

Fugitive simulator

Progetto di Davide Facchini, mat. 717327

Tema del progetto

In «Fugitive Simulator» l'utente è un latitante ricercato dalla Polizia in Italia per aver commesso dei crimini.

In questa simulazione, l'utente può predire quanti anni di galera gli possono essere dati in base al proprio genere di appartenenza, alla propria età e al reato commesso.

Oltre a ciò, l'utente (o il latitante), può decidere di spostarsi di regione in regione per sfuggire alla polizia, solo che deve stare attento alla concentrazione della polizia nelle varie regioni; più è alta la concentrazione di forze dell'ordine e più sarà alta la probabilità di essere catturati.

Per evitare ciò, l'utente si può avvalere di altre funzionalità del sistema per scoprire quale sia il rischio di cattura delle varie regioni dell'Italia, se vi è possibile passare da una regione all'altra o trovare un percorso ottimale per poter scappare per l'Italia.

Argomenti teorici

Gli argomenti teorici utilizzati per poter svolgere questo progetto sono:

- ❖ Classificatore con albero decisionale e rispettiva valutazione.
- ❖ Base di conoscenza con interrogazioni alla base.
- ❖ Ricerca in un grafo.

Classificatore

Per il classificatore, si è deciso di usare un albero decisionale in modo tale che all'utente, durante l'esecuzione del programma, vengano poste delle domande riguardanti la propria età e il reato commesso.

In base alle scelte compiute dall'utente, il sistema ragionerà in modo tale da dargli il numero di anni di galera che gli spetta.

Ad esempio: se è un minorenne, a parità di reato, avrà meno anni rispetto ad un maggiorenne. Più si va avanti con l'età, più la pena è grave in quanto ci si aspetterebbe una maturità maggiore. Inoltre gli anni di galera non saranno uguali per tutti i tipi di reato (omicidio, rapina ed estorsione), in quanto questi non hanno tutti la stessa importanza.

Valutazione

Per la valutazione del classificatore si è deciso di utilizzare la «K-Fold Cross Validation» in quanto si ha a disposizione un dataset che raggiunge un valore minimo di tuple adeguato per permettere un corretto funzionamento.

In questo caso i fold per valutare il classificatore sono 6 e quindi il sistema farà le proprie valutazioni alternando i vari fold tra esempi di training ed esempi di test.

Base di conoscenza

Per la Base di Conoscenza, si è deciso di costruirne una con la logica del primo ordine. In questo caso è più naturale eseguire delle interrogazioni alla Base, chiedendo di esplicare anche come si è giunti ad una determinata risposta.

L'utente potrà chiedere, ad esempio, se una regione in particolare avrà un determinato rischio, potendo chiedere anche come il sistema gli ha fornito la risposta tramite il comando «**how**»; oppure l'utente potrà chiedere se si può passare da una regione ad un'altra sempre potendo chiedere, come per l'altra interrogazione, come il «**reasoner**» è arrivato a determinate risposte.

Ricerca nel grafo

Per affrontare il problema di ricerca all'interno del grafo (i cui nodi sono costituiti dalle regioni dell'Italia e gli archi sono i collegamenti diretti con le regioni confinanti), si è deciso di utilizzare un algoritmo di ricerca informato, in questo caso l'algoritmo A*.

La decisione è stata presa in questa direzione, in quanto l'utente deve poter ricercare un percorso che sia abbastanza sicuro da permettergli di non essere catturato dalle forze di polizia. Per fare ciò, gli archi hanno dei pesi (quindi si ha un grafo pesato) ottenuti dal calcolo della presenza di polizia (un indice che varia da 1 a 100) all'interno di due regioni che sono collegate direttamente (ossia sono confinanti). Inoltre viene utilizzata una funzione euristica che stima il costo per spostarsi verso regioni che non sono confinanti tra di loro e, delle quali, non si ha il costo reale a disposizione.

Per complicare la vita al fuggitivo, alle regioni sono state assegnate delle pericolosità in base al valore dell'indice di presenza della polizia.

Ricerca nel grafo

Queste pericolosità sono state divise in una scala di 5 valori, ossia:

- ❖ Molto basso.
- ❖ Basso.
- ❖ Moderato.
- ❖ Alto.
- ❖ Molto alto.

In base a queste pericolosità, il sistema deciderà se ci sarà un percorso o meno da una regione verso un'altra. Infatti non ci si potrà muovere da e verso le regioni che hanno un rischio «molto alto» di poter essere catturati, in quanto si è sicuri di essere arrestati per uno spostamento, quindi in questo caso è meglio continuare a nascondersi nel proprio rifugio. Mentre, se il rischio delle regioni va da «molto basso» ad «alto», allora ci si potrà spostare.

L'algoritmo di ricerca si occuperà di trovare il percorso meno costoso (e quindi meno pericoloso) per l'utente.

Funzionamento programma

Il programma funziona eseguendo il «main» una volta aperto il codice sorgente. Appena fatto partire darà un messaggio di benvenuto e successivamente si potrà digitare i comandi, che sono:

- ◊ *classificazione - fa partire il classificatore;*
- ◊ *valutazioneClassificatore - esegue la valutazione del classificatore;*
- ◊ *trovaRegioniRischioMoltoBasso - trova le regioni con rischio «molto basso»;*
- ◊ *trovaRegioniRischioBasso - trova le regioni con rischio «basso»;*
- ◊ *trovaRegioniRischioModerato - trova le regioni con rischio «moderato»;*
- ◊ *trovaRegioniRischioAlto - trova le regioni con rischio «alto»;*
- ◊ *trovaRegioniRischioMoltoAlto - trova le regioni con rischio «molto alto»;*
- ◊ *visualizzaListaRischi – stampa la lista dei rischi che le regioni possono assumere;*
- ◊ *trovaPercorso – trova un percorso ottimale, se esiste, da regione A a regione B;*
- ◊ *verificaPassaggio – verifica se si può passare da una regione A ad una regione B;*
- ◊ *verificaRischioRegione – verifica se una determinata regione ha un determinato rischio scelto dall’utente;*
- ◊ *esci – per uscire dall’applicazione*

Link repository

<https://github.com/davfac/Fugitive-Simulator>

Grazie per l'attenzione!