UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO | DIPARTIMENTO DI INFORMATICA

Asteroid_Hazard_Detection

Documentazione sul caso di studio

Ingegneria della Conoscenza AA 2025-2026

Realizzato da: Matteo Lupis, 801159, m.lupis1@studenti.uniba.it

Repository: Asteroid_Hazard_Detection

Indice

1 Introduzione

La difesa planetaria, il monitoraggio e la classificazione dei *Near-Earth Objects* (*NEO*) rappresentano attività di elevata importanza per le agenzie spaziali internazionali. L'elevato numero di asteroidi che transitano in prossimità dell'orbita terrestre, unito alla varietà delle loro caratteristiche fisiche, rende necessario lo sviluppo di sistemi automatici in grado di stimare il livello di pericolosità associato a ciascun oggetto.

Il progetto "*Asteroid Hazard Detection*" ha come obiettivo la realizzazione di un sistema di classificazione automatica della pericolosità dei NEO, basato su una pipeline strutturata di preprocessing, addestramento di modelli di Machine Learning supervisionato e integrazione con un modulo di ragionamento logico basato su Prolog. Il sistema opera su un dataset reale fornito dalla NASA, contenente parametri orbitali e stime dimensionali di migliaia di asteroidi.

L'obiettivo finale è dimostrare come un approccio ibrido, che combina tecniche di apprendimento statistico e conoscenza simbolica, possa migliorare l'efficacia e l'interpretabilità dei sistemi di rilevazione della pericolosità degli asteroidi.

1.1 Requisiti funzionali

Per poter eseguire il progetto è necessario che siano installati Python e SWI-Prolog e che siano disponibili le seguenti librerie:

- **Imbalanced-learn**: serve per la gestione dello sbilanciamento delle classi tramite oversampling.
- **Matplotlib**: serve per la visualizzazione dei dati e permette la creazione di grafici delle prestazioni dei modelli.
- **Scikit-learn**: serve per implementare algoritmi di Machine Learning e preprocessing.
- **Numpy**: serve per la manipolazione di dati e l'elaborazione numerica.
- **PySWIP**: serve per l'integrazione di Prolog all'interno di Python.
- **Pandas**: serve per la gestione dei Dataset.

Infine, nella directory principale del progetto è presente un file requirements.txt, che contiene l'elenco completo delle dipendenze necessarie. L'installazione può essere effettuata eseguendo il comando: pip install -r requirements.txt.

Dopo aver installato Python, SWI-Prolog e le librerie necessarie, è sufficiente avviare il software dal file main.py.

2 Dataset

Il dataset utilizzato è fornito dalla NASA e contiene informazioni su migliaia di Near-Earth Objects (NEO), ognuno descritto da 40 feature che comprendono parametri orbitali, stime dimensionali, informazioni sulla velocità relativa e sulla distanza di avvicinamento alla Terra. Il dataset presenta un significativo sbilanciamento delle classi: circa l'83,9% degli asteroidi è classificato come non pericoloso (*Hazardous = 0*), mentre solo il 16,1% risulta potenzialmente pericoloso (*Hazardous = 1*).

Le feature originali del dataset includono, tra le altre:

- **Absolute Magnitude:** magnitudine assoluta dell'asteroide, inversamente correlata alla dimensione.
- **Estimated Diameter in KM (min/max)**: stima del diametro in chilometri, con *min* che indica la stima inferiore e *max* la stima superiore.
- **Relative Velocity km/s**: velocità relativa dell'asteroide rispetto alla Terra al momento del massimo avvicinamento.
- **Miss Distance (Astronomical)**: distanza minima di avvicinamento alla Terra durante il passaggio osservato, misurata in unità astronomiche.
- **Minimum Orbit Intersection Distance (MOID)**: distanza minima tra l'orbita dell'asteroide e quella della Terra.
- **Eccentricity, Inclination, Semi Major Axis**: parametri orbitali kepleriani, in particolare:
  - *Eccentricity*: parametro orbitale che misura la deviazione dell'orbita dalla circolarità.
  - *Inclination*: inclinazione dell'orbita rispetto al piano dell'eclittica, espressa in gradi.
  - *Semi Major Axis*: semiasse maggiore dell'orbita ellittica, espresso in unità astronomiche, rappresenta la dimensione media dell'orbita.
- **Jupiter Tisserand Invariant**: parametro che classifica il tipo di orbita, aiuta a distinguere gli asteroidi da oggetti di altro tipo.
- **Hazardous**: variabile target binaria che indica se l'asteroide è potenzialmente pericoloso.

2.1 Preprocessing

La fase di preprocessing trasforma il dataset grezzo **nasa.csv** in una forma strutturata e coerente con le esigenze dei modelli di Machine Learning e del successivo modulo di ragionamento logico.

Selezione delle feature: dal dataset originale a 40 colonne sono state selezionate 16 feature numeriche rilevanti per la classificazione, eliminando feature ridondanti (come ad esempio le stime dimensionali in unità diverse come miglia e piedi), identificativi non predittivi (*Neo Reference ID*, *Name*), variabili temporali non informative (*Close Approach Date*, *Epoch Date Close Approach*) e colonne costanti (*Orbiting Body*, *Equinox*). Il dataset risultante contiene 4687 osservazioni e 17 colonne (16 feature + target).

Normalizzazione della variabile target: la colonna *Hazardous*, originariamente in formato stringa (True/False), è stata convertita in formato binario intero (1/0) per compatibilità con i classificatori.

Preprocessing differenziato delle feature: le feature numeriche vengono preprocessate mediante una pipeline che include l'imputazione dei valori mancanti tramite mediana (*SimpleImputer*) e la normalizzazione tramite *StandardScaler*. L'intera fase di preprocessing è integrata nella pipeline di addestramento attraverso un *ColumnTransformer*, garantendo che le trasformazioni vengano apprese esclusivamente sui dati di training.

Suddivisione dei dati: il dataset viene suddiviso in training set (80%) e test set (20%) mediante split stratificato, preservando la distribuzione della variabile target Hazardous. Il test set viene mantenuto completamente separato e utilizzato esclusivamente per la valutazione finale delle prestazioni e per l'integrazione con il modulo Prolog.

Gestione dello sbilanciamento: per gestire lo sbilanciamento tra le classi (83,9% non pericolosi contro 16,1% pericolosi) è stato adottato il *Random Oversampling*, integrato nella pipeline di addestramento ed eseguito all'interno di ciascun fold della cross-validation. Questa scelta garantisce la correttezza metodologica, evitando che informazioni dai dati di validazione influenzino l'addestramento.

3 Apprendimento supervisionato e classificazione dei modelli

Il problema di rilevazione della pericolosità degli asteroidi è definito come un compito di classificazione supervisionata binaria, in cui l'obiettivo è distinguere asteroidi potenzialmente pericolosi (Hazardous = 1), da quelli non pericolosi (Hazardous = 0).

Sono stati confrontati cinque modelli di classificazione per valutare il più efficace per il progetto: *Logistic Regression*, *Decision Tree*, *Random Forest*, *K-Nearest Neighbors (KNN)* e *Multi-Layer Perceptron (MLP)*.

Per ciascun modello, la fase di addestramento e selezione degli iperparametri è stata condotta tramite cross-validation stratificata a 10 fold, utilizzando come metrica di ottimizzazione la **ROC-AUC** (*Compute Area Under the Receiver Operating Characteristic Curve*). La curva *ROC* è un grafico che mostra quanto il modello è bravo a distinguere tra le due classi al variare della soglia di classificazione, mentre l'*AUC* è un singolo numero che rappresenta l'area geometrica che sta "sotto" la curva ROC.

Per tutti i modelli, tranne MLP, è stata utilizzata la *GridSearchCV*. Mentre, dato che MLP è caratterizzato da una maggiore complessità e dimensionalità degli iperparametri, è stata usata la *RandomizedSearchCV*. Le curve ROC, ottenute come media sui 10 fold, forniscono una rappresentazione grafica della stabilità delle prestazioni rispetto alle diverse partizioni dei dati.

L'output probabilistico del modello migliore (Random Forest) è stato integrato all'interno di un modulo di ragionamento logico sviluppato in Prolog, dove la probabilità ML contribuisce alla costruzione di uno score di rischio pesato, realizzando un approccio ibrido che unisce apprendimento statistico e conoscenza simbolica.

3.1 Logistic Regression

La Logistic Regression stima la probabilità di appartenenza a una classe mediante una funzione logistica applicata a una combinazione lineare delle feature. Nel contesto della classificazione binaria, il modello apprende un insieme di pesi che permettono di stimare la probabilità di appartenenza a una delle due classi.

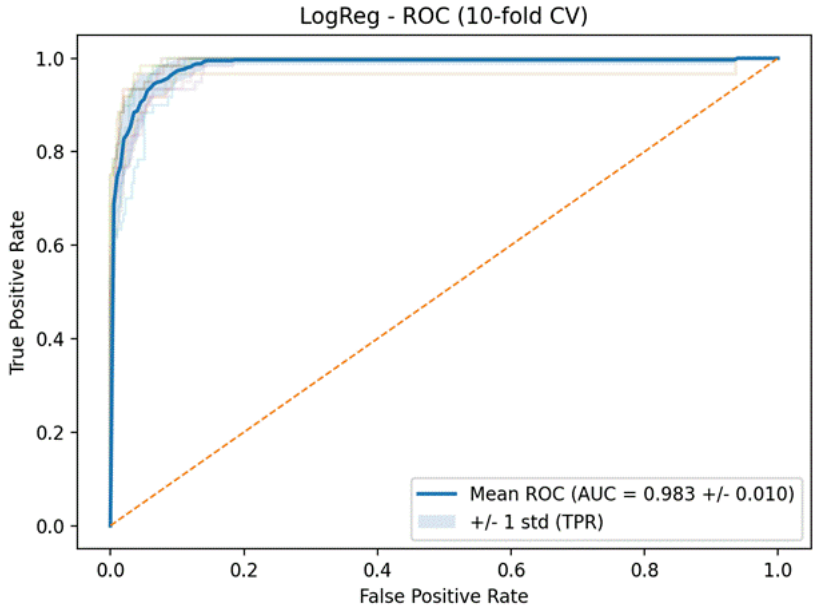
Iperparametri ottimali individuati tramite Cross-Validation:

- $C = 1.0$

  Parametro di regolarizzazione che controlla il compromesso tra adattamento ai dati e penalizzazione dei pesi.

- *class_weight* = None

  Non è stato applicato un bilanciamento interno delle classi, poiché lo sbilanciamento del dataset è stato gestito con tecniche di oversampling integrate nella pipeline, senza dover effettuare ulteriori correzioni.



La curva ROC media mostra una buona capacità discriminativa, con una bassa variabilità tra i fold, a conferma della stabilità del modello.

Metriche sul test set (migliori iperparametri):

- *Accuracy* = 0.9339 ± 0.0173
- *Precision* = 0.7306 ± 0.0536

- *Recall = 0.9338 ± 0.0596*
- *F1-score = 0.8198 ± 0.0420*
- *ROC-AUC = 0.9892 ± 0.0061*

3.2 Decision Tree

Il Decision Tree (Albero di Decisione) è un albero in cui:

- ogni nodo interno (non foglia) è etichettato con condizioni, ossia test booleani basati sui valori delle feature negli esempi e connessi con due figli, radici di sotto-alberi, attraverso archi tipicamente etichettati con *true* e *false*.
- ogni foglia dell'albero è etichettata con una stima puntuale sulla classe.

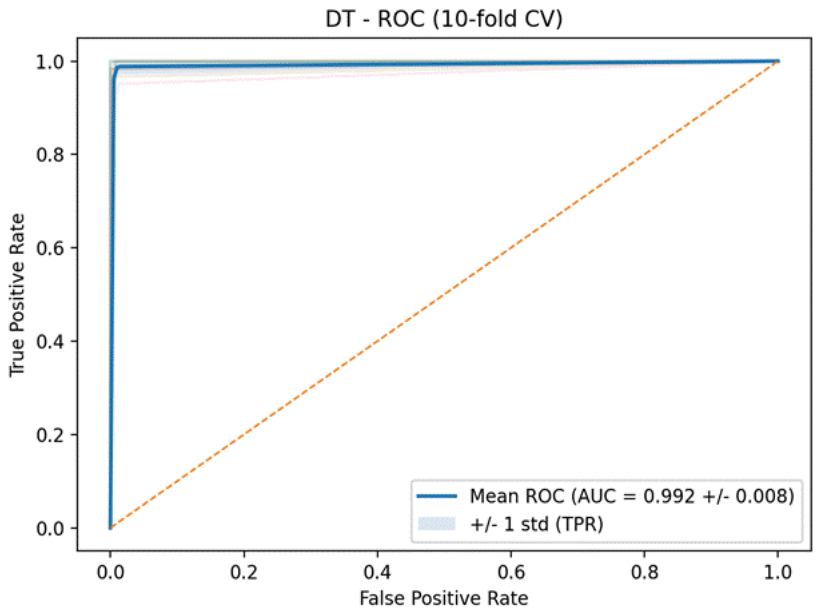
Iperparametri ottimali individuati tramite Cross-Validation:

- *max_depth* = None

  L'albero può crescere senza limiti di profondità, consentendo la cattura di pattern complessi.
- *min_samples_leaf* = 10

  Ogni foglia deve contenere almeno 10 campioni, prevenendo l'overfitting.
- *class_weight* = None

  Lo sbilanciamento è gestito dall'oversampling nella pipeline.



Il modello ha ottenuto prestazioni molto elevate, con una curva ROC che si avvicina rapidamente all'angolo superiore sinistro, indicando un'ottima capacità discriminativa.

Metriche sul test set (migliori iperparametri):

- *Accuracy = 0.9957 ± 0.0071*
- *Precision = 0.9804 ± 0.0293*
- *Recall = 0.9934 ± 0.0200*
- *F1-score = 0.9868 ± 0.0219*
- *ROC-AUC = 0.9987 ± 0.0027*

3.3 Random Forest

La Random Forest è un modello composto da più alberi di decisione. L'idea è quella di addestrare un certo numero di alberi su parti differenti del dataset. Per ogni esempio da classificare, ognuno darà una predizione, le predizioni poi verranno aggregate per formare la predizione finale per l'esempio dato. Questo approccio riduce il rischio di overfitting e migliora la generalizzazione. L'efficacia dipende dalla diversità degli alberi generati in modo da ottenere da ciascuno predizioni anche diverse.

Iperparametri ottimali individuati tramite Cross-Validation:

- *n_estimators* = 50

Numero di alberi nella foresta. Un valore pari a 50 garantisce un buon compromesso tra stabilità delle prestazioni e costo computazionale.

- *max_depth* = None

Nessun limite alla profondità degli alberi, consentendo la cattura di relazioni complesse, prevenendo l'overfitting.

- *max_features* = "sqrt"

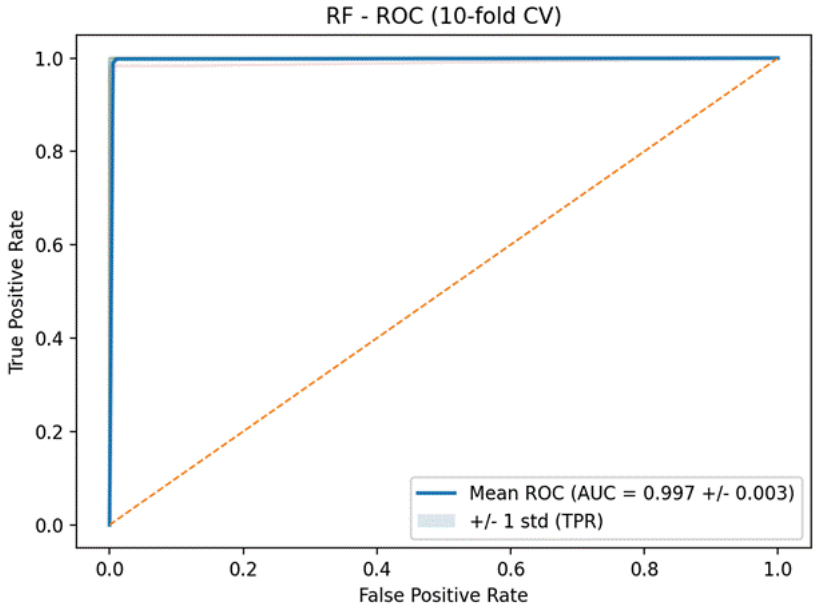
Ad ogni split viene selezionato un sottoinsieme casuale di feature pari alla radice quadrata del totale, riducendo la correlazione tra gli alberi e aumentando la diversità del modello.

- *min_samples_leaf* = 1

Ogni foglia può contenere anche un solo campione. Questo consente agli alberi di adattarsi maggiormente ai dati.

- *class_weight* = "balanced_subsample"

Pesa le classi in modo che il modello dia maggiore attenzione alle classi sbilanciate.



La curva ROC-AUC per la Random Forest mostra una separazione praticamente perfetta tra le classi, con un valore AUC prossimo a 1.0, confermando che questo è il modello più efficace per il problema in esame.

Metriche sul test set (migliori iperparametri):

- *Accuracy = 0.9989 ± 0.0032*
- *Precision = 0.9934 ± 0.0187*
- *Recall = 1.0000 ± 0.0000*
- *F1-score = 0.9967 ± 0.0097*
- *ROC-AUC = 1.0000 ± 0.0003*

3.4 K-Nearest Neighbors (KNN)

Il K-Nearest Neighbors (KNN) è un modello del *case-based reasoning*, ed è basato sulla similitudine. Funziona memorizzando i dati di addestramento e classificando i nuovi esempi in base alla maggioranza delle etichette dei vicini più prossimi, solitamente misurati con la distanza euclidea.

Iperparametri ottimali individuati tramite Cross-Validation:

- *n_neighbors* = 11

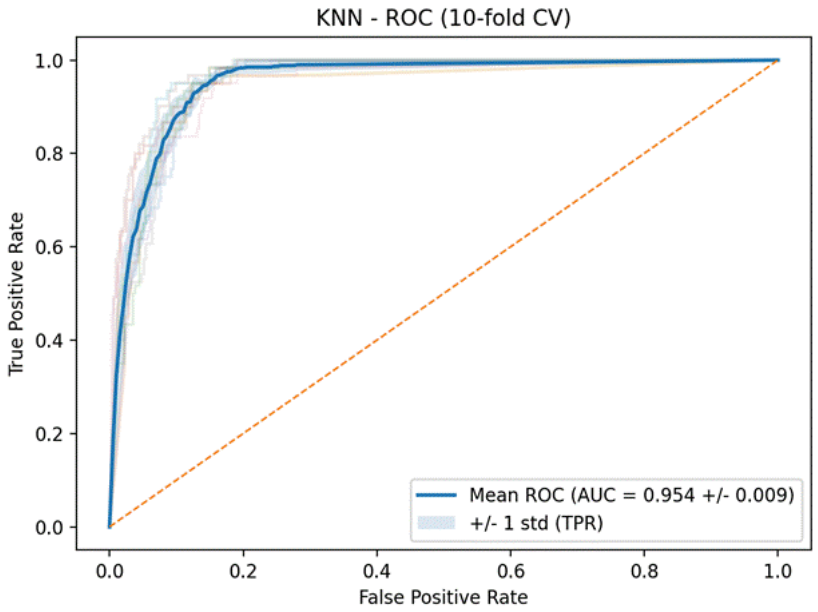
Un valore pari a 11 consente di ridurre la sensibilità al rumore presente nei dati e di ottenere decisioni più stabili, rispetto a valori più bassi di k.

- *weights* = "distance"

I pesi sono inversamente proporzionali alla distanza, favorendo i vicini più prossimi.

- *metric* = "minkowski"

La metrica di Minkowski rappresenta una generalizzazione della distanza euclidea e di Manhattan, permettendo al modello di adattarsi meglio alla struttura geometrica dei dati.



La curva ROC-AUC per il modello KNN mostra una buona separazione. Il valore AUC, sebbene buono, risulta inferiore rispetto agli altri modelli, risentendo della dimensionalità elevata dello spazio delle feature.

Metriche sul test set (migliori iperparametri):

- *Accuracy = 0.8838 ± 0.0269*
- *Precision = 0.5897 ± 0.0655*
- *Recall = 0.9139 ± 0.0602*
- *F1-score = 0.7169 ± 0.0529*
- *ROC-AUC = 0.9532 ± 0.0214*

3.5 Multi-Layer Perceptron (MLP)

Il Multi-Layer Perceptron (MLP) è una rete neurale feedforward composta da uno o più strati nascosti. La presenza di questi strati permette al modello di apprendere relazioni non lineari tra le variabili di input e l'output, rendendolo più flessibile rispetto ai modelli lineari.

Iperparametri ottimali individuati tramite Cross-Validation:

- *hidden_layer_sizes* = (128, 64)

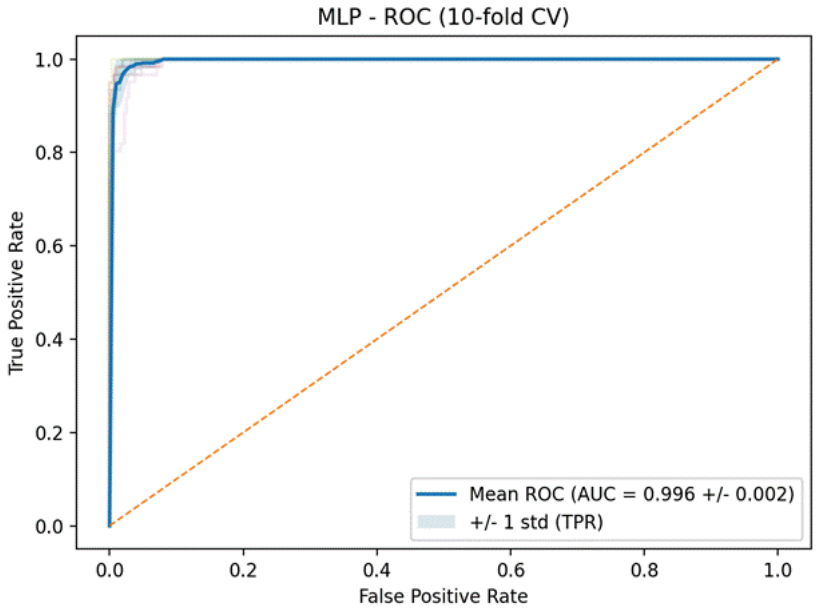
La rete è composta da due strati nascosti con 128 e 64 neuroni. Consente di modellare strutture complesse nei dati senza aumentare eccessivamente la complessità del modello.

- *alpha* = 0.001

Parametro di regolarizzazione che limita la crescita eccessiva dei pesi, e riduce il rischio di overfitting.

- *learning_rate_init* = 0.001

Velocità di apprendimento iniziale per l'ottimizzatore.



La curva ROC-AUC per l'MLP indica un'ottima capacità discriminativa, con un AUC molto vicino a 1.0.

Metriche sul test set (migliori iperparametri):

- *Accuracy = 0.9904 ± 0.0100*
- *Precision = 0.9671 ± 0.0420*

- *Recall = 0.9735 ± 0.0327*
- *F1-score = 0.9703 ± 0.0307*
- *ROC-AUC = 0.9994 ± 0.0013*

3.6 Confronto delle prestazioni dei modelli utilizzati

Modello	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.9339 (± 0.0173)	0.7306 (± 0.0536)	0.9338 (± 0.0596)	0.8198 (± 0.0420)	0.9892 (± 0.0061)
Decision Tree	0.9957 (± 0.0071)	0.9804 (± 0.0293)	0.9934 (± 0.0200)	0.9868 (± 0.0219)	0.9987 (± 0.0027)
Random Forest	**0.9989 (± 0.0032)**	**0.9934 (± 0.0187)**	**1.0000 (± 0.0000)**	**0.9967 (± 0.0097)**	**1.0000 (± 0.0003)**
KNN	0.8838 (± 0.0269)	0.5897 (± 0.0655)	0.9139 (± 0.0602)	0.7169 (± 0.0529)	0.9532 (± 0.0214)
MLP	0.9904 (± 0.0100)	0.9671 (± 0.0420)	0.9735 (± 0.0327)	0.9703 (± 0.0307)	0.9994 (± 0.0013)

Dall'analisi delle metriche riportate in tabella emerge che tutti i modelli mostrano buone prestazioni, ma con alcune differenze significative.

Il *KNN* ottiene un Recall elevato, ma presenta valori più bassi di Precision e F1-score, indicando una maggiore presenza di falsi positivi. La *Logistic Regression* migliora le prestazioni complessive rispetto al KNN e mostra una buona stabilità, ma rimane inferiore rispetto ai modelli più complessi. Il *Decision Tree* e l'*MLP* raggiungono risultati molto elevati su tutte le metriche, con ROC-AUC prossimi a 1, dimostrando un'ottima capacità di separazione tra le classi.

Ma il modello che ottiene le prestazioni migliori è la *Random Forest*. In particolare in ROC-AUC (1.0000), F1-score (0.9967) e Recall (1.0000).

Il recall perfetto indica che il modello riesce a identificare correttamente tutti gli asteroidi potenzialmente pericolosi presenti nel test set, il che è molto importante in un problema di questo tipo, dove minimizzare i falsi negativi è fondamentale. Inoltre, la deviazione standard molto contenuta tra i fold evidenzia una buona stabilità delle prestazioni.

Per queste ragioni, la Random Forest risulta il modello più adatto per l'integrazione con il modulo di ragionamento logico, in quanto fornisce predizioni affidabili che contribuiscono alla costruzione di uno score robusto.

4 Ragionamento logico

Il progetto integra un modulo di ragionamento logico basato su *Prolog*, un linguaggio di programmazione dichiarativo fondato sulla logica del primo ordine. Questo approccio permette di rappresentare la conoscenza del dominio attraverso regole esplicite e di affidare al sistema inferenziale la valutazione delle interrogazioni.

Nel progetto, il ragionamento logico non sostituisce i modelli di Machine Learning, ma li affianca. Il suo ruolo è quello di interpretare e strutturare le informazioni prodotte nella fase di apprendimento supervisionato.

In particolare, le probabilità fornite dal modello vengono combinate con alcune caratteristiche orbitali e fisiche dell'asteroide all'interno di un sistema di regole. Questo processo consente di individuare in modo esplicito determinate condizioni di rischio e di rendere più interpretabile la valutazione finale.

4.1 Knowledge Base

La *Knowledge Base (KB)* rappresenta il nucleo del sistema basato su regole ed è progettata come un *Knowledge-Based System (KBS)*, in cui la conoscenza astronomica viene formalizzata tramite regole logiche. Non funziona come un semplice archivio di dati, ma come un meccanismo di inferenza che applica regole esperte alle informazioni fornite in ingresso dal sistema di Machine Learning.

La rappresentazione della conoscenza utilizza le feature originali dell'asteroide, mantenute nella loro forma non preprocessata per conservarne il significato.

I parametri utilizzati nel processo di ragionamento includono:

- *magnitudine assoluta*
- *diametro massimo stimato*
- *velocità relativa*
- *distanza minima di passaggio (miss distance)*
- *MOID (Minimum Orbit Intersection Distance)*
- *eccentricità*
- *incertezza orbitale*
- *invariante di Tisserand rispetto a Giove*
- *probabilità di pericolosità stimata dal modello di Machine Learning*

4.2 Regole

Le regole rappresentano l'elemento centrale del modulo di ragionamento logico e descrivono in modo formale le principali condizioni associate alla pericolosità di un asteroide. Attraverso queste regole, il sistema non si limita a fornire una valutazione finale del rischio, ma produce anche un insieme di motivazioni esplicite (*reasons*) che spiegano quali condizioni sono state soddisfatte.

Regole basate su condizioni elementari

La prima categoria comprende le regole che descrivono singole condizioni di rischio, valutate indipendentemente tra loro. Ciascuna regola si attiva quando un determinato parametro supera (o scende sotto) una soglia predefinita, e sono:

- *large_asteroid / very_large_asteroid*

  Si attiva quando la magnitudine assoluta è bassa (≤ 22.0 o ≤ 20.0). Una magnitudine più bassa indica dimensioni maggiori, quindi la regola segnala asteroidi potenzialmente più pericolosi in caso di impatto.
- *close_moid / very_close_moid*

  Si attiva quando la distanza minima di intersezione orbitale è ridotta (≤ 0.05 AU o ≤ 0.01 AU), indicando un'orbita che può avvicinarsi in modo significativo a quella terrestre.
- *close_approach / very_close_approach*

  Si attiva per distanze di avvicinamento ridotte (≤ 0.1 AU o ≤ 0.02 AU), evidenziando un passaggio ravvicinato rispetto alla Terra.
- *high_velocity / very_high_velocity*

Si attiva per velocità relative elevate (≥ 20 km/s o ≥ 35 km/s), fattore che aumenta l'energia cinetica di un eventuale impatto.

- *high_eccentricity*

  Si attiva per orbite molto ellittiche (eccentricità ≥ 0.6), generalmente associate a traiettorie variabili e meno prevedibili.

- *uncertain_orbit*

  Si attiva quando l'incertezza orbitale è elevata (≥ 5), riducendo l'affidabilità delle previsioni.

Regole basate sulla combinazione di fattori di rischio

La seconda categoria comprende le regole che identificano scenari particolarmente critici, in cui l'esistenza di più condizioni aumenta in modo significativo il livello di pericolosità, e sono:

- *large_and_close*

  Si attiva quando un asteroide è sia di grandi dimensioni sia caratterizzato da un passaggio ravvicinato alla Terra, il che aumenta notevolmente il rischio di impatto.

- *large_and_fast*

  Si attiva quando un asteroide è contemporaneamente grande e veloce. Infatti, un eventuale impatto comporterebbe un'elevata energia cinetica, con conseguenze potenzialmente gravi.

- *close_moid_uncertain*

  Si attiva quando la MOID è bassa e l'orbita presenta un'elevata incertezza, il che rende più difficile prevedere con precisione la traiettoria futura dell'asteroide.

- *close_fast_approach*

  Si attiva in presenza di un passaggio ravvicinato ad alta velocità. La combinazione riduce il tempo disponibile per eventuali contromisure.

- *large_eccentric*

  Si attiva quando un asteroide di dimensioni significative presenta un'eccentricità elevata, indicando un'orbita molto ellittica e meno prevedibile.

Integrazione dell'output dei modelli di Machine Learning

La terza ed ultima categoria include le regole che integrano l'output probabilistico dei modelli di Machine Learning come ulteriore segnale di rischio all'interno del ragionamento logico, e sono:

- *ml_high*

  Si attiva quando la probabilità stimata di pericolosità è ≥ 0.55, e identifica situazioni in cui il modello evidenzia una tendenza verso la classe "pericoloso", pur non trattandosi necessariamente di un caso estremo.

- *ml_very_high*

  Si attiva quando la probabilità stimata di pericolosità è ≥ 0.80. In questo caso il modello esprime una confidenza elevata nella classificazione come oggetto pericoloso.

4.3 Valutazione dei rischi

La valutazione dei rischi deriva dall'aggregazione delle evidenze individuate. Il sistema calcola un punteggio di rischio grezzo (*ScoreRaw*) sommando i pesi associati a ciascun trigger attivato, lo normalizza su scala 0-100 e lo traduce in un livello di rischio diviso in:

- LOW (ScoreRaw < 7): asteroide classificato come SAFE.
- MEDIUM (7 ≤ ScoreRaw < 15): asteroide sottoposto a REVIEW.
- HIGH (ScoreRaw ≥ 15): asteroide classificato come MONITOR_PRIORITY.

I pesi differenziati per trigger consentono di esprimere la diversa gravità delle condizioni di rischio: ad esempio, *very_large_asteroid* e *very_close_moid* hanno peso 4, mentre *ml_high* ha peso 1. Il punteggio massimo teorico è fissato a 30. Questo approccio garantisce coerenza tra dati osservati, rendendo il sistema non solo efficace, ma anche interpretabile e facilmente verificabile.

5 Conclusione e sviluppi futuri

Un possibile sviluppo futuro riguarda l'estensione della Knowledge Base con *regole temporali*, capaci di considerare l'evoluzione orbitale di un asteroide nel tempo. In questo modo sarebbe possibile analizzare la ricorrenza degli avvicinamenti alla Terra e modellare il comportamento dell'oggetto su più passaggi ravvicinati, ottenendo una valutazione del rischio migliore.

Un ulteriore miglioramento potrebbe consistere nell'integrazione con fonti di dati aggiornate in tempo reale, come il servizio *CNEOS* (*Center for Near Earth Object Studies*) della NASA [1]. Questo permetterebbe di trasformare il sistema da strumento di analisi statica a componente di una pipeline di monitoraggio continuo, capace di aggiornare automaticamente le valutazioni al variare dei parametri orbitali.

Dal punto di vista della componente di Machine Learning, si potrebbe approfondire l'applicazione di tecniche di *feature selection automatica*, al fine di validare e ottimizzare la scelta delle variabili più informative.

Nel complesso, questi sviluppi contribuirebbero a rendere il sistema più dinamico, scalabile e aderente a scenari operativi reali.

6 Bibliografia

[1] NASA Center for Near Earth Object Studies (CNEOS), Jet Propulsion Laboratory, https://cneos.jpl.nasa.gov/