Chicago Crimes

Relazione Caso di Studio

AA 2022-23

Gruppo di lavoro

- Walter Mangione, 747833, w.mangione@studenti.uniba.it
- Antonio Mari, 741267, a.mari8@studenti.uniba.it

https://github.com/antoniomari/Ingegneria-della-conoscenza

Introduzione

Il caso di studio, sviluppato tramite l'utilizzo del linguaggio Python, tratta dei crimini avvenuti nella città di Chicago utilizzando dati reali, caricati dal **Chicago's Department of Innovation and Technology (DoIT)**, sul **Chicago Data Portal**.

Sono stati usati dataset sui crimini registrati dall'anno 2010 al 2022, sugli arresti effettuati dalla polizia di Chicago, sulle vittime di crimini coinvolgenti armi da fuoco e su dati sanitari ed economici sulle Community Areas, ovvero su aree di suddivisione geografica della città.

Ci si è occupati del pre-processing e dell'integrazione dei dati all'interno di una base di conoscenza Prolog, ingegnerizzando clausole che permettessero di inferire nuove informazioni attraverso il ragionamento automatico: queste vengono sfruttate in task di apprendimento supervisionato.

E' stato infine affrontato il problema della classificazione dei crimini in omicidi/non-omicidi, per il quale si sono tentati diversi approcci (per trattare feature categoriche e numeriche).

In particolare:

1. I dataset contengono diverse features categoriche (numeri che corrispondono a identificatori di diverse unità geografiche) con valori su cui non è definito un ordine naturale.
2. La rappresentazione proposta dei dati prevede la creazione di una Knowledge Base, in modo tale da poter modellare la struttura di individui e relazioni tra essi presenti nei dataset. Questo ha semplificato la creazione di features volte a riassumere i dati sugli arresti e sulle vittime associati a ogni crimine.
3. Un ulteriore aspetto su cui ci si è concentrati si incentra su alcuni dati delle vittime non direttamente utilizzabili per task di apprendimento supervisionato: stiamo parlando dei loro nomi e cognomi, i quali vengono analizzati separatamente effettuando clustering.
    a. Si è automatizzato il recupero di testo contenuto nelle pagine web riguardanti ciascuna vittima
    b. Si è calcolato il corrispettivo *embedding*, sfruttando Word2Vec
    c. Si è utilizzato l'algoritmo *k-means* per l'individuazione dei cluster.

Una sezione del documento riguarda l'estensione alla Knowledge Base pensata in un secondo momento, a seguito di alcuni risultati raggiunti lavorando per il task di classificazione.

Elenco argomenti di interesse

- **Rappresentazione e ragionamento relazionale**: utilizzo di Prolog per il ragionamento su una base di conoscenza ingegnerizzata partendo dai dati contenuti nei dataset, permettendo di inferire nuove informazioni.
- **Apprendimento supervisionato:** alberi di decisione e metodi ensemble basati su di essi (Random Forest, Ada Boost, Gradient Boosting), Naive Bayes categorico e KNN per portare a termine un task di classificazione binaria.
- **Apprendimento non supervisionato:** utilizzo del k-means per la creazione di cluster di embedding di documenti recuperati dal web.

Descrizione del dominio e dati utilizzati

L'idea originaria del caso di studio è partita dal dataset trovato su Kaggle, contenente dati riguardanti i crimini registrati nella città metropolitana di Chicago, raggruppati per anno a partire dal 2010 fino al 2022.

Il dataset contiene dati riguardanti

- Data e ora del crimine
- Locazione geografica del crimine (tra cui un valore approssimato di latitudine e longitudine) e descrizione del luogo (es. "street", "house", …)
- Tipologia del crimine (secondo Illinois Uniform Crime Reporting code e FBI Code)
- Altre caratteristiche del crimine (se è domestico, se vi è stato un arresto associato).

In particolare, alcune feature meritano una descrizione appropriata (le altre sono intuitive e ben spiegate nella pagina al link riportato):

Nome Feature	Descrizione	Tipo
Block	Corrisponde a un isolato di Chicago	Stringa di indirizzo (ad esempio "07300 S KINGSTON AVE")
IUCR, Primary Type, Description	Lo IUCR(Illinois Uniform Crime Reporting) è un codice di 4 cifre usato per classificare incidenti criminali. La feature *Primary Type* è una descrizione associata allo IUCR stesso. Invece, la feature *Description* è una sottocategoria di Primary Type, quindi anch'essa dipendente da IUCR.	Codice di 4 cifre (ad esempio 1631)
Beat	Il *beat* è la più piccola area di suddivisione dei distretti di Chicago. Conseguentemente, è dipendente da *District*.	Codice di 4 cifre (ad esempio 0925)
District	La feature *district* fa riferimento al distretto di polizia in cui il crimine si è verificato.	Codice di 2 cifre (ad esempio 16)
Ward	La feature *Ward* individua un'area amministrativa/politica. Ogni Ward è amministrato da un assessore. Ci sono 50 Ward	Codice numerico (ad esempio 44)
Community area	La feature *Community Area* fa riferimento ad una zona geografica della città. Ve ne sono 77.	Codice numerico (ad esempio 30)

Data la povertà del dataset (escluso l'aspetto geografico) si sono effettuate ulteriori ricerche sull'origine dei dati, scoprendo che essi provengono direttamente dal **Chicago Data Portal**, portale gestito dal **Chicago's Department of Innovation and Technology (DoIT)**, su cui sono presenti oltre 1000 dataset relativi a diversi domini di pertinenza dell'amministrazione cittadina.

Si è andati alla ricerca di ulteriori dataset per integrarli con quello trovato, al fine di creare una knowledge base in cui includere conoscenza più ampia per poter ricostruire quante più informazioni sui crimini: i diversi valori geografici ampliano le prospettive di ricerca, consentendo di collegare informazioni su diversi campi.

In particolare, sebbene fonti che includessero informazioni sui crimini specifici fossero le priorità, si sono cercati anche dati relativi ad altri campi tra quelli analizzati. In corrispondenza con il singolo crimine (tramite Case Number), sono stati trovati due dataset:

1. il primo comprende dati sugli arresti portati a termine dal **Chicago Police Department** (**CPD**), contenente data dell'arresto, etnia dell'arrestato e i capi di imputazione (*charge*);
2. il secondo comprende dati sulle vittime di crimini (fatali e non) in cui sono coinvolte armi da fuoco. Contiene informazioni geografiche (molte delle quali sono ripetizioni delle stesse presenti nel primo dataset trovato), informazioni sulla vittimizzazione (*victimization*), quali tipologia di crimine o se sono riportate ferite, e informazioni sulle vittime (età, sesso, etnia e in particolare nomi e cognomi di vittime di omicidi).

Inoltre è stato trovato un dataset comprendente statistiche riguardanti la sanità, i cui dati possono essere messi in corrispondenza con i vari crimini sulla base del valore Community Area: nello specifico sono presenti tassi di incidenza di diverse malattie nonché indicatori economici e di istruzione/tasso di disoccupazione caratteristici di ogni Community. Questi ultimi sono di maggiore interesse per il nostro caso.

Preprocessing dei dati

Innanzitutto si sono selezionati solo i crimini che hanno una corrispondenza nei dati degli arresti e delle vittime: in questo modo non si potrà effettuare ragionamento e apprendimento che siano validi in generale sui crimini nella città di Chicago, ma si lavorerà con la distribuzione dei "*crimini che coinvolgono armi da fuoco, condizionata dal fatto che vi è stato almeno un arresto*".

Innanzitutto si è notato che la corrispondenza tra crimini e il resto dei dati correlati è di tipo uno a molti

- Ci possono essere più arresti collegati allo stesso crimine, ognuno dei quali è relativo a un singolo arrestato. Ovvero, se vengono arrestati 2 criminali per lo stesso crimine saranno registrati 2 arresti.
- Ci possono essere più vittime collegate allo stesso crimine, ognuna corrispondente a una *vittimizzazione*.

Di seguito sono riportate le quantità di dati disponibili i quali hanno corrispondenze nei 3 dataset.

Dato	Numero	Numero medio per crimine	Valore massimo per crimine
Crimini	2505	-	-
Arresti	3062	1.22	6
Vittime	3228	1.29	13

Eliminazione di feature

Per il dataset dei crimini, oltre all'eliminazione di dati poco significativi (es. "Updated On"):

- IUCR, Primary Type, Description e FBI Code sono stati eliminati in quanto il dataset delle vittime contiene dati inerenti più specifici, discussi in seguito.
- Il campo Arrest è stato eliminato in quanto, essendo un valore booleano che indica se c'è stato un arresto per un dato crimine, risulta inconsistente col fatto che ogni crimine abbia una corrispondenza con almeno un arresto.
    - Si ipotizza che tale campo, se impostato a 1, indichi che l'arresto è stato registrato con il crimine stesso (0 altrimenti).
- Sono stati eliminati i dati di contenenti coordinate geografiche ridondanti, trattenuti solamente "Latitude" e "Longitude".

Per il dataset degli arresti, sono state inizialmente trattenute tutte le feature.

Per il dataset delle vittime, vanno innanzitutto effettuati chiarimenti su alcune delle feature originariamente presenti: le righe in rosso corrispondono alle feature rimosse, quelle in blu invece non sono state rimosse.

Nome Feature	Descrizione	Tipo
BLOCK, WARD, COMMUNITY_AREA, DISTRICT, BEAT, LATITUDE,	Le stesse sono presenti nel dataset precedente.	-

LONGITUDE, POSITION, LOCATION_DESCRIPTION		
MONTH, HOUR	Le stesse sono ridondanti e ricavabili dalla data	-
VICTIMIZATION_PRIMARY	Testo descrivente la tipologia di crimine commesso nei confronti della vittima (associato a un codice IUCR).	Stringa di testo (ad esempio “homicide”)
INCIDENT_PRIMARY	Testo descrivente la tipologia del crimine più grave commesso. Ad esempio, se in un crimine ci sono tre vittime e solo una è stata uccisa, la vittimizzazione per le altre due sarà diversa da omicidio mentre il valore di INCIDENT_PRIMARY per tutte e 3 sarà sempre “homicide”.	Stringa di testo (ad esempio “battery”)
AREA	Area geografica in cui il crimine è avvenuto. Ve ne sono 5.	Numero compreso tra 1 e 5.
STREET_OUTREACH_ORGANIZATION	Eventuale organizzazione di volontariato (nei confronti dei senzatetto) che opera nella zona in cui il crimine è accaduto.	Stringa di testo (ad esempio “Claretian Associates-South Shore”)
VICTIMIZATION_FBI_CD, INCIDENT_FBI_CD, INCIDENT_FBI_DESCR, VICTIMIZATION_IUCR_SECONDARY, VICTIMIZATION_FBI_DESCR	Non aggiungono alcuna nuova informazione di interesse, ricalcano aspetti correlati a *VICTIMIZATION_PRIMARY* e *INCIDENT_PRIMARY*	-
STATE_HOUSE_DISTRICT, STATE_SENATE_DISTRICT	Distretti individuati a livello legislativo in cui il crimine è avvenuto.	-

Le altre informazioni mantenute provenienti da questo dataset

- DATE, DAY_OF_WEEK
- GUNSHOT → indica se la vittima ha riportato una ferita da arma da fuoco
- ZIP_CODE → informazione geografica aggiuntiva non presente nel dataset dei crimini
- AGE → originariamente presente come fascia d’età (es [20-29]), tale campo è stato rielaborato scegliendo un valore centrale dell'intervallo come migliore approssimazione per l’età di una vittima. In seguito si è tenuto conto dell’ampio margine di errore: questa modifica è stata fatta per permettere di trattare numericamente questo dato.

- SEX → ‘m’, ‘f’ o ‘unknown’
- VICTIM_RACE → etnia della vittima, i cui valori possibili sono
    - ‘blk’ → black
    - ‘whi’ → white
    - ‘wwh’ → white hispanic
    - ‘wbh’ → black hispanic
    - ‘api’ → asian / pacific islander
    - ‘i’ → amer indian / alaskan native
    - ‘unknown’
- VICTIM_FIRST_NAME, VICTIM_MI, VICTIM_LAST_NAME → nomi e cognomi delle vittime, sono dati univoci e non numerici e non sono trattabili numericamente.
    - Sono disponibili solo per alcune delle vittime che sono state uccise (in tutto le coppie <nomi, cognomi> sono 1334)
    - Sono stati sfruttati nel clustering, trattato più avanti.

Per il dataset dei dati sanitari, sono state selezionate le seguenti features (numeriche), nella speranza che possano fornire un contributo per task di apprendimento supervisionato.

- INCOME → reddito pro-capite
- ASSAULT HOMICIDE → mortalità attribuita ad omicidi su 100,000 persone
- FIREARM → mortalità attribuita ad armi da fuoco su 100,000 persone
- BELOW_POVERTY_LEVEL → percentuale di famiglie sotto il livello di povertà
- HIGH_SCHOOL_DIPLOMA → percentuale di persone di almeno 25 anni che possiedono un diploma di scuola superiore
- UNEMPLOYMENT → percentuale dei disoccupati tra le persone (in forza lavoro) di età almeno 16 anni
- BIRTH_RATE → su 1000 persone

Conclusioni

Il fatto stesso di trattare relazioni uno a molti non permette un’integrazione immediata dei dati ma, per lavorare considerando il singolo crimine come esempio proveniente da una popolazione, è necessario passare a una diversa rappresentazione della conoscenza.

A tal fine, di seguito si illustra come è stata creata la knowledge base che ha permesso quest’integrazione, utile per:

- effettuare ragionamento automatico sfruttando la struttura di individui e relazioni, potendo inferire nuova conoscenza
- ingegnerizzare nuove feature numeriche e (in particolare) booleane, sfruttabili per task di apprendimento automatico
    - In contrapposizione a molte già esistenti, le quali hanno dominio discreto ma non una relazione d’ordine significativa. Feature di questo tipo saranno usate unicamente con l’algoritmo **Naive Bayes categorico**, sfruttando il fatto che tratti tutti i valori di un dominio discreto separatamente.

nota: la gestione dei valori nulli viene posticipata nelle fasi successive e, a seconda del caso, si prenderanno decisioni diverse. Verrà specificato ogni qualvolta si riscontrino dei valori nulli; in tutti gli altri casi si lascia implicito il fatto che non ce ne siano.

Creazione e Utilizzo Knowledge Base

Come anticipato, è stata creata una knowledge base sfruttando il linguaggio Prolog, interfacciandosi ad esso tramite la libreria **pyswip**, scritta per il linguaggio Python. A seguito della creazione e del popolamento con fatti provenienti direttamente dai dataset, la base di conoscenza è stata sfruttata per ingegnerizzare feature da utilizzare nella successiva fase di apprendimento supervisionato.

Individui

Si è scelto di modellare il dominio individuando le seguenti classi di individui, ognuna con un simbolo di funzione associato:

- crime(C): rappresenta il crimine il cui Case Number è C.
- arrest(A): rappresenta l'arresto il cui identificativo è A, includendo i dati dell'arrestato (che si limitano unicamente alla etnia, per cui si è scelto di non rappresentare separatamente individui di classe "arrestato").
- victim(V): rappresenta una vittima di un crimine, il cui codice identificativo è V.

Il caso di Community area fa eccezione, in quanto una community area è rappresentata come un individuo ma non utilizza un simbolo di funzione: per tutti i predicati specifici per essa, il primo argomento è il numero identificativo della community area.

Collegamento tra individui

Le relazioni che permettono di collegare le vittime e gli arresti a un dato crimine sono

```
hasArrest(crime(C), arrest(A))
victimization(crime(C), victim(V), T)
```

nota: T è il valore del campo VICTIMIZATION_PRIMARY per la vittima V.

Proprietà

Le proprietà associate agli individui corrispondono a feature presenti nei rispettivi dataset, salvo diversamente specificato. Non sono presenti tutti i campi ma solo quelli utili per inferire nuove informazioni tramite il ragionamento.

Le proprietà per un generico crimine C

```
location_description(crime(C), L)
beat(crime(C), B)
district(crime(C), D)
comm_area(crime(C), CA)
ward(crime(C), W)
crime_date(crime(C), datime(date(Anno, Mese, Giorno, Ora, Min, Sec)))
block(crime(C), B)
```

Le proprietà per un generico arresto A

```
arrest_date(arrest(A), datime(date(Anno, Mese, Giorno, Ora, Min, Sec)))
criminal_race(arrest(A), R)
num_of_charges(arrest(A), N)
```

num_of_charges è calcolato contando il numero di campi relativi ai dati "CHARGE_DESCRIPTION" avvalorati (minimo: 1, massimo: 4) e corrisponde al numero di capi di imputazione per un arresto.

Le proprietà per una generica vittima V

```
date_shoot(victim(V), datime(date(Anno, Mese, Giorno, Ora, Min, Sec)))
victim_race(victim(V), R)
victim_sex(victim(V), S)
incident(victim(V), I)
zip_code(victim(V), ZC)
victim_day_of_week(victim(V), DW)
state_house_district(victim(V), SHD)
state_senate_district(victim(V), SSD)
street_org(victim(V), Org)*
victim_age(victim(V), A)*
```

** gli ultimi due attributi possono presentare valori nulli nel dataset, in tal caso non sarà presente il fatto corrispondente alla vittima col valore nullo.*

Le proprietà per una community area il cui id sia CA (numerico in {1, …, 77})

```
comm_birth_rate(CA, BR)
comm_assault_homicide(CA, AH)
comm_firearm(CA, FA)
comm_poverty_level(CA, PL)
comm_hs_diploma(CA, HSD)
comm_income(CA, I)
comm_unemployment(CA, U)
```

Clausole definite

Sono state scritte diverse clausole definite per effettuare ragionamento: si presentano i principali schemi adottati per la loro definizione, riportandone testualmente alcune e riportando nomi e semantica (in linguaggio naturale) di altre (sarebbe inutilmente ripetitivo presentare la sintassi prolog anche di queste).

Clausole per conteggi geografici

Una tipologia di feature derivata per un crimine consiste nel numero totale di crimini avvenuti nella stessa unità geografica del dato crimine. Questo discorso viene applicato definendo come "unità geografica" le diverse informazioni di suddivisione del territorio disponibili.

In particolare, considerando come unità una delle seguenti (district, beat, ward, community area, block), essendo il collegamento tra crimine e unità geografica immediatamente presente, le clausole definite sono (*a titolo di esempio si mostrano quelle che riguardano district*):

```
same_district(crime(C1), crime(C2)) :- district(crime(C1), D),
district(crime(C2), D)
```

```
num_of_crimes_in_district(crime(C), N) :- findall(C1, same_district(crime(C),
crime(C1)), L), length(L, N)
```

Funzionamento: *si calcola la lista di tutti i crimini presenti nella stessa unità geografica di crime(C), utilizzando con findall/3; il numero di tali crimini sarà dato dalla lunghezza della lista.*

Per quanto riguarda il collegamento con unità geografiche che non sono direttamente delle proprietà dei crimini si presenta il caso esemplificativo dello zip_code, proprietà di un individuo vittima. L'unica

differenza col caso precedente sta nel fatto che la proprietà crime_zip_code è inferita per mezzo di victimization.

```
crime_zip_code(crime(C), Z) :- victimization(crime(C), victim(V), T),
zip_code(victim(V), Z)

same_zip_code(crime(C1), crime(C2)) :- crime_zip_code(crime(C1), Z),
crime_zip_code(crime(C2), Z)

num_of_crimes_in_zip_code(crime(C), N) :- findall(C1, same_zip_code(crime(C),
crime(C1)), L), length(L, N)
```

nota: se stranamente più vittime di uno stesso crimine fossero legate a diversi zip code, il crimine risulterebbe collocato in più zip code, preservando comunque la semantica relativa al "numero di crimini in un dato zip code".

Clausole per proprietà di community area

Per quanto riguarda le proprietà inferite per un crimine dalla community area, si riporta l'esempio della feature income

```
crime_area_income(crime(C), I) :- comm_area(crime(C), COM), comm_income(COM,
I)
```

Sono definite clausole analoghe per tutte le altre proprietà di un individuo community area.

Clausole per proprietà di vittime/arresti

Allo stesso modo, clausole che seguono lo schema precedente permettono di recuperare proprietà associate alle vittime o agli arresti: c'è da tener conto, però, che le istanze ground delle teste di queste clausole potranno essere molteplici per un dato crimine. Si veda l'esempio per l'etnia della vittima (*similmente a questo, vengono definite clausole per il sesso della vittime e per l'etnia dell'arresto (arrestato)*).

```
crime_victim_race(crime(C), VR) :- victimization(crime(C), victim(V), T),
victim_race(victim(V), VR)
```

Per casi di questo tipo, per cui possono essere vere più clausole ground per un dato crimine (es. due vittime di due etnie diverse), lo script python, a seguito della query "*crime_victim_race(crime(hz334455), VR)*", effettua un controllo sul numero dei risultati.

Un caso particolare è il calcolo della feature booleana "*is_ratial(crime(C))*", vera se esistono una vittima e un arrestato per il crimine C le cui etnie sono diverse

```
is_ratial(crime(C)) :- crime_arrested_race(crime(C), PR),
crime_victim_race(crime(C), VR), dif(VR, PR)
```

Clausole per il calcolo di medie

Un altro caso particolare riguarda l'età delle vittime: per poter utilizzare tale valore anche per crimini con più vittime si è scelto di calcolare e attribuire la media delle età.

```
aver_age(crime(C), Avg) :- findall(A, (victimization(crime(C),
victim(V), T), victim_age(victim(V), A)), L), sumlist(L, Sum),
length(L, Length), Length > 0, Avg is Sum / Length
```

sumlist/2 calcola la somma dei valori della lista L delle età delle vittime del crimine C; notiamo che, sapendo che per ogni crimine esiste sempre almeno una vittima, la condizione Length > 0 è ridondante.

Clausole per calcolare il numero di occorrenze

In generale, per calcolare un numero di occorrenze di qualche elemento, la sintassi utilizzata è la seguente (nell'esempio si calcola il numero di vittime di un crimine):

```
num_of_victims(crime(C), N) :- findall(V, victimization(crime(C), victim(V), T), L), length(L, N)
```

Nota: la stessa clausola con "homicide" sostituito a T permette di calcolare il numero di morti.

Altre clausole

Le altre clausole aggiunte rientrano negli schemi precedenti o richiedono l'aggiunta di semplici espressioni

- "num_of_arrest(crime(C), N)": N è il numero di arresti del crimine C
- "is_homicide(crime(C))": vero se esiste una vittima la cui vittimizzazione è "homicide"
- "night_crime(crime(C))": vero se l'orario in cui il crimine è avvenuto è tra le 20 e le 6.
- "immediate_arrest(crime(C))": vero se esiste un arresto collegato al crimine C che è registrato alla stessa data e ora
- "same_month_arrest(crime(C))": vero se esiste un arresto collegato al crimine C che è registrato nello stesso mese e anno in cui è avvenuto il crimine
- "is_there_a_child(crime(C), T)": vero se al crimine C è associata una vittima la cui età è inferiore o uguale a 15 e la vittimizzazione corrispondente è T
- "is_killed_a_child(crime(C))": corrisponde a "is_there_a_child(crime(C), homicide)"
- "crime_by_ group(crime(C))": vero se ci sono almeno 2 arresti per il crimine C
- "avg_num_charge(crime(C), Avg)": calcolo simile a quello per calcolare la media delle età però usando il predicato
    - `has_arrest(crime(C), arrest(A)), num_of_charges(arrest(A), NC)`

Query

Per preparare il dataset contenente tutte le feature inferite, si effettua la query corrispondente a ogni feature che si intende calcolare, sostituendo opportunamente la variabile C in "crime(C)" con il codice di un crimine.

- Per le feature booleane, basta controllare se la lunghezza della risposta è 0: se è così, la feature avrà valore falso, altrimenti avrà valore vero. Questo perché possono essere restituiti molteplici risultati tutti uguali tra loro perché dimostrabili seguendo rami diversi di computazione.
- Per feature categoriche, se la query corrispondente può generare più risultati *differenti*, il modo per trattarli viene definito caso per caso via codice python.
- Per altre feature si è certi che le query corrispondenti avranno esattamente un solo risultato (es. query di conteggio `num_of_crimes_in_district`).

Classificazione omicidi

Introduzione

Considerando sia le feature originarie dei dataset che quelle ingegnerizzate, si è individuato un possibile target per un task di classificazione: IS_HOMICIDE per un dato crimine è

- VERO se almeno una vittima è stata uccisa
- FALSO altrimenti

L'obiettivo è cercare di capire quali feature (eliminando quelle immediatamente collegate, scontati predittori, i.e. NUM_OF_DEAD) contribuiscano nel poter distinguere gli omicidi dagli altri crimini.

Sono stati condotti esperimenti utilizzando diversi modelli di classificazione, operando modifiche e variazioni ai dataset di riferimento dove necessario: nei paragrafi a seguire verranno illustrate le tecniche relative ai vari classificatori utilizzati.

Nota: ci sono 52 valori nulli sul campo "AVER_AGE" → si è deciso di eliminare gli esempi incompleti.

Approcci seguiti

A causa della presenza di tipi diversi di features di input, si sono adottati tre approcci (utilizzando la libreria python **Scikit Learn**) :

- Le feature relative alle coordinate geografiche sono state utilizzate per eseguire l'algoritmo **k-Nearest-Neighbors**, per scoprire se crimini avvenuti nei luoghi meno distanti da un dato crimine ne condividono la natura.
- Causa il grande numero di features inferite automaticamente, non sapendo quali di queste possano risultare utili per il task, si è scelto di sperimentare modelli di classificazione basati su alberi di decisione, in modo tale da effettuare automaticamente feature selection.
    - In Scikit Learn gli alberi di decisione supportano unicamente feature numeriche, per cui non si è potuto utilizzare molti dei valori originari dei dataset, essendo categorici (senza alcuna relazione d'ordine definita). Tutte le feature inferite automaticamente sono invece numeriche (o in particolare booleane, mappate su valori 0/1), per cui utilizzando queste si è sperimentato l'apprendimento di **DecisionTreeClassifier**, **RandomForest**, **AdaBoost** e **GradientBoostingClassifier**
- Le feature categoriche sono state invece utilizzate per l'addestramento di classificatori **Naive Bayes Categorical**, adatti al caso in quanto per ogni feature considerano separatamente i possibili valori per il calcolo delle probabilità condizionate, non richiedendo che sia definita una relazione d'ordine.

L'apprendimento di tutti i modelli ha seguito lo schema di 10-fold cross-validation e le metriche di valutazione riportate sono tutte mediate sui 10 modelli appresi.

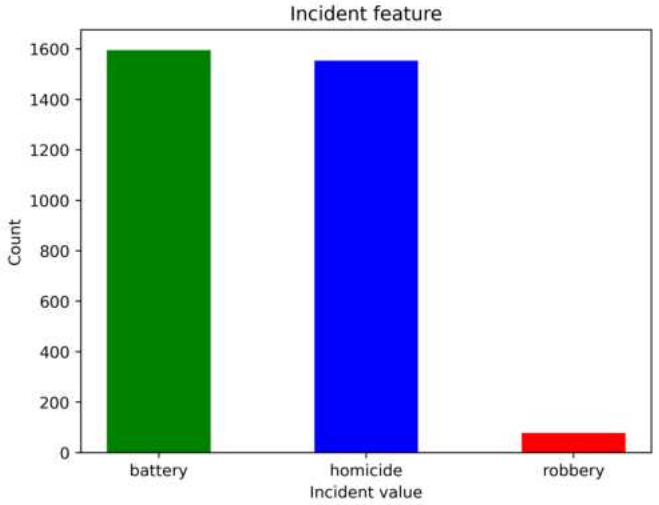
- In alcuni casi si è studiato l'andamento delle prestazioni dei modelli al variare di qualche iperparametro: per ogni valore dell'iperparametro è stata effettuata la cross-validation ed è stata calcolata la media delle prestazioni. In seguito si è calcolata la media delle medie delle prestazioni per ogni valore dell'iperparametro.
- C'è da compiere una precisazione: siamo coscienti del fatto che inserire l'intero dataset nella knowledge base falserebbe i valori di alcune features (ad esempio il numero di crimini in uno stesso distretto)

- Si sarebbero dovuti inserire unicamente i dati di training e aggiungere singolarmente le istanze di test per calcolarne i valori derivati e subito rimuoverle
- Sebbene ciò si sarebbe potuto fare per uno split tra dati di training e dati di test, dal momento in cui si utilizza la 10-fold cross-validation (sfruttando **KFold** con shuffle=True), sarebbe stato proibitivo in termini di tempo dover effettuare questo processo per ogni addestramento di un classificatore.
- Si è scelto conseguentemente di calcolare un'unica volta i valori delle feature derivate e costruire un unico dataset che ne contenga i valori, tenendo conto dei valori falsati per le feature interessate (nel caso in cui queste avessero dimostrato potere predittivo sul target). Si noti che le uniche feature con questo problema sono quelle relative al numero di crimini nella stessa unità geografica.

Descrizione target

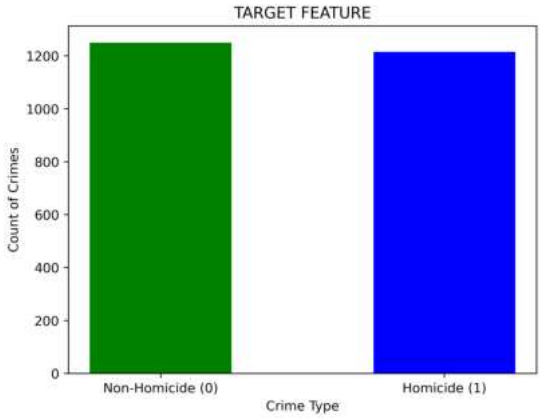
La feature target IS_HOMICIDE è stata derivata sulla base dei campi VICTIMIZATION_PRIMARY e INCIDENT_PRIMARY (in realtà si è sfruttato il primo tra i due anche se, coerentemente con la semantica, si sarebbe potuto in alternativa usare il secondo), per cui si effettuerà un confronto tra INCIDENT_PRIMARY e IS_HOMICIDE.

INCIDENT_PRIMARY: si mostra come sono distribuiti i valori, precisando che contiene duplicati, ovvero per ogni crimine con più di una vittima il valore del campo sarà ripetuto per ogni vittima. Il diagramma a barre si può interpretare come "*il numero di vittime che ha preso parte a un crimine di un determinato tipo*". Dire "il numero di vittime uccise" è diverso da dire "il numero di vittime che ha preso parte a un omicidio", in quanto in quest'ultimo caso non tutte sono necessariamente state uccise (ne basta anche solo una).



Per questo motivo, si è scelto di definire la feature IS_HOMICIDE (già menzionata nella sezione relativa alla creazione della knowledge base), con lo scopo di distinguere unicamente gli omicidi dagli altri crimini e di evitare la ripetizione spiegata.

```
is_homicide(crime(C)) :- victimization(crime(C), victim(V), homicide)
```



Il diagramma a barre per visualizzare la distribuzione dei valori di IS_HOMICIDE ci permette di concludere che i valori della feature target sono quasi perfettamente bilanciati (la percentuale di omicidi rispetto al totale ammonta al 49.29 %), quindi l'**accuratezza predittiva** risulta un buon indicatore delle performance del sistema. Per completezza, si riportano anche i valori delle metriche **precision**, **recall** e **F1-score**, calcolate mediante (precision_recall_fscore_support): tutti i valori sono mediati eseguendo una *macromedia*.

K-NN

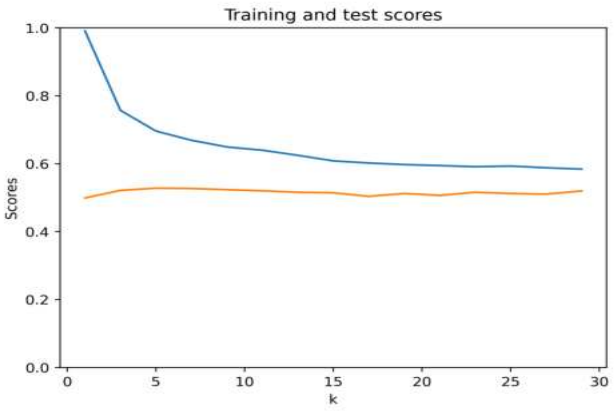
Innanzitutto, si è sperimentato l'utilizzo dell'algoritmo **k-Nearest-Neighbors** sfruttando i dati geografici relativi a latitudine e longitudine in cui un caso è stato registrato. Si prova a prevedere la feature target *IS_HOMICIDE* sulla base delle distanze (euclidee) tra i diversi crimini, per scoprire se crimini avvenuti in poco distanti da un dato crimine ne condividono la natura.

Quindi, il K-NN è stato sostanzialmente eseguito sul dataset avente 3 features.

- Features di input: *Latitude*, *Longitude*.
- Features di output: *IS_HOMICIDE*.

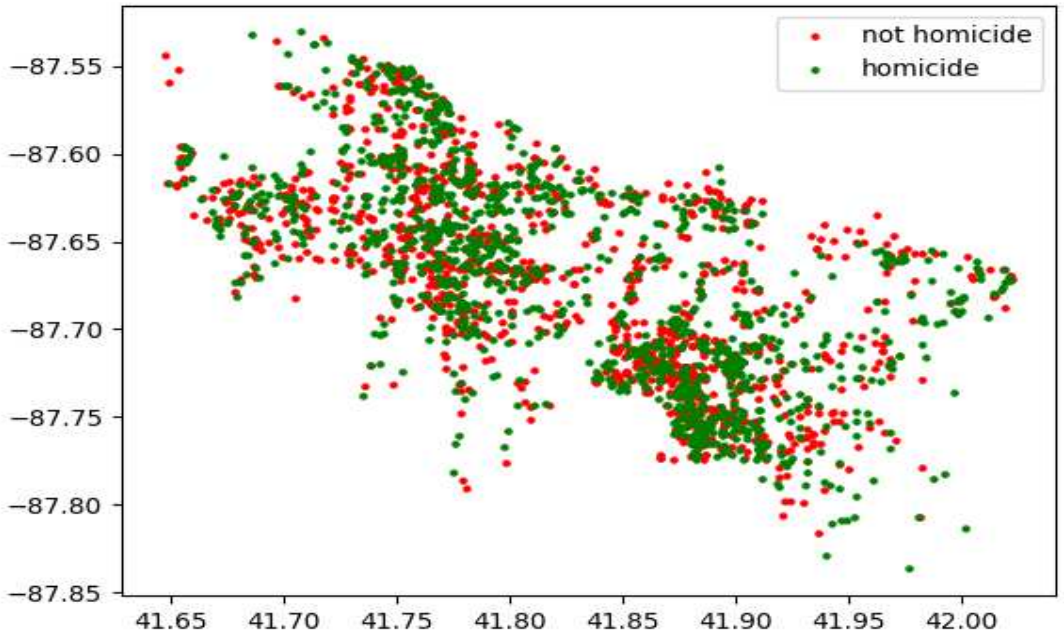
Sempre con l'uso della 10-fold Cross-Validation, si sono misurate le prestazioni del classificatore al variare del suo unico parametro, *k*.

Accuratezza	**0.51547049**
Precisione	**0.51525443**
Richiamo	**0.51515696**
F1-score	**0.51341652**



La tabella riporta la metriche mediate sui valori k dispari nel range (1,29). Sul grafico, invece, si possono vedere le prestazioni del modello al variare del parametro k (la linea azzurra corrisponde all'andamento dell'accuratezza sui dati di training, la linea gialla sui dati di test).

Infine, si illustra una mappa geografica rappresentante le coordinate (latitudine e longitudine), sulla quale sono raffigurati i crimini: i punti in verde individuano luoghi geografici degli omicidi, i punti rossi invece si riferiscono agli altri crimini.



Conclusioni: il K-nn, utilizzando le coordinate geografiche, si è rivelato completamente inefficace, di poco superiore a un classificatore "random", per cui si decide di passare all'approccio successivo.

Alberi di decisione

Si riportano di seguito i grafici relativi all'accuratezza (che corrisponde allo "score" per il classificatore, così come documentato da Scikit Learn) su dati di training (linea azzurra) e sui dati di test (linea gialla). Si sono misurate le performance al variare della profondità dell'albero: sperimentando con i criteri di selezione (per effettuare lo split) disponibili (**log_loss**, **gini**, **entropy**) non si sono notate differenze significative tra le prestazioni. D'ora in poi si sottointenderà l'utilizzo del criterio "entropy".

Accuratezza	**0.52752392**
Precisione	**0.53038027**
Richiamo	**0.52760289**
F1-score	**0.52120575**

Feature importance (di seguito si riportano le feature più importanti derivate dall'addestramento degli alberi, l'importanza di tutte le feature somma a 1).

Nota: si riportano unicamente i valori ottenuti, senza però trarre alcuna conclusione

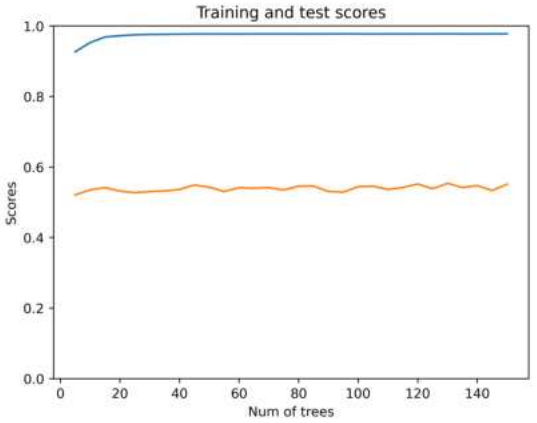
1) AVER_AGE: 0.2574052392928358
2) NUM_CRIMES_ZIP_CODE: 0.1763175335981358
3) IMMEDIATE_ARREST: 0.11501429778129887
4) AREA_HIGH_SCHOOL_DIPLOMA: 0.11115482621726087

Conclusioni: alberi di decisione inefficaci, leggermente migliori di un classificatore "random". All'aumentare della profondità tendono unicamente a overfittare i dati.

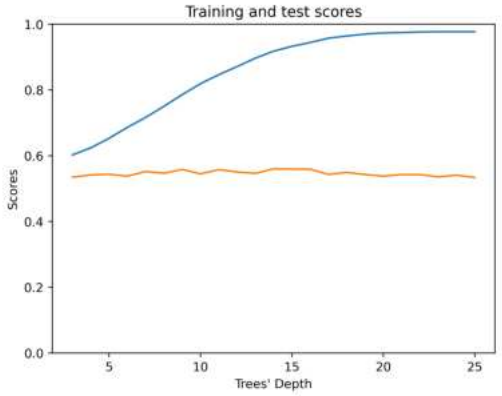
Random Forest

Si è sperimentata la tecnica di bagging RandomForest al variare del numero di alberi appresi (con profondità massima 5) e, constatato che non vi è incremento nell'accuratezza all'aumentare del numero di alberi, si è sperimentato anche l'andamento dell'accuratezza predittiva al variare della profondità massima degli alberi (utilizzando stavolta il valore di default per il numero di alberi appresi).

Accuratezza	**0.53753990**
Precisione	**0.53793883**
Richiamo	**0.53787979**
F1-score	**0.53650870**



Accuratezza	**0.54634568**
Precisione	**0.54774865**
Richiamo	**0.54703350**
F1-score	**0.54413878**



Feature importance

1) AVER_AGE: 0.15040813610904263
2) NUM_CRIMES_ZIP_CODE: 0.08758953749509481
3) NUM_CRIMES_BEAT: 0.06555026074977488
4) AVG_NUM_CHARGES: 0.062152501429210416

Conclusioni: anche le Random Forest risultano inefficaci, con un esiguo miglioramento rispetto agli alberi di decisione.

AdaBoost

Si è sperimentato sulla base del numero di weak learner (alberi) utilizzati nell'apprendere il modello di boosting.

Accuratezza	**0.54134223**
Precisione	**0.54229974**
Richiamo	**0.54175583**
F1-score	**0.53923085**

Feature importance

1) AVER_AGE: 0.25
2) NUM_CRIMES_ZIP_CODE: 0.15
3) AREA_ASSAULT_HOMICIDE: 0.1

Conclusioni: anche AdaBoost risulta inefficace, con l'unica differenza di aver riscontrato un overfitting decisamente inferiore rispetto ai casi precedenti (come si nota dalla distanza inferiore tra la predizione sui dati di training e quella sui dati di test).

Gradient Boosting

Anche in questo caso si è sperimentato sulla base del numero di weak learner (alberi) utilizzati nell'apprendere il modello di boosting.

Accuratezza	**0.54718848**
Precisione	**0.54900675**
Richiamo	**0.54763002**
F1-score	**0.54338716**

Feature importance

1) AVER_AGE: 0.29841941913382397
2) NUM_CRIMES_ZIP_CODE: 0.1286656936547603
3) AREA_HIGH_SCHOOL_DIPLOMA: 0.08334142736880222
4) IMMEDIATE_ARREST: 0.07733120351799719

Conclusioni: anche il Gradient Boosting risulta inefficace, mostrando un overfitting maggiore rispetto all'AdaBoost all'aumentare del numero di alberi.

Naive Bayes Categorico

Per l'addestramento di Naive Bayes Categorici si è utilizzata una combinazione di feature binarie e categoriche (si sono opportunamente evitate quelle numeriche continue, inappropriate per il modello in questione). Le feature utilizzate sono:

- Location Description, Domestic, Beat, District, Ward, Community Area → *non inferite dalla KB*
- NUM_OF_ARREST, IS_DOMESTIC, NIGHT_CRIME, HAS_STREET_ORGANIZATION, IS_RATIAL, ARRESTED_RACE, VICTIM_RACE, VICTIM_SEX → *inferite dalla KB*

Sono stati utilizzati dei LabelEncoder per trasformare i valori di alcune feature in interi incrementali (da 0 a n-1, con n il numero di valori distinti per la feature): questo è necessario in quanto richiesto da CategoricalNB.

Sui dati di training l'accuratezza media è 0.78269112, mentre sui dati di test

Accuratezza	Precisione	Richiamo	F1-score
0.71360225	**0.71391229**	**0.71245490**	**0.71209337**

Le prestazioni sono nettamente superiori rispetto ai casi precedenti, per cui si è deciso di effettuare ulteriori prove:

1. Si effettua una prova per ogni feature f
    - si addestra sul dataset, avendo rimosso la sola feature f
    - si confrontano gli score di training e test ottenuti con quelli già presentati
2. Si cercano le feature che fanno variare maggiormente gli score

Di seguito sono riportati i risultati delle prove

---| Dropping Location Description |---
Score train: 0.6423715627515624
Score test: 0.5355024521905137
---------| Dropping Domestic |-------
Score train: 0.7832317297412177
Score test: 0.7148168263059148
---------| Dropping Beat |-------
Score train: 0.7644806046314496
Score test: 0.7285737796649222
---------| Dropping District |-------
Score train: 0.7873340983741123
Score test: 0.7168477008656726
---------| Dropping Ward |-------
Score train: 0.787739036300562
Score test: 0.7204930713274743
---------| Dropping Community Area |-------
Score train: 0.7885058582916373
Score test: 0.7228843685197985

---------| Dropping NUM_OF_ARREST |-------
Score train: 0.7801670018460943
Score test: 0.7164132187880583
---------| Dropping IS_DOMESTIC |-------
Score train: 0.7834573002810792
Score test: 0.7152216846055099
---------| Dropping NIGHT_CRIME |-------
Score train: 0.7840430075367625
Score test: 0.7135923768144565
| Dropping HAS_STREET_ORGANIZATION |
Score train: 0.7848546713744848
Score test: 0.7204831967348013
---------| Dropping IS_RATIAL |-------
Score train: 0.7813384976294979
Score test: 0.7135726276291102
---------| Dropping ARRESTED_RACE |-------
Score train: 0.779986293470889
Score test: 0.7184589052368257

-----| Dropping VICTIM_RACE |-------
Score train: 0.781158276886517
Score test: 0.7107633060136269
-----| Dropping VICTIM_SEX |-------
Score train: 0.783592272817226
Score test: 0.7160198808465819
-----| Dropping AVER_AGE |-------
Score train: 0.7821496535169864
Score test: 0.7156347717323327
----| Dropping IMMEDIATE_ARREST |---
Score train: 0.782375102148792
Score test: 0.7115647937855897
----| Dropping SAME_MONTH_ARREST |---
Score train: 0.7830514277261995
Score test: 0.719298245614035

Si conclude quindi che la feature "Location Description" è l'unica la cui assenza determina una grande differenza rispetto ai risultati ottenuti nella prova iniziale.

A questo punto si è provato a fare l'opposto: partendo da un dataset contenente la sola feature "Location Description", si addestra (sempre 10-fold cross-validation) un classificatore, misurandone le

prestazioni e si esegue una ricerca per aggiungere feature (seguendo uno schema di tipo **Hill Climbing**, utilizzando l'accuratezza predittiva sul test come funzione da massimizzare):

Finché non sono state aggiunte tutte le altre feature (al dataset di lavoro):

1. Per ogni feature f non ancora aggiunta
    a. si addestra un classificatore dopo aver aggiunto la feature f
    b. si valuta il classificatore
    c. si rimuove la feature
2. Si aggiunge definitivamente la feature a cui è associata la valutazione massima
3. Si stampa il valore di accuratezza predittiva sul training e sul test

I risultati sono stati i seguenti (segue l'ordine dato dalla colonna sinistra verso destra)

Initial Score - only Location Description
TR: 0.7617758915440914
TE: 0.7537424706230869
---------| Adding ARRESTED_RACE |-------
Score train: 0.7644349703824377
Score test: 0.7541226424409994
---------| Adding NUM_OF_ARREST |-------
Score train: 0.7663280399500828
Score test: 0.7589776505052501
---------| Adding AVER_AGE |-------
Score train: 0.7684021836171013
Score test: 0.7577992824462657
---------| Adding VICTIM_RACE |-------
Score train: 0.7709717819422472
Score test: 0.7594549224844476
------| Adding IMMEDIATE_ARREST |----
Score train: 0.7742167712163702
Score test: 0.7634590698133701
---------| Adding Domestic |-------
Score train: 0.7731804714672162
Score test: 0.763516671603963
| Adding HAS_STREET_ORGANIZATION |
Score train: 0.7739461150137491
Score test: 0.7650768572463053
------| Adding IS_DOMESTIC |-------
Score train: 0.7741714214194892
Score test: 0.7646933938975018
-------| Adding NIGHT_CRIME |-------
Score train: 0.7722334490511692
Score test: 0.7545768737039598
----| Adding SAME_MONTH_ARREST |--
Score train: 0.7716924210980585
Score test: 0.7582288272275436
---------| Adding IS_RATIAL |-------
Score train: 0.7740818596342514
Score test: 0.759841677364142
---------| Adding VICTIM_SEX |-------
Score train: 0.7680401979624287
Score test: 0.7440192883710214
---------| Adding District |-------
Score train: 0.7634441626562303
Score test: 0.7419917053421546
-------| Adding Community Area |---
Score train: 0.7655626808556808
Score test: 0.7330453243803694
---------| Adding Ward |-------
Score train: 0.7641203663255814
Score test: 0.7282150027978013
---------| Adding Beat |-------
Score train: 0.7821503443293045
Score test: 0.7209127415160

Conclusioni: da sola la feature Location Description dimostra un alto potere predittivo; aggiungendo man mano le feature seguenti si ha un esiguo incremento sia sul train che sul test. Si è deciso di provare a sfruttare l'informazione fornita da Location Description anche per i metodi basati sugli alberi di decisione, come spiegato nel prossimo paragrafo.

Modifica della KB

Sulla base dei risultati ottenuti dal Naive Bayes, si è scelto di ingegnerizzare ulteriori feature derivate da "Location Description" e di riprovare ad addestrare i modelli basati su alberi visti precedentemente.

Non è possibile effettuare un encoding numerico di Location Description in quanto non è definito un ordinamento tra i vari valori (conta 87 valori diversi): una possibilità consiste nell'effettuare il one-hot-encoding dei valori della variabile, però ciò presenterebbe diversi problemi

1. Creerebbe un numero eccessivo di feature (87), ognuna delle quali avvalorata a 0 per la maggior parte degli esempi
2. Ci sono molti valori sintatticamente distinti ma dalla stessa (o molto simile) semantica (es. per indicare un parcheggio ci sono 6 valori diversi): sarebbe opportuno unire ogni insieme di questi in un unico valore
3. Con feature di questo tipo, un albero può discriminare solo pochi valori alla volta e i due rami risulterebbero sbilanciati. Servirebbe anche aumentare la profondità massima, rendendo gli alberi più proni all'overfitting.

La soluzione adottata consiste nel modificare la knowledge base ingegnerizzando separatamente degli individui di tipo “location(X)”, con X il valore del campo Location Description, e raggrupparli sulla base di categorie stabilite in base alla semantica (dei diversi luoghi possibili) e in base alla frequenza di ciascun valore nel dataset

- Alcuni valori specifici sono trattati separatamente in quanto molto frequenti, altri sono aggregati con valori simili in quanto poco frequenti
- I valori meno frequenti in assoluto, non collegati direttamente a nessuna delle categorie individuate, sono stati semplicemente ignorati

Procedendo con due livelli diversi di astrazione, si è deciso di mappare ogni location in 4 possibili categorie di appartenenza:

- veicoli → rientrano sia veicoli privati che luoghi correlati ai trasporti pubblici
- luoghi pubblici → rientrano parchi, parcheggi, negozi/bar/pub, benzinai
- all’aperto → rientrano strade, vicoli, marciapiedi
- luoghi residenziali → rientrano appartamenti, case (house), residence, luoghi all’aperto facenti parte di residence (es. cortile)

Sono state quindi ingegnerizzate le seguenti clausole

```
is_vehicle(location(L)) :- is_private_vehicle(location(L));
is_public_vehicle(location(L))

is_public_place(location(L)) :- is_parking(location(L));
is_store_pub(location(L)); is_gas_station(location(L));
is_park(location(L))")

is_outside(location(L)) :- is_street(location(L)); is_sidewalk(location(L));
is_alley(location(L))

is_residential(location(L)) :- is_apartment(location(L));
is_house(location(L)); is_residence(location(L));
is_residential_outside(location(L))")
```

Queste corrispondono ai 4 predicati booleani relativi alle 4 categorie presentate (livello più alto di astrazione). Sono state costruite immaginando che potessero diventare delle potenziali scelte che permettessero agli alberi di partizionare i dati in rami meno sbilanciati rispetto all’uso di sole feature più specifiche.

I predicati che compaiono nei corpi di queste clausole corrispondono alle feature booleane di livello di astrazione più basso: sono stati aggiunti separatamente i fatti che contengono istanze ground di tali predicati, corrispondenti ai diversi valori selezionati tra quelli assunti da Location Description.

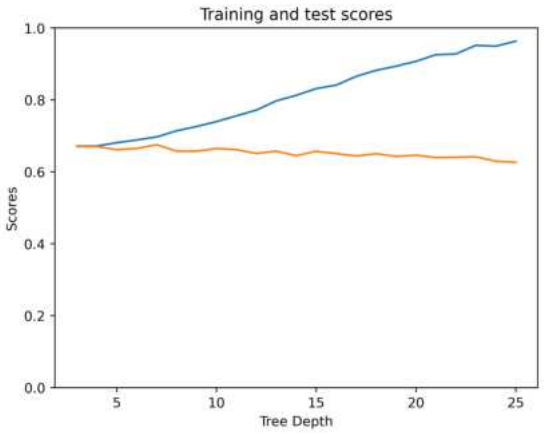
Per associare tali valori a ciascun crimine, per ognuno dei predicati presentati ne è stato creato un ulteriore sulla base del seguente schema (esempio “vehicle”)

```
location_vehicle(crime(C)) :- location_description(crime(C), location(L)),
is_vehicle(location(L))
```

Aggiunte queste nuove feature booleane al dataset sul quale si erano precedentemente addestrati gli alberi e gli ensemble, si presentano i risultati dell’addestramento degli stessi metodi sul nuovo dataset arricchito.

Alberi di decisione

Accuratezza	**0.65252689**
Precisione	**0.67633451**
Richiamo	**0.65396290**
F1-score	**0.64428360**



Feature importance

1) LOCATION_sidewalk:0.5837180332930592
2) LOCATION_residence: 0.19567166882171724
3) LOCATION_house: 0.13642557948846434

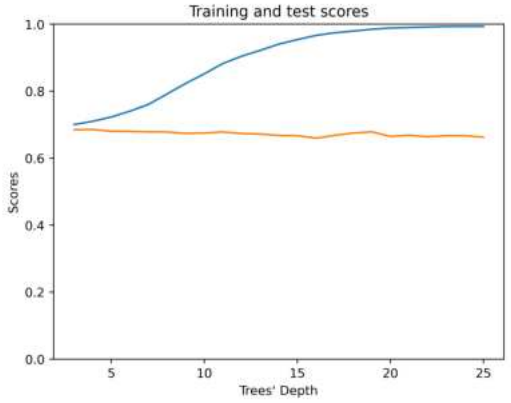
Random Forest

Al variare del numero di alberi nella Random Forest (profondità utilizzata: 5)

Accuratezza	**0.66404435**
Precisione	**0.66584113**
Richiamo	**0.66468948**
F1-score	**0.66293256**

Al variare della profondità degli alberi nella Random Forest (valore di default per il numero di alberi)

Accuratezza	**0.67266812**
Precisione	**0.68786897**
Richiamo	**0.67478913**
F1-score	**0.66739453**

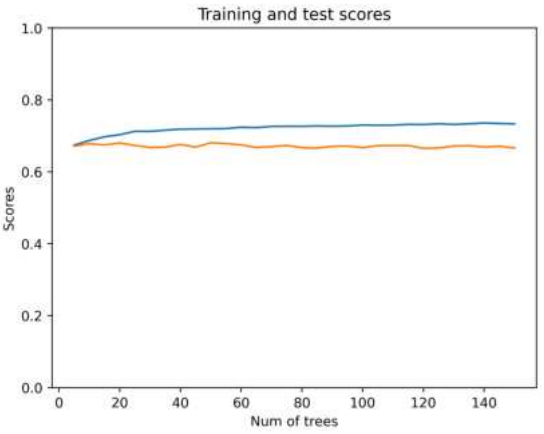


Feature importance

1) LOCATION_sidewalk: 0.3705258148043648
2) LOCATION_residence: 0.12010560354083842
3) LOCATION_house: 0.10391421630561001

AdaBoost

Accuratezza	**0.67176832**
Precisione	**0.68538446**
Richiamo	**0.67359043**
F1-score	**0.66699319**



Feature importance

1) AREA_UNEMPLOYMENT: 0.15
2) LOCATION_sidewalk: 0.1
3) *tutto il resto dell'importanza è spartita tra diverse feature, ognuna con peso 0.05*

Gradient Boosting

Accuratezza	**0.67550848**
Precisione	**0.71602729**
Richiamo	**0.67846172**
F1-score	**0.66214402**

Feature importance

1) LOCATION_sidewalk: 0.6180708829283752
2) LOCATION_residence: 0.21321236321674422
3) LOCATION_house: 0.10215237272131088

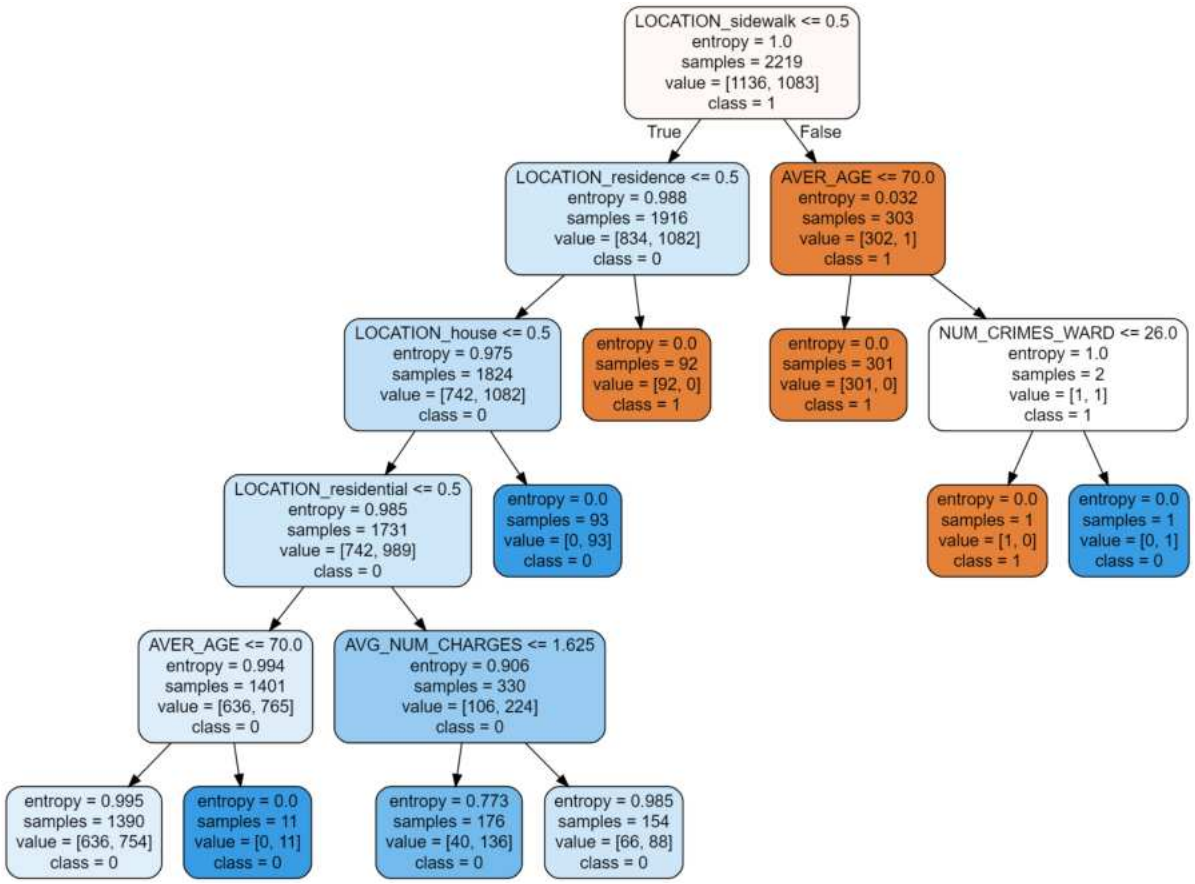
Conclusioni

Il miglioramento evidente è dato dall'aggiunta delle nuove informazioni e, in particolare

- l'AdaBoost sembra comportarsi meglio rispetto agli altri modelli (minore overfitting)
- il Gradient Boosting presenta un picco rispetto agli altri per quanto riguarda la precisione (71.6%)

Ciononostante, si ritiene che tali feature (concettualmente) mal si adattano agli alberi: viene infatti "sprecata" una condizione per controllare un singolo valore.

Si riporta l'immagine di un albero appreso automaticamente: si può notare come il ramo sinistro sia dominato da condizioni sul luogo del crimine e come in generale la maggior parte degli esempi cade sulla foglia all'estrema sinistra (tutti i crimini il cui luogo è diverso da *sidewalk*, *residence*, *house* e di categoria diversa rispetto a *residential*). Feature siffatte provocano lo "sbilanciamento" dell'albero.



Confronto con Naive Bayes Categorico

- Le performance di un singolo albero di decisione (il più semplice tra i modelli basati su alberi) sono decisamente più basse rispetto al Naive Bayes categorico addestrato sulla sola feature Location Description (0.65 di accuratezza contro 0.75). Stesso discorso per modelli ensemble più complessi che superano di poco le performance dell'albero.
- Incrementare la profondità massima dell'albero non aiuta, portando invece a overfitting. Anche in questo il naive bayes vince, con un overfitting quasi assente.

La semplicità del Naive Bayes gioca ulteriormente a suo favore, aprendo come possibile sviluppo futuro, per migliorare nel task, la creazione di un meta-learner (*stacking*) che permetta di combinare predizioni date dal Naive Bayes (sfruttando questa feature categorica) con predizioni di modelli diversi che lavorino con feature continue.

Un'altra possibilità riguarda la modifica del modello Naive Bayes categorico in modo tale da considerare dipendenze tra le variabili: ad esempio, un **tree augmented Naive Bayes Classifier**.

Clustering

Una sezione a parte è stata dedicata a un esperimento indipendente dal task di classificazione presentato: per alcune vittime assassinate sono presenti nome e cognome (in tutto 1328), per cui si è cercato il modo per ottenere informazioni aggiuntive tramite essi.

- Si è pensato che, automatizzando ricerche sul web per recuperare notizie di cronaca su ogni vittima, sarebbe stato possibile ricavare ulteriori informazioni utili o distintive di qualche insieme di crimini.
- Ci si è chiesti se queste eventuali informazioni potessero eventualmente essere connesse a qualche feature dei dataset di lavoro (Ad esempio, “è possibile che news riguardanti omicidi che coinvolgono vittime di etnie diverse si distinguano dalle altre?”).

Per effettuare un’analisi esplorativa dei dati, non caratterizzata da alcun target esplicito di cui predire il valore, si è utilizzato un approccio non supervisionato:

1. **Embedding**: data una vittima di omicidio per cui si conoscono nome e cognome, tramite la libreria **GoogleNews** è stato possibile automatizzare la ricerca

   “<*nome*> <*cognome*> chicago homicide”

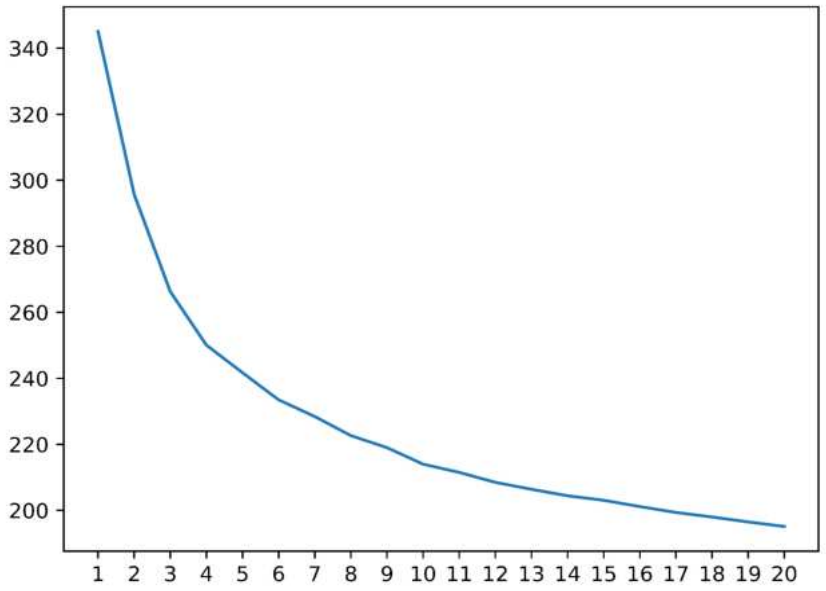
   recuperando e concatenando i primi 3 risultati in un unico “documento”, trasformato in un vettore embedding a 300 dimensioni utilizzando l’algoritmo **Word2Vec** (è stata usata la libreria **Gensim** ed è stato sfruttato un modello pre-addestrato, ‘word2vec-google-news-300’).

2. **Clustering:** Ricavati gli embedding per ogni vittima, è stato usato l’algoritmo **KMeans** per trovare cluster di vittime.

Valutazione cluster

Per determinare il valore più appropriato di k si è utilizzata l’ *elbow rule*: il grafico presentato considera sull'asse delle ordinate la somma delle distanze di ogni elemento dal centroide del proprio cluster e sulle ascisse il numero di clusters (k).

- Come è possibile visualizzare dall’immagine, il ‘gomito’ della curva può essere considerato in punti differenti: sarebbe possibile scegliere il valore di k come 4 o 6.
- Tuttavia, si è scelto k=4 clusters perché l’altra scelta avrebbe portato cluster eccessivamente piccoli (come si può notare dalla tabella sottostante).

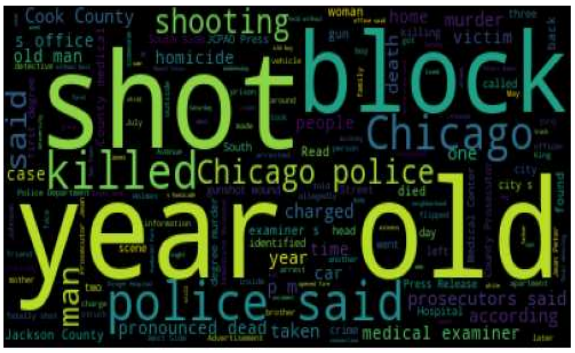


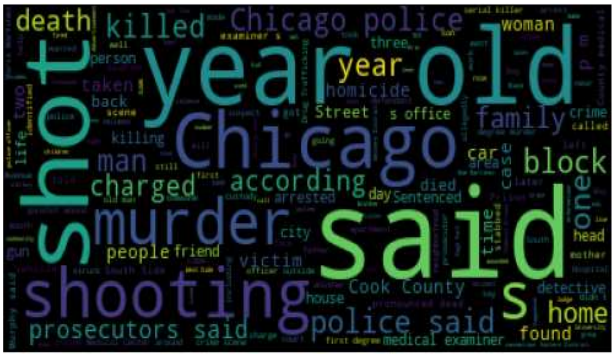
Di seguito il numero di dati per ogni cluster, al variare di k:

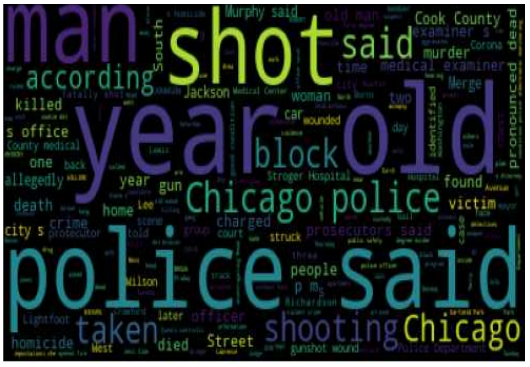
K	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
4	509	606	521	53	-	-	-	-	-	-
6	448	42	519	262	9	263	-	-	-	-
10	27	425	201	40	139	214	1	1	199	326

Si è optato per la scelta di test statistici (**One-Way Anova**, il quale testa l'ipotesi nulla che due o più popolazioni abbiano la stessa media) per verificare se differenze nella media tra valori di feature per popolazioni associate a diversi cluster fossero statisticamente significativi. I risultati tuttavia non ci hanno permesso di raggiungere alcuna conclusione.

Infine, un'ultima idea è stata adottata per poter visualizzare i termini maggiormente utilizzati nei documenti di cluster diversi: è stata generata una WordCloud per ogni cluster, sulla base della concatenazione di tutti i documenti corrispondenti agli esempi del cluster (anche in questo caso nessuna differenza evidente).







Conclusioni

Non è stato possibile determinare differenze significative tra i cluster trovati dall'algoritmo K-means. Tuttavia, ulteriori sviluppi possono essere i seguenti:

- attingere news da portali specifici, per assicurarsi dell'affidabilità e della qualità dei documenti ritrovati;
- effettuare information extraction dai documenti ritrovati per poter ricavare valori di feature mancanti (es. età della vittima) o valori di nuove feature;
- sperimentare diversi algoritmi di clustering e diversi modelli per mappare dei documenti in uno spazio vettoriale.