

Università degli studi di Bari facoltà di
scienze MM.FF.NN

Progetto Data Mining
NASA - Nearest Earth Objects hazard
detection

by

Vito Proscia mat. 735975



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Anno accademico 2022-2023

Contents

1	Introduzione	3
1.1	Contesto	3
1.2	Definizione obiettivo principale	3
1.3	Tool utilizzati	4
2	Analisi del dataset	4
2.1	Descrizione features	4
2.2	Preparazione dati	5
3	Machine Learning	7
4	Analisi esplorativa dei dati	7
5	Conclusioni	7

1 Introduzione

1.1 Contesto

[Near-Earth Objects](#) (NEO) dataset contiene una serie di informazioni, raccolte dalla NASA, che caratterizzano degli oggetti rilevati vicino alla terra, molti di questi oggetti sono a migliaia di chilometri dalla superficie terrestre, ma su scala astronomica queste distanze sono molto piccole e possono influenzare fenomeni naturali, quali per esempio cambiamenti nella marea, eventi sismici, cambiamento atmosferico, variazioni magnetiche e così via.

È importante sottolineare che la maggior parte degli corpi celesti che passano vicini alla Terra sono di piccole dimensioni e passano ad una distanza sicura, solitamente non hanno un impatto significativo sui fenomeni naturali, ma quelli di dimensioni maggiori o che si avvicinano molto possono avere degli effetti.

La natura dei Near-Earth Objects (NEO) si può dividere in:

- **Comete:** corpo celeste relativamente piccolo, composto da gas ghiacciati frammenti di rocce e metalli
- **Asteroidi:** corpi minori di un sistema planetario originati dallo stesso processo di formazione dei pianeti ma le cui fasi di accrescimento si sono interrotte più o meno presto, oppure formati attraverso la collisione tra altri corpi celesti, sono composti principalmente da silicati di nichel, ferro e magnesio

1.2 Definizione obiettivo principale

L'obiettivo principale del progetto è quello di addestrare un modello per andare a predire, in base ad alcuni parametri, quali corpi celesti rilevati attorno alla terra possono provocare danni, questo perchè è ormai ampiamente accettato dalla comunità scientifica che le collisioni di asteroidi con la Terra avvenute in passato hanno avuto un ruolo significativo nel disegnare la storia geologica e biologica del pianeta, per questo risulta interessante effettuare un task di classificazione binaria che coinvolge la feature *hazardous* con classi:

- **True:** oggetto potenzialmente pericoloso
- **False:** oggetto non pericoloso

1.3 Tool utilizzati

Per la sperimentazione sono stati usati diversi strumenti, quali:

- [Google Colab](#), strumento presente nella suite Google che consente di scrivere python notebook direttamente dal proprio browser, utilizzando risorse messe a disposizione da remoto.
- [Weka](#), software contenente una collezione di algoritmi per data Mining e apprendimento Automatico, scritto in Java e sviluppato presso University of Waikato New Zealand

2 Analisi del dataset

2.1 Descrizione features

Il dataset inizialmente si compone di 90836 osservazioni per dieci features che vanno a descrivere una serie di caratteristiche dei corpi celesti registrati, in particolare abbiamo:

1. *id* [numeric]: identificatore univoco per ogni oggetto
2. *name* [string]: nominativo dato dalla NASA
3. *est_diameter_min* [numeric]: diametro minimo stimato (Km)
4. *est_diameter_max* [numeric]: diametro massimo stimato (Km)
5. *relative_velocity* [numeric]: Velocità relativa rispetto alla terra (Km/h)
6. *miss_distance* [numeric]: ???
7. *orbiting_body* [string]: Corpo rispetto al quale l'oggetto sta orbitando
8. *sentry_object* [boolean]: Copro incluso o meno in sentry (sistema di monitoraggio automatico delle collisioni)
9. *absolute_magnitude* [numeric]: descrizione della luminosità dell'oggetto (energia radiata dal corpo al secondo)
10. *hazardous* [boolean]: Indica se il corpo è pericoloso o meno

2.2 Preparazione dati

2.2.1 Analisi delle input features

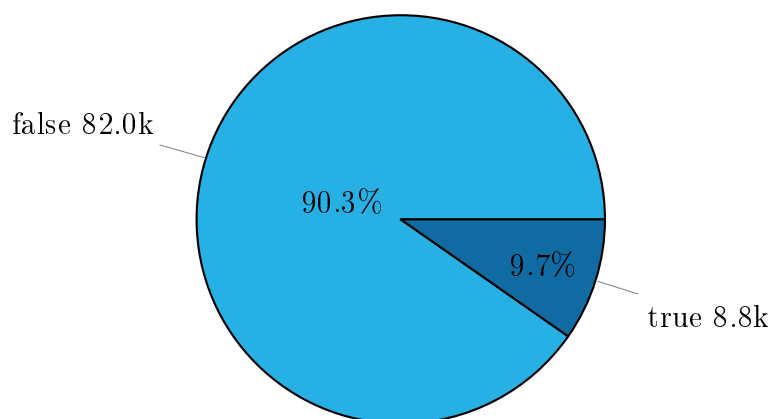
Andando a considerare direttamente il dataset come ci viene fornito ci sono una serie di problematiche legate ad alcune features, alcune di queste sono inutili per lo scopo di addestramento, quali:

- *id* (nessuna correlazione con la feature su cui fare predizione),
- *name* (nessuna correlazione con la feature su cui fare predizione),
- *orbiting_body* (ha un unico valore)
- *sentry_object* (ha un unico valore)

Un'altra considerazione si potrebbe fare sulle features *est_diameter_min* e *est_diameter_max*, andando a descrivere la dimensione di diametro massima e minima, si potrebbero accoppiare i dati delle due caratteristiche con un'unica che andrebbe a rappresentare la media matematica dei due valori (*est_diameter_mean*).

2.2.2 Analisi della target feature

Il "problema" più grande del lavoro riguarda la natura delle osservazioni inerenti alla target feature *hazardous*, che presenta una distribuzione di valori fortemente sbilanciata (90.3% per false e 9.7% per True)



2.2.3 Oversampling vs Undersampling

L'oversampling e l'undersampling sono due metodi per andare a risolvere lo sbilanciamento delle classi target, in particolare l'undersampling prevede il ridimensionamento di una classe prelevando da una popolazione un suo sottoinsieme, questo si applicherebbe alla classe più numerosa, mentre l'oversampling sposta il focus sulla classe con meno occorrenze andando a creare "sinteticamente" delle nuove osservazioni a partire da quelle già a disposizione.

2.2.4 Approccio ibrido

Nel nostro caso, come già accennato, le classi della target feature *hazardous* (true/false) sono molto sbilanciate (causa del fatto che per fortuna sono pochi i corpi celesti che si rivelano potenzialmente pericolosi), se da una parte i *false* superano gli 80k, dall'altra i *true* arrivano a malapena a 8k, quindi se applicassimo una tecnica di undersampling si avrebbe un'enorme perdita di informazioni, mentre se usassimo l'oversampling, per bilanciare i dati avremo moltissimi dati sintetici, per questo sarebbe meglio usare una tecnica ibrida.

Algoritmo: SMOTE-ENN

L'algoritmo SMOTE-ENN [Batista et al 2004] combina l'abilità di SMOTE (Synthetic Minority Oversampling) di generare istanze sintetiche per la classe minoritaria, con quella di ENN (Edited Nearest Neighbor) [Wilson 1972] di eliminare da entrambe le classi alcune osservazioni.

Algorithm 1 SMOTE-ENN

Input: Tr : Training set,

p : number of nearest neighbors in SMOTE,

k : number of nearest neighbors in ENN

Output: New_Tr : Training set after using SMOTE-ENN

 1. Divide Tr into positive and negative subsets:

$Tr \leftarrow Pos \cup Neg$;

 2. Oversampling the minority class using SMOTE to balance class distribution:

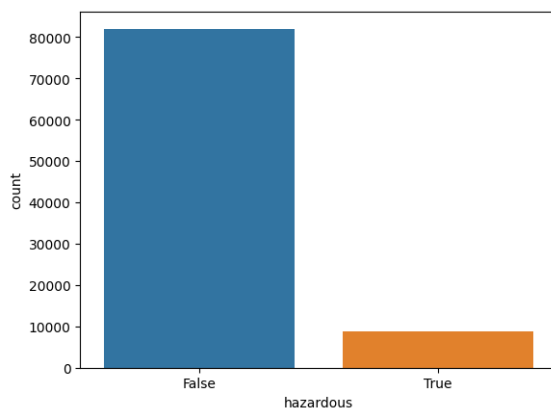
$New_Pos \leftarrow SMOTE(Pos, p)$;

$|New_Pos| = |Neg|$;

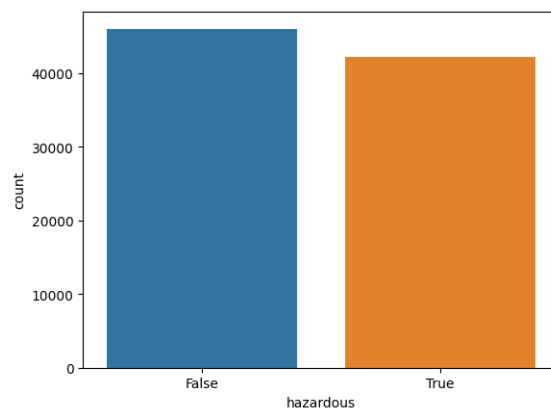
 3. $New_Tr \leftarrow New_Pos \cup Neg$;

 4. Remove noisy examples using ENN:

$New_Tr \leftarrow ENN(New_Tr)$;



(a) Prima di SMOTE-ENN



(b) Dopo SMOTE-ENN

3 Machine Learning

4 Analisi esplorativa dei dati

5 Conclusioni