Beatrice Fumagalli 784549
Matteo Porcino 748876
Nicolò Monti 769709
Pierluigi Tagliabue 835211

APPLICAZIONE DI ALGORITMI DI MACHINE LAERNING PER IL MIGLIORAMENTO DELLA DATA QUALITY SUI DATABASE DEL BALTIMORA POLICE DIPARTMENT

Abstract

Storicamente, la città di Baltimora è tra le dieci più pericolose degli Stati Uniti. Come è tipico delle grandi città, la criminalità continua a dilagare nei quartieri della città. Il potenziale dei Big Data è quello di sfruttare in maniera più efficace informazioni già note, scoprire i legami tra fenomeni diversi e prevedere quelli futuri.
Essendo venuti a conoscenza di questa immensa risorsa, anche i governi locali hanno deciso di mettere a disposizione interi dataset riguardanti diversi aspetti della vita della propria città attraverso siti Open Data. Lo stesso governo di Baltimora ha reso disponibili diversi database, tra i quali quelli riguardanti la criminalità del paese.

Abbiamo provato a integrare due database: il primo riguardante i crimini commessi nella città di Baltimora e il secondo circa gli arresti nella medesima, migliorandone la qualità, con il fine di creare un dataset che fosse il più idoneo possibile ad una successiva applicazione di predizione dei crimini futuri. Il nostro report presenta l'intero processo da noi compiuto per ottimizzare i dati a nostra disposizione e creare una solida base di Machine Learning per successive applicazioni a favore della polizia di Baltimora.

Indice

Introduzione

Baltimora, situata sulla costa orientale degli Stati Uniti d'America, è la più grande città dello stato del Maryland. Fondata nel 1729 in una posizione strategica per il commercio, è cresciuta molto rapidamente, diventando un importante scalo marittimo. Dopo la seconda guerra mondiale si è assistito ad un boom economico che ha portato al progressivo spostamento del ceto medio verso la periferia con un conseguente ed altrettanto progressivo degradamento del centro. Questo andamento, nonostante i numerosi interventi attuati dalla municipalità nel corso degli anni, non è ancora stato pienamente risolto. Ad oggi, secondo Forbes, Baltimora è la settima città più pericolosa degli Stati Uniti, e seconda dopo a Detroit tra i centri con più di 500.000 abitanti. La popolazione della città conta un totale di 626.849 cittadini mentre la possibilità di essere vittima di un crimine violento è del 17,95% su un campione di 1000 residenti, percentuale molto più elevata rispetto al 4,72% del Maryland e alla media nazionale pari al 4%. La popolazione, formato dal 47% da uomini e dal 53% da donne, è così etnicamente composta: 28,1% popolazione bianca, 3.8% popolazione ispanica, 63,2% popolazione nera, 2,4% popolazione asiatica, 2% popolazione mista e 0,5% altre etnie.

1. Scopo dell'analisi

Lo scopo del nostro progetto è stato quello di vedere come le tecniche forniteci durante le lezioni, avessero un riscontro effettivo su un dataset reale, con tutti i problemi che sarebbero potuti sorgere e se poi effettivamente fosse possibile predire, con un certo grado di accuratezza, i valori mancanti degli attributi dei nostri dataset.

2. Dataset e preprocessing

2.1 I Dataset selezionati

Il governo di Baltimora mette a disposizione numerosi dataset riguardanti diversi aspetti della città sul sito "Open Baltimore". Abbiamo utilizzato due database provenienti da quest'ultimo. Il primo database fa riferimento ai crimini commessi nella città dal 01/01/2012 al 02/09/2017. Fornisce informazioni circa il luogo del crimine commesso: distretto, quartiere, indirizzo, se è avvenuto all'interno o all'esterno di un edificio, coordinate (longitudine e latitudine), distanza dalla caserma di polizia più vicina al luogo del crimine; il tempo: orario in cui è avvenuto il crimine; la tipologia di crimine commesso e l'arma utilizzata. Il secondo database fa riferimento agli arresti dei 130mila criminali e fornisce i dati riguardanti l'età, il sesso, l'etnia, il luogo dell'arresto, il crimine commesso, il distretto, la distanza dalla caserma di polizia più vicina al luogo dell'arresto, il quartiere, l'indirizzo dove ha avuto luogo il crimine con le relative coordinate (latitudine e longitudine), la data dell'arresto, l'orario e il codice identificativo del reato.

2.2 Crimes Dataset

CRIMES DATASET: 276529 rows		
	COLUMN TYPE	MISSING VALUES
Row ID		
CrimeDate	String	0
CrimeTime	String	0
CrimeCode	String	0
Location	String	2207
Description	String	0
Inside/Outside	String	10279
Weapon	String	180952
Post	Number (Int)	224
District	String	80
Neighborhood	String	2740
Geo_coordinates	String	2204
Premise	String	10757

2.2.1 Feature Selection

Tra i numerosi attributi del dataset abbiamo ritenuto opportuno eliminare i seguenti tre: "Total incidents", "Latitude" e "Longitude" dal database dei crimini in quanto il primo non era utile al fine del progetto e i due seguenti erano già presenti all'interno dell'attributo "Location1", successivamente rinominato come "Geo_coordinates".

2.2.2 Preprocessing

Abbiamo ritenuto opportuno manipolare i dati al fine di renderli più coerenti per la successiva fase di analisi.
CrimeDate: con il nodo *String Manipulation* abbiamo cambiato il formato della data da MM/dd/yyyy a yyyy-MM-dd.
Inside/Outside: utilizzando il nodo *String Replace* abbiamo abbreviato i valori dell'attributo *Inside/Outside* in I/O.
District: con il nodo *String Replace Dictionary* abbiamo creato un dizionario per normalizzare i valori dell'attributo *district.*
Neighborhood e Premise: con il nodo *Case Converter* abbiamo trasformato in maiuscolo tutti i valori degli attributi. Con il nodo *String Replace Dictionary* abbiamo poi creato un dizionario per normalizzare i valori dell'attributo *Neighborhood* al fine di renderlo uguale in entrambi i dataset.
Geo_coordinates: partendo da una colonna contenente latitude e longitudine, abbiamo ritenuto opportuno separarle attraverso l'utilizzo dei seguenti nodi: *Cell Splitter* (abbiamo splittato longitudine e latitudine), *Column Rename* (abbiamo rinominato i nuovi attributi in Latitudine e Longitudine), *Column Filter* (abbiamo eliminato la precedente colonna *Geo_coordinates*), *String Manipulation* (abbiamo normalizzato le due nuove colonne rimuovendo i simboli in eccesso).

I restanti attributi non sono stati oggetto di manipolazione.

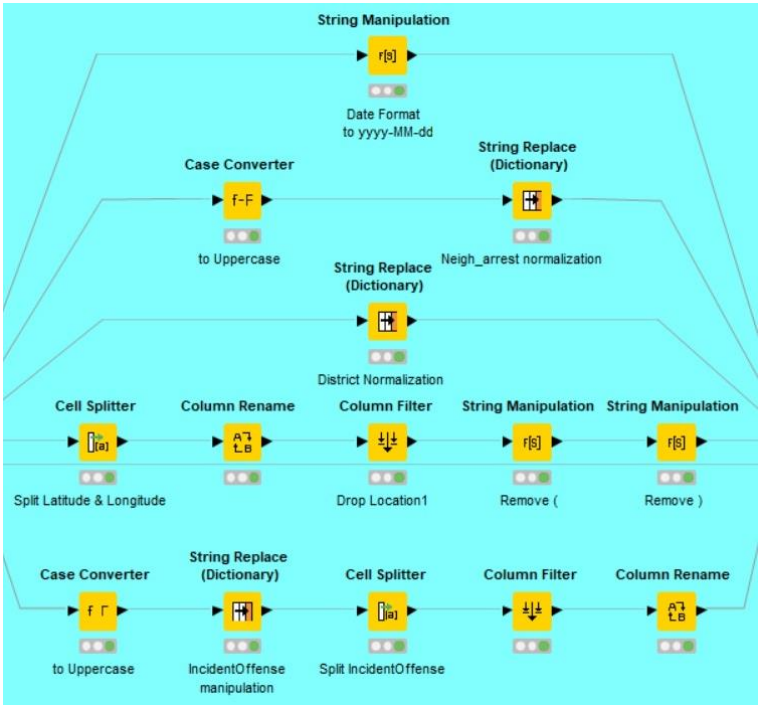
2.3 Arrest Dataset

ARREST DATASET: 130713 rows		
	COLUMN TYPE	MISSING VALUES
Row ID		
Arrest	Number (Int)	7014
Age	Number (Int)	28
Sex	String	0
Race	String	0
ArrestDate	String	0
ArrestTime	String	0
ArrestLocation	String	52118
IncidentOffense	String	0
IncidentLocation	String	53726
Charge	String	16458
ChargeDescription	String	502
District	String	52112
Post	Number (Int)	52130
Neighborhood	String	52118
Location 1	String	52047

2.3.1 Feature Selection

Tra i numerosi attributi del dataset abbiamo ritenuto opportuno eliminare *Charge* e *Arrest* in quanto non li abbiamo ritenuti utili al fine del progetto.

2.3.2 Preprocessing



ArrestDate: con il nodo *String Manipulation* abbiamo cambiato il formato della data da MM/dd/yyyy a yyyy-MM-dd.
IncidentOffense: con il nodo *Case Converter* abbiamo reso maiuscoli tutti i valori. Successivamente abbiamo utilizzato il nodo *String Replace Dictionary* al fine di normalizzare tutti i valori dell'attributo. Con *Cell Splitter* abbiamo splittato la colonna originale in due. Con *Column Filter* abbiamo rimosso la colonna di partenza e infine con *Column Rename* abbiamo rinominato le due nuove colonne in *CrimeCode* e *ArrestDescription.*
IncidentLocation, ChargeDescription, Neighborhood: con il nodo *Case Converter* abbiamo trasformato

in maiuscolo tutti i valori degli attributi. Successivamente con il nodo *String Replace Dictionary* abbiamo creato un dizionario per normalizzare i valori dell'attributo *Neighborhood* al fine di renderlo uniforme al dataset precedente.
District: con il nodo *String Replace Dictionary* abbiamo creato un dizionario per normalizzare l'attributo *district* al fine di renderlo uguale in entrambi i dataset.
Location1: partendo da una colonna contenente latitude e longitudine, abbiamo ritenuto opportuno separarle attraverso l'utilizzo dei seguenti nodi: *Cell Splitter* (abbiamo splittato longitudine e latitudine), *Column Rename* (abbiamo rinominato i nuovi attributi in Latitudine e Longitudine), *Column Filter* (abbiamo eliminato la precedente colonna *Location1*), *String Manipulation* (abbiamo normalizzato le due nuove colonne rimuovendo i simboli in eccesso).

2.4 Join

Al fine di creare un'unica tabella contente tutti gli attributi con relativi valori per la fase successiva di analisi, abbiamo fatto ricorso a singoli nodi *Joiner.* La scelta di creare più di una *Join* è stata dettata dalla necessità di avere dati il più coerenti e completi possibile. A tal proposito, abbiamo deciso di correlare l'attributo *CrimeDate, Primary key* del dataset *Crimes* con l'attributo *ArrestDate, Primary Key* del dataset *Arrest.* La struttura da noi creata in questa fase di progetto prevede per ciascun attributo un nodo *Joiner,* che analizzeremo successivamente, e una serie di nodi utilizzati per analizzare la differenza tra la longitudine e la latitudine rispettiva dei due differenti dataset, con il fine di uniformare le differenze minime di geolocalizzazione. Prevede inoltre un nodo *Java Snippet* utilizzato per il *match* tra gli attributi dei due dataset, al fine di valutare, attraverso valori *booleani* la corrispondenza o non corrispondenza tra di essi.

2.4.1 Joiner Date – Location

Abbiamo eseguito una *Join* tra gli attributi *Location* (nel dataset *Crimes*) e *IncidentLocation* (nel dataset *Arrest*) con lo scopo di valutare quali crimini descritti nell'attributo *IncidentLocation* fossero avvenuti nel medesimo luogo del crimine compiuto alla medesima data.

2.4.2 Joiner Date – Post

Abbiamo eseguito una *Join* tra gli attributi *Post* contenuti nel dataset *Crimes* e nel dataset *Arrest* con lo scopo di valutare quali crimini siano avvenuti nella stessa data riportando il medesimo codice della caserma di polizia più vicina al luogo dell'accaduto. Abbiamo successivamente deciso di escludere la *Join* in quanto troppo generica per il nostro scopo.

2.4.3 Joiner Date – Neighborhood

Abbiamo eseguito una Join tra gli attributi Neighborhood contenuti nel dataset Crimes e nel dataset Arrest con lo scopo di valutare quali crimini siano avvenuti nella stessa data e nel medesimo quartiere.

2.4.4 Joiner Date – District

Abbiamo eseguito una Join tra gli attributi *District* contenuti nel dataset Crimes e nel dataset Arrest con lo scopo di valutare quali crimini siano avvenuti nella stessa data e nel medesimo distretto.

2.4.5 Joiner Date – Coordinates

Abbiamo eseguito una Join tra gli attributi *Latitude* e *Longitude* contenuti nel dataset Crimes e nel dataset Arrest con lo scopo di valutare quali crimini siano avvenuti nella stessa data e nel luogo indicato dalle medesime coordinate geografiche.

2.4.6 Match

Dopo aver confrontato gli attributi dei due dataset, abbiamo effettuato un'analisi dei risultati ottenuti. In corrispondenza del *match* con risultato pari a 1, si ha una perfetta correlazione di tutti e tre gli attributi (*District, Location e Code*). Abbiamo ritenuto coerente validare anche i *match* con risultato maggiore o uguale allo 0,5, andando però ad effettuare un ulteriore *match* al fine di ottenere e di inserire nella tabella finale solamente le righe con una completa corrispondenza. Per effettuare i *match* abbiamo utilizzato il nodo *Java Snippet,* nella

cui scrittura abbiamo confrontato i diversi attributi servendoci del metodo ".equals" e utilizzando la seguente formula matematica:

$$Match = \frac{\text{numero degli attributi che } matchano}{\text{numero degli attributi che dovrebbero } matchare}$$

3. Metodologie e problemi affrontati

Viene riportata la lista dei classificatori che abbiamo cercato di utilizzare e i problemi riscontrati e affrontati.

3.1 I classificatori

Nel progetto sono stati utilizzati i seguenti classificatori e successivamente validati con metodologia *Cross Validation* e *Hold Out*.

- Modelli di regressione e separazione: Support Vector Machines (SMO).
- Modelli Probabilistici: Naive Bayes Multinomial Text (NBMT), Naive Bayes Tree (NBT).
- Modelli Euristici: Random Forest con alberi e features di *default* per iterazione (RFOR).

3.2 Problemi affrontati

Il principale problema riscontrato è stato legato alla difficoltà nell'individuare i modelli che fossero in grado di lavorare sugli attributi dei dataset da noi selezionati e dalle dimensioni di quest'ultimi (16mila righe * 21 attributi).
Feature selection:
Partendo dai dati originali e dopo una scrematura iniziale, abbiamo condotto una selezione degli attributi durante la stima dei classificatori utilizzando un ciclo di *Backward Feature Elimination*, che consiste nella selezione automatica e indipendente dagli altri, del miglior sottoinsieme di attributi dalla loro totalità.

Inoltre, per riuscire ad addestrare i modelli da noi scelti al fine di predire i *missing values* presenti all'interno dei dataset, abbiamo fatto ricorso ai nodi *Parallel Chunk Start/End*, che partiziona il dataset di input in sottoinsiemi, suddividendone le istanze.

Un'ulteriore problematica riscontrata riguarda la tipologia dei dati a disposizione. Trattandosi probabilmente di dataset creati manualmente da più persone e avendo la necessità di uniformare i dati al fine di creare un unico dataset, abbiamo dovuto utilizzare frequentemente il nodo *String Replacer Dictionary*, andando a creare un dizionario in formato *.txt* per omogeneizzare i valori degli attributi. Essendo però la maggior parte dei dati di tipo nominale, abbiamo dovuto utilizzare più volte il nodo *Domain Calculator* per aggiornare il dominio dell'attributo dei dataset.

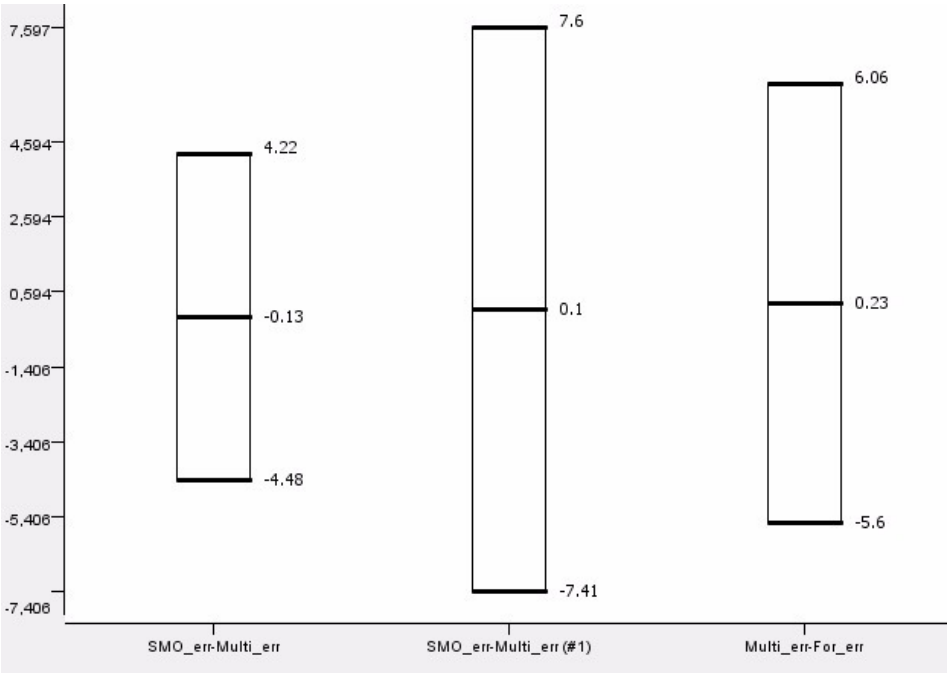
Avendo cercato di ottenere il maggior numero di righe usufruibili possibile nell'integrazione dei due dataset attraverso l'utilizzo di più nodi *Joiner,* già descritti, e potendo avere un possibile problema di duplicazione delle righe, abbiamo fatto ricorso al nodo *Java Snippet*. Mediante quest'ultimo, abbiamo confrontato riga per riga il dataset al fine di controllare se fossero presenti duplicati. Qualora si verificasse questa condizione, siamo intervenuti rimuovendo la riga in questione.

4. Risultati e commenti

Per l'utilizzo dei classificatori abbiamo sfruttato l'interfaccia *Weka* di *Knime.*

4.1 Premise

Per predire i *missing values* dell'attributo *Premise* del dataset ottenuto integrando i due di partenza, abbiamo utilizzato i seguenti modelli: Support Vector Machines (SMO), Naive Bayes Multinomial Text (NBMT) e Random Forest con alberi e features di *default* per iterazione (RFOR).

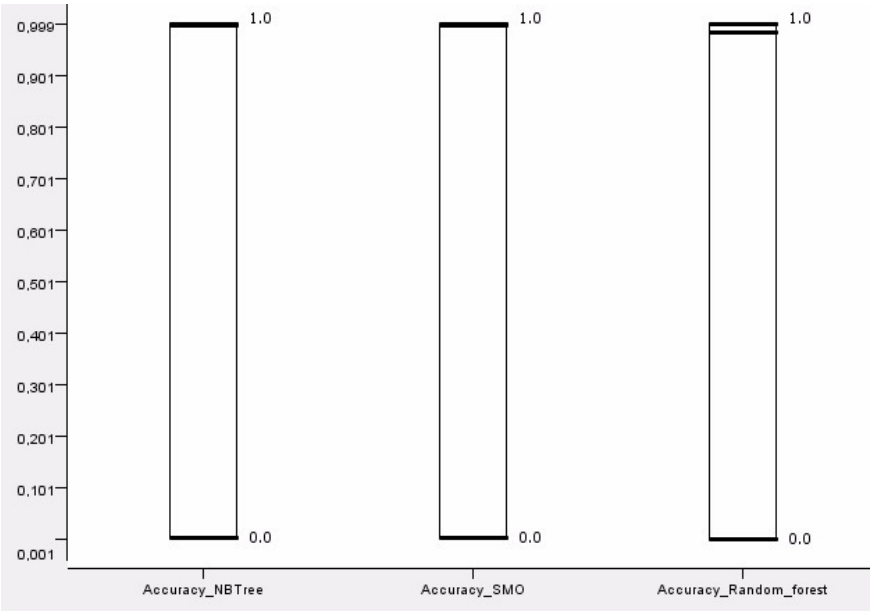


Premise	
	Acc
SMO	0.468
NBMT	0.336
RFOR	0.541

Osservando il grafico possiamo giungere alla conclusione che non esiste una sostanziale differenza statistica tra i diversi modelli. Pertanto, abbiamo deciso di utilizzare il modello RFOR per la predizione dell'attributo Premise in quanto presentava un grado di accuratezza maggiore rispetto gli altri due.

4.2 Inside/Outside

Per predire i *missing values* dell'attributo *Inside/Outisde* abbiamo utilizzato i seguenti modelli: Naive Bayes Tree (NBT), Support Vector Machines (SMO) e Random Forest con alberi e features di default per iterazione (RFOR).

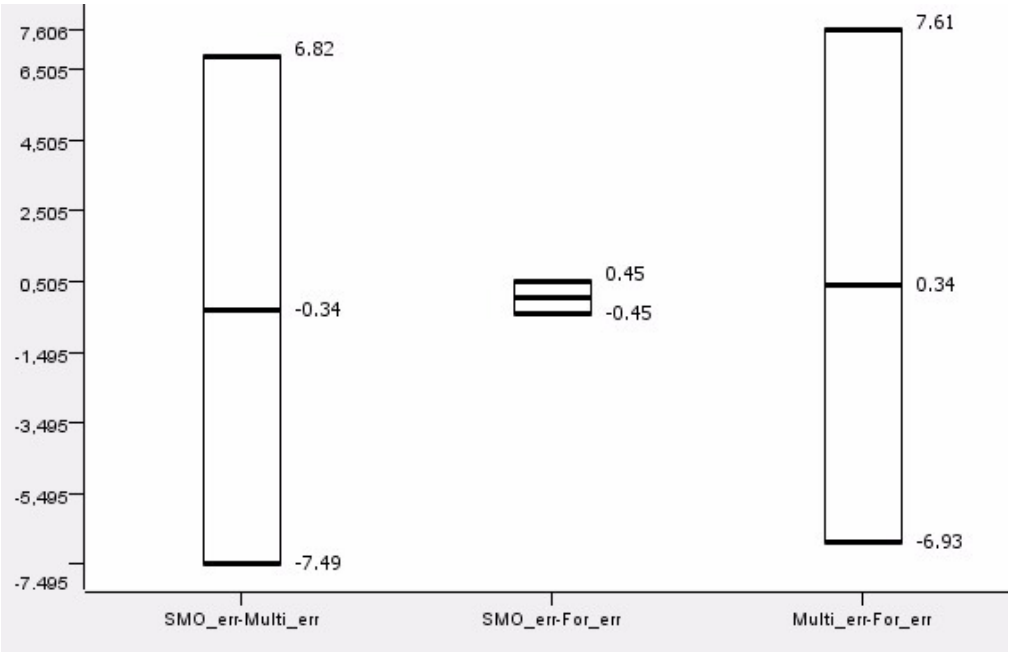


Inside/Outside			
	Acc	Rec	Prc
NBT	0.996	0.995	0.998
SMO	0.997	0.997	0.998
RFOR	0.983	0.997	0.978

Non essendoci una sostanziale differenza tra i risultati di *Accuracy, Recall e Precision* ottenuti dall'applicazione dei tre diversi modelli, abbiamo preferito utilizzare il modello *Naive Bayes Tree*, in quanto con le risorse a nostra disposizione l'addestramento di tale modello è risultato più performante in termini di risorse computazionali.

4.3 Weapon

Per predire i *missing values* dell'attributo *Weapon* del dataset abbiamo utilizzato i seguenti modelli: Support Vector Machines (SMO), Naive Bayes Multinomial Text (NBMT) e Random Forest con alberi e features di *default* per iterazione (RFOR).



Weapon	
	Acc
SMO	0.488
NBMT	0.496
RFOR	0.496

A prima vista, abbiamo scelto di eliminare il modello predittivo SMO, in quanto era quello che presentava il minor grado di accuratezza. Successivamente, a parità di grado di Accuracy, abbiamo deciso di adottare il modello RFOR, in quanto, anche se non fosse presente una differenza statisticamente rilevante, era quello che presentava una miglior classificazione dei valori dell'attributo predetto.

5. Conclusioni e prospettive

Nel corso del nostro progetto abbiamo voluto percorrere le principali fasi di manipolazione dei dati a cui dovrebbe essere sottoposto un dataset durante il processo di data cleansing, al fine di renderlo disponibile per successive analisi. Nello svolgimento di queste fasi, abbiamo potuto "toccare con mano", nonostante le dimensioni ridotte del campione da noi utilizzato, le medesime difficoltà che si troverebbe ad affrontare un team di Data Scientist all'interno del dipartimento della polizia di Baltimora.
Abbiamo pertanto cercato di affrontare i problemi sorti progressivamente nella maniera più professionale che ci fosse concessa dai mezzi e dalle conoscenze a nostra disposizione. Riteniamo che, al fine di raggiungere risultati migliori nell'integrazione di dataset differenti, sarebbe opportuno:

- Aumentare la data quantity inserendo un maggior numero di attributi per dataset con elementi rilevanti ai fini di una maggiore precisione dell'elaborazione.
- Aumentare la data quality, utilizzando dei metodi standard per l'inserimento dei dati nel database e riducendo così la libertà di interpretazione dei medesimi da parte dell'addetto.

Riteniamo pertanto di aver raggiunto il nostro scopo, ossia di aver uniformato al meglio i dati a nostra disposizione, predicendone, quando richiesto, i dati mancanti, generando così un database ordinato e pulito che possa essere un punto di partenza per analisi future. Nello specifico crediamo che il dataset da noi creato, sfruttando le potenzialità delle tecniche di machine learning, possa rappresentare una discreta base per la predizione dei crimini futuri di Baltimora.

References

https://www.kaggle.com/sohier/crime-in-baltimore
https://www.kaggle.com/arathee2/arrests-by-baltimore-police-department
https://data.baltimorecity.gov/
http://www.baltimoresun.com/news/maryland/crime/bal-police-codes-box-story.html

Appendice

Le variabili dei dataset.

Archivio Crimes:

CrimeDate: data in cui è stato commesso il crimine. Formato yyyy-MM-dd.
CrimeTime: orario in cui è stato commesso il crimine. Formato HH:mm:ss.
CrimeCode: codice identificativo del crimine.
Location: indirizzo del luogo in cui è avvenuto il crimine.
Description: descrizione del crimine.
Inside/Outside: I = il crimine è avvenuto all'interno di un edificio. O = il crimine è avvenuto all'esterno.
Weapon: arma utilizzata per compiere il crimine. Knife, Firearm, Hands, Other.
Post: codice della caserma più vicina al luogo del crimine.
District: distretto. SD = southern; ND = northern; CD = central; WD = western; ED = eastern; SW = southwestern; NW = northwestern; SE = southeastern; NE = northeastern.
Neighborhood: quartiere di Baltimora.
Geo_coordinates: coordinate geografiche, latitudine e longitudine.
Premise: tipologia del luogo in cui è avvenuto il crimine.

Archivio Arrest:

Arrest: codice dell'arresto.
Age: età del criminale. Compresa tra 0 e 87 anni.
Sex: genere del criminale. M = male; F = female.
Race: etnia del criminale. W = white; B = black; A = asian; I = indian; U = unknown;
ArrestDate: data dell'arresto. Formato yyyy-MM-dd.
ArrestTime: orario dell'arresto. Formato HH:mm.
ArrestLocation: indirizzo del luogo in cui è avvenuto l'arresto.
IncidentOffense: reato commesso.
IncidentLocation: indirizzo del luogo in cui è avvenuto il crimine.
Charge: codice dell'accusa.
ChargeDescription: tipologia di reato commesso.
District: distretto. SD = southern; ND = northern; CD = central; WD = western; ED = eastern; SW = southwestern; NW = northwestern; SE = southeastern; NE = northeastern.
Post: codice della caserma più vicina al luogo dell'arresto.
Neighborhood: quartiere di Baltimora.
Location 1: coordinate geografiche, latitudine e longitudine.