

ELABORATO DI MACHINE LEARNING & DATA MINING

CLASSIFICAZIONE DEI TIPI DI CRIMINE DI SAN FRANCISCO



Università degli Studi di Brescia

Dipartimento di Ingegneria dell’Informazione

Corso di Laurea Magistrale in Ingegneria Informatica



BETTINELLI LUCA 85947

MARIO MATTEO 86015

MASCIALE MICHELE 85982

# indice dei contenuti

[indice dei contenuti II](#_Toc447640765)

[Capitolo 1 – Descrizione del problema 1](#_Toc447640766)

[1.1 - Introduzione 1](#_Toc447640767)

[1.2 – Struttura del dataset 1](#_Toc447640768)

[1.3 – Strumenti utilizzati 2](#_Toc447640769)

[Capitolo 2 – Preprocessing dei dati 3](#_Toc447640770)

[2.1 – Fase preliminare 3](#_Toc447640771)

[2.2 – Attributo *Dates* 3](#_Toc447640772)

[2.2.1 – Scomposizione in Season e DailyRange 4](#_Toc447640773)

[2.2.2 – Scomposizione in *Year*, *Month*, *DayOfMonth*, *Hour* e *Minute* 5](#_Toc447640774)

[2.3 – Attributo *DayOfWeek* 5](#_Toc447640775)

[2.4 – Attributo *Address* 6](#_Toc447640776)

[2.4.1 – Creazione dell’attributo *isCross* 7](#_Toc447640777)

[2.4.2 – Creazione dell’attributo *streetType* 8](#_Toc447640778)

[2.4.3 – Riduzione dell’attributo *Address* 9](#_Toc447640779)

# Capitolo 1 – Descrizione del problema

## 1.1 - Introduzione

Dal 1934 al 1963, San Francisco era nota come patria di alcuni dei criminali più famigerati al mondo. Oggi la città è famosa più per la sua tecnologia che non per il suo passato criminale. Tuttavia, con la crescente disuguaglianza sociale, carenza di alloggi e proliferazione di mezzi tecnologici, tra cui BART (Bay Area Rapid Transit District), non vi è certamente scarsità di crimine. Dai distretti di Sunset a South of Market, da quelli di Marina a Excelsior, il dataset a nostra disposizione offre quasi 13 anni di notizie di reati da tutti i quartieri di San Francisco. Data l’ora e la posizione in cui si sono verificati i reati, è necessario prevederne la categoria. Si tratta di una delle competizioni accessibili a chiunque sul sito di Kaggle.

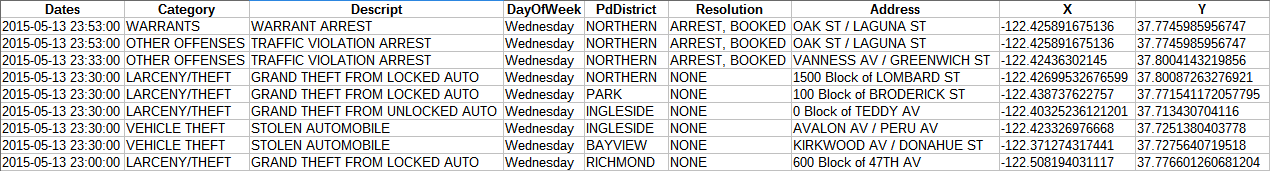
## 1.2 – Struttura del dataset

Il dataset a nostra disposizione è suddiviso in due file in formato CSV: il training set (circa 125 MB) e il test set (circa 89 MB). Questo dataset contiene le informazioni su reati e crimini registrati dal sistema di report dei dipartimenti di polizia di San Francisco. In particolare sono stati raccolti i dati del periodo che va dal 01/01/2003 al 13/05/2015 e sono stati successivamente distribuiti nei due file a settimane alterne: le settimane dispari appartengono al test set, quelle pari al training set.

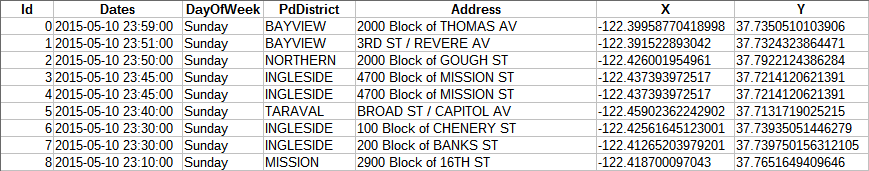
Di seguito l’elenco degli attributi che descrivono un crimine nel dataset:

* **Id**: un numero identificativo auto-incrementale presente solo nel test set;
* **Dates**: una stringa che rappresenta il timestamp, ovvero la data in cui è avvenuto il crimine;
* **DayOfWeek**: una stringa che rappresenta il giorno della settimana in cui è avvenuto il crimine;
* **PdDistrict**: una stringa che rappresenta il nome del distretto nel cui dipartimento di polizia si è registrato il crimine;
* **Address**: una stringa che rappresenta l’indirizzo della via in cui è avvenuto il crimine;
* **X**: un numero che rappresenta la longitudine;
* **Y**: un numero che rappresenta la latitudine;
* **Category**: una stringa che rappresenta la categoria del crimine, presente solo nel training set. Questa è proprio la variabile target da predire;
* **Descript**: una stringa che rappresenta una descrizione dettagliata del crimine, presente solo nel training set;
* **Resolution**: una stringa che rappresenta la modalità con cui il crimine è stato condannato, presente solo nel training set.

In *Figura 1.1* e *Figura 1.2* è riportato un esempio di dataset, estratto da entrambi i file:



*Figura 1.1: Un estratto del training set.*

**

*Figura 1.2: Un estratto del test set.*

## 1.3 – Strumenti utilizzati

Il linguaggio di programmazione utilizzato per questo progetto è Python nella sua versione a 64 bit per un uso meno limitato della memoria. Tale linguaggio vanta, tra le altre qualità, di offrire svariate librerie appositamente dedicate al Machine Learning e al preprocessing dei dati, come pure quelle rivolte al tracciamento di grafici. In particolare sono state utilizzate le seguenti librerie:

* **pandas**: dotata di funzioni per la lettura e scrittura di dataset;
* **sklearn**: dotata di funzioni che implementano numerosi algoritmi di Machine Learning, addestrando i relativi classificatori ed estraendo le predizioni su nuovi dati;
* **matplotlib**: dotata di funzionalità per la generazione e personalizzazione di grafici.

# Capitolo 2 – Preprocessing dei dati

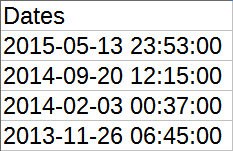
## 2.1 – Fase preliminare

La prima fase in assoluto è stata rivolta alla riorganizzazione degli attributi di training set e test set in modo che questi fossero comparabili, ovvero avessero gli stessi attributi su cui i classificatori potessero lavorare. In particolare sono stati rimossi dal training set gli attributi *Descript* e *Resolution*, decisamente ridondanti data la categoria del crimine; è stato inoltre rimosso l’attributo *Id* dal test set in quanto inutile ai fini della predizione e non presente in fase di addestramento.

Al termine di questa fase training set e test set risultavano avere gli stessi attributi ed è stato perciò possibile iniziare il vero e proprio processing dei dati.

## 2.2 – Attributo *Dates*

Il primo attributo che ha meritato la nostra attenzione è proprio quello del timestamp, ovvero la data in cui il crimine è avvenuto. Riportiamo per comodità un esempio estratto dal dataset in *Figura 2.1*.



*Figura 2.1: Esempi di valori per l’attributo Dates*

Come si può notare dalla *Figura 2.1* i valori assunti da questo attributo sono molto specifici per il crimine in quanto la precisione del timestamp raggiunge il campo dei minuti. Inoltre grazie a Weka è stato possibile osservare come il numero di valori distinti nel training set per questo attributo è 389.257, numero che sfiora la metà del totale delle istanze. Abbiamo dunque pensato di elaborare l’attributo *Dates*, sostituendolo con attributi meno specifici che portassero maggiori informazioni in fase di addestramento. I seguenti paragrafi mostrano le soluzioni da noi adoperate.

### 2.2.1 – Scomposizione in Season e DailyRange

La prima soluzione è stata quella di ridurre drasticamente il numero di possibili valori per la data, sostituendo l’attributo *Dates* con i nuovi attributi *Season* e *DailyRange*, che rappresentano rispettivamente la stagione e la fascia giornaliera in cui è avvenuto il crimine. In particolare per quanto riguarda la stagione, abbiamo elaborato il campo del mese del timestamp secondo questo schema approssimativo:

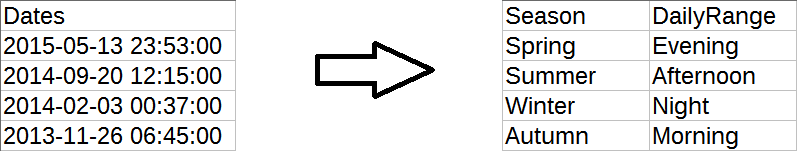
* mesi 1-3: corrispondono alla stagione invernale (*Winter*);
* mesi 4-6: corrispondono alla stagione primaverile (*Spring*);
* mesi 7-9: corrispondono alla stagione estiva (*Summer*);
* mesi 10-12: corrispondono alla stagione autunnale (*Autumn*).

Per quanto riguarda invece la fascia giornaliera, abbiamo preso in considerazione il campo dell’ora del timestamp, secondo le seguenti regole:

* ore 6-12: corrispondono alla fascia mattutina (*Morning*);
* ore 12-18: corrispondono alla fascia pomeridiana (*Afternoon*);
* ore 18-24: corrispondono alla fascia serale (*Evening*);
* ore 0-6: corrispondono alla fascia notturna (*Night*).

Come si può intuire il numero di possibili valori per la data è stato ridotto a 16, ovvero il numero di combinazioni possibili tra i 4 valori della stagione e i 4 valori della fascia giornaliera.

In *Figura 2.2* riportiamo come esempio la suddetta scomposizione di alcune date:

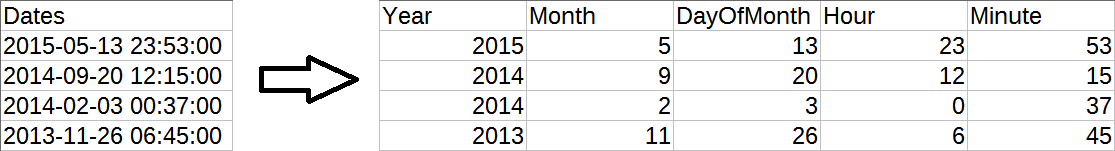


*Figura 2.2: Esempio di scomposizione in Season e DailyRange*

### 2.2.2 – Scomposizione in *Year*, *Month*, *DayOfMonth*, *Hour* e *Minute*

La seconda soluzione è invece stata quella di sostituire la data con un attributo per ogni campo del timestamp a nostra disposizione (ad eccezione dei secondi in quanto sempre nulli). In questo caso il numero di possibili combinazioni dei valori dei nuovi attributi nel caso peggiore è 6.963.840, in quanto si hanno 13 valori diversi per l’anno, 12 per il mese, 31 per il giorno, 24 per l’ora e 60 per il minuto. Nonostante ciò abbiamo pensato che per la fase di classificazione sarebbe stato più opportuno avere più attributi con meno valori disponibili per ognuno piuttosto che un solo attributo con moltissimi valori.

In *Figura 2.3* riportiamo come esempio la suddetta scomposizione di alcune date:

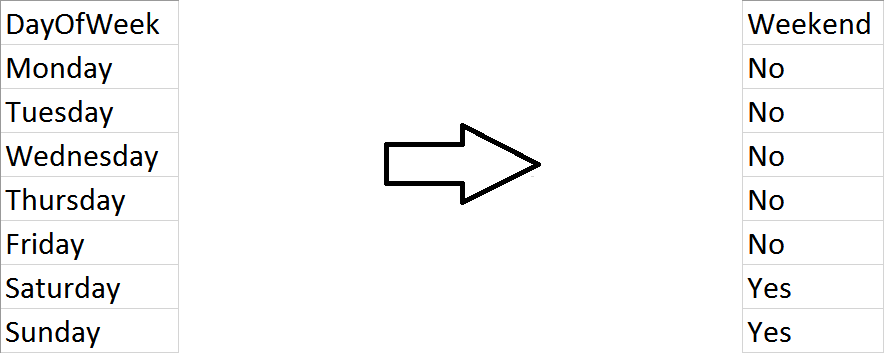


*Figura 2.3: Esempio di scomposizione in Year, Month, DayOfMonth, Hour e Minute*

## 2.3 – Attributo *DayOfWeek*

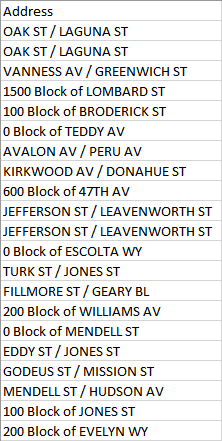
Per quanto riguarda il giorno della settimana, l’unico tipo di elaborazione a cui abbiamo pensato è stato quello di adottare una distinzione binaria tra i giorni del weekend e gli altri. In particolare abbiamo aggiunto un nuovo attributo, *Weekend*, che assumesse il valore *Yes* per i giorni *Saturday* e *Sunday*, e il valore *No* per tutti gli altri.

In *Figura 2.4* riportiamo un esempio di trasformazione dell’attributo in questione.



*Figura 2.4: Esempio di elaborazione dell’attributo DayOfWeek*

## 2.4 – Attributo *Address*

Un altro attributo che secondo noi andava rielaborato è quello che descrive testualmente il luogo di avvenimento del crimine, ovvero l'indirizzo. Riportiamo per comodità un esempio estratto dal dataset in *Figura 2.5*.

*Figura 2.5: Esempi di valori per l’attributo Address*

Come si può notare dalla *Figura 2.5* i valori assunti da questo attributo sono molto specifici

per il crimine in quanto rappresentano il nome di una via con il relativo numero civico; nel caso di un incrocio tra due strade, il valore assunto comprende i nomi, separati da "/", delle vie con il

relativo numero civico. Inoltre grazie a Weka è stato possibile osservare come il numero di valori distinti nel training set per questo attributo è 23.228, decisamente troppo elevato

per applicare un criterio di *splitting* delle istanze. Abbiamo dunque pensato di elaborare l'attributo

Address, creando nuovi attributi più generici che portassero maggiori informazioni in fase di addestramento.

I seguenti paragrafi mostrano le soluzioni da noi adoperate.

### 2.4.1 – Creazione dell’attributo *isCross*

La prima soluzione è stata quella di ridurre drasticamente il numero di possibili valori per l'indirizzo,

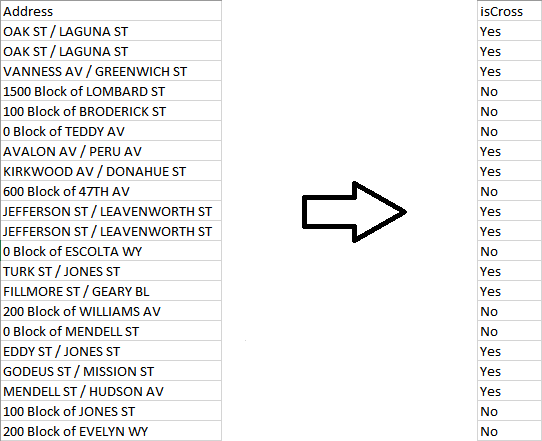
creando il nuovo attributo *isCross*, che rappresenta un valore booleano secondo la seguente semplice regola:

* *Yes*: corrisponde ad un incrocio tra due strade;
* *No*: corrisponde ad una sola strada.

Come si può intuire il numero di possibili valori per l'indirizzo è stato ridotto a 2, numero di gran

lunga inferiore rispetto a quello iniziale.

In *Figura 2.6* riportiamo come esempio la trasformazione di alcuni indirizzi.



*Figura 2.6: Esempio di creazione dell’attributo isCross*

### 2.4.2 – Creazione dell’attributo *streetType*

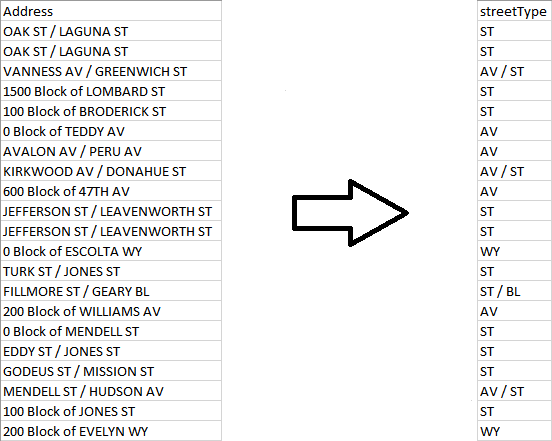
La seconda soluzione è invece stata quella di creare l'attributo *streetType* che rappresentasse il tipo

di strada in cui è avvenuto il crimine.

I valori assunti da questo nuovo attributo seguono il seguente schema:

* se l'indirizzo non è un incrocio: il valore assunto è il tipo di strada descritto nell'indirizzo;
* se l'indirizzo è un incrocio tra due strade dello stesso tipo: il valore assunto è il tipo delle strade dell'incrocio;
* se l'indirizzo è un incrocio tra due strade di diverso tipo: il valore assunto è rappresentato dai tipi delle due strade separati da "/".

Adoperando questa convenzione il numero di possibili valori per l'indirizzo è stato ridotto a 98, numero di gran lunga inferiore rispetto a quello iniziale.

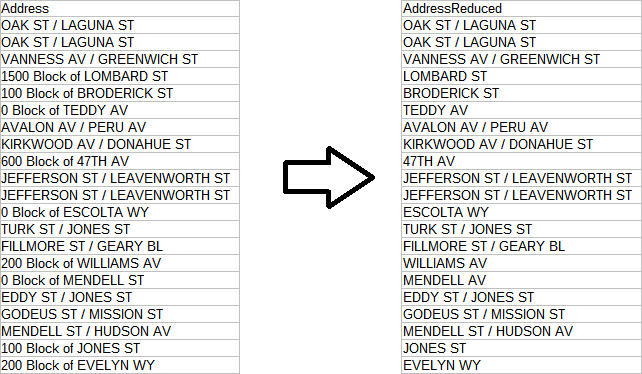
In *Figura 2.7* riportiamo come esempio la trasformazione di alcuni indirizzi.

*Figura 2.7: Esempio di creazione dell’attributo streetType*

### 2.4.3 – Riduzione dell’attributo *Address*

Un ulteriore tentativo di ridurre il numero di valori possibili per l'attributo *Address* è stato quello di mantenere solo il nome della via, rimuovendo l'eventuale numero civico. L'idea era quella di raggruppare i crimini in base alla via in cui sono stati commessi.

In *Figura 2.8* riportiamo come esempio la riduzione di alcuni indirizzi.



*Figura 2.8: Esempio di riduzione dell’attributo Address*