvate nella KB riguardano il primo semestre dei tre anni del corso di Informatica.

Contenuto del package *Prolog*:

* *prolog*: file che contiene la creazione della KB e la funzione per interrogarla.

Librerie utilizzate:

* [pytholog](https://mnoorfawi.github.io/pytholog/)

**Interfaccia grafica**

Per collegare le varie funzionalità, è stata creata un’interfaccia grafica molto semplice attraverso l’utilizzo del tool [*Qt Designer*](https://github.com/altendky/pyqt-tools).



Librerie:

* [PyQt5](https://www.riverbankcomputing.com/software/pyqt/)

**Sviluppi futuri**

Questo sistema può essere esteso in più modi, ad esempio: ampliando la KB, inserendo altri luoghi e/o strade sulla mappa o aggiungendo altre funzionalità.