



Classificazione del rischio di collisione in orbita mediante SVM e tecniche Ensemble: una pipeline per l'individuazione automatica di eventi ad alto rischio nei messaggi CDM

Facoltà di Ingegneria dell'informazione, informatica e statistica
Laurea Magistrale in Ingegneria Gestionale

Luca Bigi
Matricola 1817398

Relatrice
Prof.ssa Laura Palagi

Correlatore
Cap. Lorenzo Dilauro

Classificazione del rischio di collisione in orbita mediante SVM e tecniche Ensemble: una pipeline per l'individuazione automatica di eventi ad alto rischio nei messaggi CDM

Tesi di Laurea Magistrale. Sapienza Università di Roma

© 2025 Luca Bigi. Tutti i diritti riservati

Questa tesi è stata composta con L^AT_EX e la classe Sapthesis.

Email dell'autore: bigi.1817398@studenti.uniroma1.it

Sommario

Il crescente affollamento dell'orbita terrestre bassa (LEO) e la diffusione di detriti spaziali rendono sempre più necessario lo sviluppo di strumenti in grado di prevedere automaticamente e con precisione il rischio finale di collisione associato a ciascun evento di ravvicinamento tra due oggetti in orbita (evento di congiunzione), in particolare tra satelliti e detriti o tra due satelliti.

Il presente lavoro è stato svolto nell'ambito di un tirocinio curriculare presso il *Comando delle Operazioni Spaziali (COS)*, in collaborazione con la *Sapienza Università di Roma*, e ha previsto lo sviluppo di un sistema predittivo basato su tecniche di machine learning, in grado di stimare, con almeno due giorni di anticipo, il rischio finale associato a ciascun evento di congiunzione, utilizzando esclusivamente le informazioni contenute nei Conjunction Data Messages (CDM), messaggi standardizzati che descrivono le condizioni orbitali relative ai due oggetti coinvolti.

Come riferimento per la valutazione delle prestazioni, sono stati utilizzati il dataset e i criteri ufficiali della *Spacecraft Collision Avoidance Challenge*, promossa dall'Agenzia Spaziale Europea (ESA) nel 2019, così da garantire un confronto oggettivo e standardizzato sull'efficacia dei metodi sviluppati.

Dopo una panoramica delle criticità del problema e dei principali algoritmi esplorati, la tesi si concentra sulla descrizione dettagliata della pipeline finale di classificazione: un sistema multilivello, basato su un'architettura modulare ed ensemble di modelli. Il cuore del sistema è costituito da 56 Support Vector Machine, coordinate attraverso meccanismi di voto e consenso per identificare eventi classificabili con maggiore affidabilità. Gli eventi più incerti vengono gestiti da fasi successive, che impiegano classificatori specializzati addestrati su sottoinsiemi mirati del dataset, con l'obiettivo di affinare progressivamente le decisioni finali del sistema.

A complemento della pipeline di classificazione è stato sviluppato in parallelo un modulo di regressione, in grado di fornire una stima continua del rischio per gli eventi etichettati come ad alto rischio dal modulo di classificazione. L'integrazione dei due moduli ha portato alla costruzione di un sistema predittivo completo, capace di combinare classificazione binaria e stima quantitativa del rischio.

Il modello finale è stato testato sul dataset ufficiale della challenge, ottenendo uno score pari a 0.559, risultato che lo collocherebbe al secondo posto assoluto tra i 97 team partecipanti alla Challenge, e al primo posto tra tutti i modelli interamente basati su tecniche di machine learning.

Indice

1 Introduzione	1
1.1 La natura del rischio di collisione e i Conjunction Data Messages (CDM)	1
1.2 La Spacecraft Collision Avoidance Challenge (ESA 2019)	2
1.3 Obiettivi della tesi	3
2 Analisi del problema	5
2.1 Descrizione e struttura del dataset ESA	5
2.2 Criticità del problema: squilibrio, salti di rischio e rappresentatività	7
3 Analisi della letteratura	11
3.1 Modelli selezionati dalla Challenge ESA 2019: analisi comparativa .	12
3.2 Considerazioni finali sui Modelli della Challenge	13
4 Approccio metodologico	15
4.1 Strategia generale: Sviluppo Progressivo di Modelli per la Collision Avoidance	15
4.2 Considerazioni finali sui Modelli Sviluppati	21
5 Preprocessing del dataset per l'Ensemble di Classificazione	23
5.1 Ingegnerizzazione e selezione preliminare delle feature	23
5.2 Selezione delle feature e analisi di importanza	25
5.3 Tecnica di Bilanciamento tramite Campionamento a Bin	27
5.4 Tecnica di normalizzazione adottata	29
5.5 Considerazioni sull'uso di dati sintetici	29
6 Ensemble di Classificazione	31
6.1 Obiettivo e struttura dell'Ensemble di Classificazione	31
6.2 Pseudocodice dell'Ensemble di Classificazione	36
6.3 Tuning e validazione dell'Ensemble di Classificazione	40
7 Integrazione della Regressione nella Previsione Finale	47
7.1 Ruolo della Regressione nel sistema predittivo combinato	47
7.2 Integrazione dei due Moduli	48
8 Discussione critica	49
8.1 Valutazione del Sistema Combinato sul test set della challenge . . .	49
8.2 Punti di forza del Sistema Combinato	51

8.3	Limiti e potenzialità di generalizzazione	52
9	Conclusioni	55
9.1	Possibili implicazioni operative per il Comando delle Operazioni Spaziali	55
9.2	Configurazione hardware per l'esecuzione locale dei moduli su Python	56
	Ringraziamenti	57
	Bibliografia	59

Capitolo 1

Introduzione

Negli ultimi decenni, il crescente affollamento dell'orbita terrestre bassa (LEO) ha sollevato preoccupazioni sempre più rilevanti per la sicurezza operativa dei satelliti [22, 21, 19]. In orbita terrestre si contano attualmente circa 14.000 satelliti, dei quali circa 11.700 sono ancora operativi [12].

Un esempio concreto delle conseguenze potenziali di questo sovraffollamento si è verificato il 10 febbraio 2009, quando il satellite Cosmos-2251 e il satellite Iridium-33 hanno colliso generando due nubi di detriti che hanno contaminato una delle fasce più trafficate della LEO [32]. L'analisi dell'evento ha rilevato che tale collisione ha prodotto migliaia di frammenti, di cui oltre 1800 di dimensioni superiori a 10 cm, classificandosi come la più grave frammentazione accidentale mai registrata [18]. Inoltre le simulazioni indicano che molti di questi detriti resteranno in orbita per diversi decenni, fino a un secolo [4].

Considerando che un impatto con oggetti di tali dimensioni è nella maggior parte dei casi sufficiente a causare la distruzione completa di un satellite attivo [32], la permanenza di questi detriti in orbita costituisce una minaccia significativa per la sicurezza delle missioni spaziali.

Per rispondere a questo tipo di minacce, le agenzie spaziali a livello internazionale hanno definito normative e procedure operative volte a ridurre la probabilità di collisioni in orbita e a garantire la sostenibilità dell'ambiente spaziale nel lungo periodo: il processo operativo noto come *Collision Avoidance* è ormai parte integrante della gestione quotidiana delle missioni satellitari. [13, 23, 6]

1.1 La natura del rischio di collisione e i Conjunction Data Messages (CDM)

Il processo di gestione del rischio di collisione è supportato dallo Space Debris Office, che monitora costantemente eventi di avvicinamento ravvicinato (conjunction events) utilizzando i dati forniti dal Combined Space Operations Center (CSpOC) statunitense, sotto forma di Conjunction Data Messages (CDM). Questi messaggi vengono inviati regolarmente a partire da diversi giorni prima del possibile Time of Closest Approach (TCA), ovvero il momento in cui due oggetti in orbita si troveranno alla minima distanza tra loro, fino a pochi minuti prima del TCA; vengono generati

a partire da osservazioni radar e ottiche e tengono traccia di 103 caratteristiche relative all'avvicinamento tra l'oggetto chaser e il target, tra cui:

- Codice identificativo dell'evento di congiunzione (*event_id*) e il relativo TCA
- Una stima del rischio di collisione corrente - *risk* - calcolata mediante algoritmi come quello di Alfriend e Akella [1] a partire dai dati riportati nella CDM stessa.

La frequente ricezione e aggiornamento delle CDM, fino a tre volte al giorno per ciascun evento di congiunzione rilevato, consente una stima progressivamente più accurata del rischio di collisione man mano che si avvicina il Time of Closest Approach. Qualora si preveda che il rischio in prossimità del TCA superi una soglia predefinita, viene attivata una procedura di allarme che può portare alla pianificazione di una manovra anticollisione [32]. Tuttavia, è importante considerare che l'esecuzione di tali manovre comporta costi e consumo di carburante. Per questo motivo, è fondamentale ridurre il numero di falsi allarmi, prestando però estrema attenzione a non trascurare alcun evento realmente pericoloso, che potrebbe causare la distruzione di uno o più satelliti e la formazione di nuove nubi di detriti orbitali.

In questo contesto operativo, l'adozione di metodi basati su machine learning si propone come supporto avanzato alle decisioni, con l'obiettivo di anticipare l'evoluzione del rischio utilizzando le informazioni fornite dalle CDM ricevute con almeno 2 giorni di anticipo dal TCA. Questo perché, nella pratica operativa, è essenziale disporre di una stima affidabile del rischio di collisione con un margine temporale sufficiente a valutare ed eventualmente pianificare una manovra di *Collision Avoidance*. Il lavoro di ricerca presentato in questa tesi si inserisce in questa prospettiva, con l'obiettivo di migliorare la capacità predittiva e la generalizzazione di modelli per l'allerta precoce di eventi potenzialmente critici.

1.2 La Spacecraft Collision Avoidance Challenge (ESA 2019)

Nel 2019 l'Agenzia Spaziale Europea (ESA), in collaborazione con l'Università di Scienze e Tecnologie AGH di Cracovia, ha organizzato la *Spacecraft Collision Avoidance Challenge* [11], una competizione internazionale volta a stimolare lo sviluppo di modelli predittivi per la stima del rischio di collisione in orbita. L'obiettivo era esplorare il potenziale contributo di tecniche automatiche, incluse quelle basate su machine learning, nel supportare le decisioni operative in scenari di *Collision Avoidance*.

La challenge ha messo a disposizione dei partecipanti un dataset operativo anonimo suddiviso in training e test set blind, costruito a partire da CDM relative a eventi di congiunzione reali per i quali non sono state effettuate manovre evasive, al fine di garantire la coerenza tra l'evoluzione naturale del rischio e i dati forniti.

La richiesta era quella di prevedere, con almeno due giorni di anticipo, il valore del rischio di collisione finale associato a ciascun evento osservato nel test set blind, e quindi classificare ciascun evento come ad alto o a basso rischio, utilizzando esclusivamente le CDM disponibili fino a quel momento. Secondo le linee guida

operative comunemente adottate per le manovre anticollisione, un evento viene considerato ad alto rischio se il rischio di collisione in prossimità del TCA supera una soglia prestabilita (tipicamente -5 o -4)¹. Tuttavia, nella challenge organizzata da ESA, per cercare di ridurre lo sbilanciamento del dataset verso gli eventi a basso rischio e rendere i modelli sviluppati più sensibili agli eventi potenzialmente critici ma rari, è stata adottata una soglia più bassa: un evento è stato etichettato come ad *alto rischio* se ha riportato un rischio maggiore o uguale di -6 nell'ultima CDM ricevuta con tempo rimanente per il TCA minore di 1 giorno, viceversa, è stato classificato come a *basso rischio* se il rischio risultava inferiore a tale soglia [32].

1.3 Obiettivi della tesi

Il presente lavoro è stato svolto nell'ambito di un tirocinio curriculare presso il *Comando delle Operazioni Spaziali* (COS), in collaborazione con la *Sapienza Università di Roma*. L'obiettivo principale è lo sviluppo di un sistema avanzato di classificazione del rischio di collisione atteso al TCA e associato agli eventi di congiunzione, utilizzando i dati storici estratti dai Conjunction Data Messages (CDM). In particolare, il sistema mira a prevedere con almeno due giorni di anticipo dal TCA, se un evento evolverà in una condizione di rischio elevato o rischio basso. Il lavoro si pone come estensione sperimentale della *Spacecraft Collision Avoidance Challenge* organizzata dall'Agenzia Spaziale Europea (ESA) nel 2019. In particolare questa tesi si propone di:

- Comprendere a fondo la struttura e le peculiarità del dataset della Challenge, analizzando i limiti della sua rappresentatività, in particolare la distribuzione fortemente sbilanciata a favore degli eventi a basso rischio, la presenza di casi con salti improvvisi di rischio, e la scarsità complessiva di eventi ad alto rischio, che rende particolarmente difficile l'addestramento di modelli generalizzabili.
- Progettare, implementare e validare un sistema di classificazione modulare, integrabile in una pipeline multifase basata su Support Vector Machine, alberi decisionali, boosting, bagging, votazioni multiple e filtri progressivi, al fine di identificare con accuratezza gli eventi a rischio, riducendo al minimo i falsi positivi e soprattutto i falsi negativi.
- Analizzare criticamente le scelte metodologiche adottate durante lo sviluppo. In particolare, si valuteranno le strategie di feature engineering, le tecniche di bilanciamento del dataset, la selezione delle soglie decisionali e l'impiego di votazioni multiple come meccanismo di rafforzamento decisionale.
- Dimostrare che, in un contesto con dati limitati e fortemente sbilanciati, modelli di machine learning semplici e mirati che sfruttano solo l'ultima informazione disponibile, se ben configurati, possono ottenere prestazioni superiori a quelle di modelli più complessi che sfruttano l'intera serie storica.

¹Salvo diversa indicazione, tutti i valori di rischio saranno considerati in scala logaritmica, coerentemente con quanto adottato nell'algoritmo sviluppato in questa tesi e al formato utilizzato nel dataset della Challenge.

Il modello descritto in questa tesi è stato progettato per integrarsi con un modulo di regressione sviluppato nella tesi parallela a cura del collega Rocco Salvatore, permettendo una previsione finale del rischio più robusta e precisa. Le performance del sistema integrato verranno quantificate tramite metriche mirate tra cui F_2 (3.2), MSE_{HR} (3.3) e Score (3.4). Tali metriche corrispondono a quelle ufficialmente adottate dalla Challenge, garantendo così piena coerenza con i criteri di valutazione della competizione.

Capitolo 2

Analisi del problema

In questo capitolo si analizza nel dettaglio il problema proposto durante la *Spacecraft Collision Avoidance Challenge* [11], con l’obiettivo di comprenderne le implicazioni tecniche e operative. Dopo aver descritto la struttura del dataset fornito durante la challenge, verranno evidenziate alcune criticità rilevanti per l’addestramento e la valutazione di modelli predittivi supervisionati, tra cui lo sbilanciamento delle classi e la scarsità di eventi ad alto rischio. Questi aspetti hanno influenzato direttamente le scelte metodologiche adottate nella fase di modellazione, trattate nei capitoli successivi.

2.1 Descrizione e struttura del dataset ESA

Il dataset fornito durante la Challenge, è stato costruito a partire da una raccolta di Conjunction Data Messages (CDM) anonimizzati, relativi a eventi di congiunzione reali in orbita terrestre bassa (LEO). Come anticipato nella Sezione 1.1, a ciascun evento è associata una serie temporale di CDM, ciascuna delle quali descrive lo stato relativo dei due oggetti orbitanti coinvolti nell’evento di congiunzione identificato da uno specifico *event_id*.

Feature	Description
<i>event_id</i>	Unique identifier for a conjunction event (shared by all CDMs of the same event)
<i>risk</i>	Self-computed value at the epoch of each CDM [base 10 log]
<i>time_to_tca</i>	Time interval between CDM creation and time-of-closest approach [days]
<i>max_risk_scaling</i>	Scaling factor used to compute maximum collision probability
<i>max_risk_estimate</i>	Maximum collision probability obtained by scaling combined covariance
<i>miss_distance</i>	Relative position between chaser and target at tca [m]
<i>mahalanobis_distance</i>	Statistical distance between predicted trajectories, accounting for orbital uncertainties [2]
<i>relative_speed</i>	Relative speed between chaser and target at tca [m/s]
<i>last_cdm_risk</i>	Target variable: <i>risk</i> of the final CDM with <i>time_to_tca</i> < 1

Tabella 2.1. Descrizioni delle variabili utilizzate.

Ogni CDM include un insieme di 103 feature, nel presente lavoro, si è selezionato per analisi successive un sottoinsieme di tali feature, partendo inizialmente dalle 20 ritenute più rilevanti secondo l’analisi conclusiva pubblicata al termine della competizione [32], successivamente però alcune di queste sono state escluse dalle successive analisi a causa dell’elevata presenza di valori "NaN". Le CDM non vengono

distribuite a intervalli temporali regolari, ma sono emesse in base all’evoluzione del tracciamento orbitale e alla valutazione del rischio. Di conseguenza, sia l’intervallo di tempo tra una CDM e la successiva, sia il numero totale di CDM generate per un determinato evento possono variare in modo significativo da un caso all’altro.

A scopo esemplificativo, si riporta nella Tabella 2.1 l’elenco delle variabili selezionate per l’addestramento del modello finale descritto in questa tesi, le quali saranno frequentemente richiamate e analizzate nei capitoli successivi. Tali feature rappresentano un sottoinsieme considerato informativo e privo di valori mancanti, costruito a partire dal dataset ufficiale della challenge.

Il database della challenge è suddiviso in:

- Training set, che comprende anche CDM con $time_to_tca < 2$
- Test set, contenente solo CDM con $time_to_tca \geq 2$.

Nel training set sono presenti CDM con qualsiasi valore di $time_to_tca$, incluse anche le osservazioni più vicine all’evento di massimo avvicinamento, cioè con $time_to_tca < 1$, indispensabili per definire la variabile target *last_cdm_risk* per ciascun *event_id* univoco presente nel dataset.

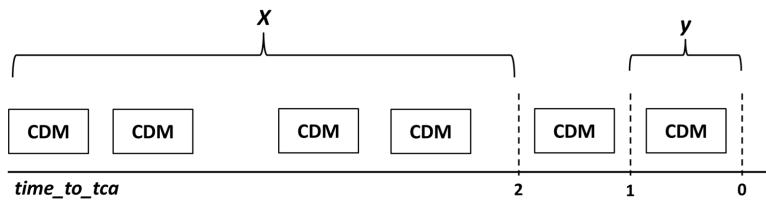


Figura 2.1. Struttura del training set

Nel test set tutte le CDM con $time_to_tca \leq 2$ sono state rimosse intenzionalmente dagli organizzatori della challenge [32]. Questa scelta è stata fatta per impedire che i partecipanti potessero accedere al valore di rischio reale da prevedere (con $time_to_tca < 1$), simulando così un contesto operativo in cui le decisioni devono essere prese con almeno 2 giorni di anticipo.

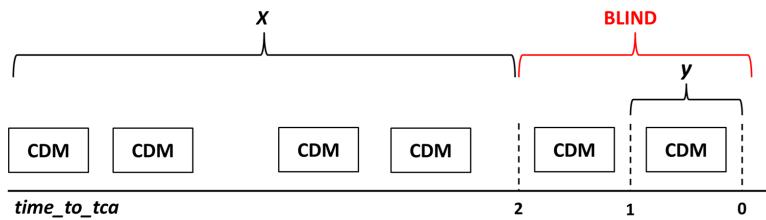


Figura 2.2. Struttura del test set

Di conseguenza, per ogni evento presente nel test set, le previsioni devono essere effettuate utilizzando esclusivamente le CDM con $time_to_tca \geq 2$, con l’obiettivo di stimare il rischio che si registrerà nell’ultima CDM con $time_to_tca < 1$, ovvero quella più vicina al momento del TCA.

I dati coprono un intervallo temporale compreso tra il 2015 e il 2019, e includono informazioni su più di 15.000 eventi di congiunzione, per un totale di circa 187.000 CDM. In particolare la Tabella 2.2 mostra la dimensione dei due dataset forniti per la competizione.

	CDM	Total events	High risk	Low risk
Train set	162634	13154	365	12789
Test set	24484	2167	150	2017

Tabella 2.2. Composizione dei dataset della challenge.

Nonostante l'apparente ampiezza del training set, i dati sono fortemente sbilanciati verso i casi a basso rischio, inoltre una porzione significativa degli eventi classificabili come ad alto rischio non risulta effettivamente utilizzabile per addestrare modelli di apprendimento supervisionato: per poter associare a ciascun evento un valore target, è necessario disporre di almeno una CDM con $time_to_tca < 1$, nella quale è contenuto il valore di riferimento finale del rischio, tuttavia una parte degli eventi presenti nel training set non contiene alcuna CDM con $time_to_tca < 1$, rendendo impossibile la costruzione del target. Questo limita di fatto il numero di eventi realmente utilizzabili.

In particolare, tra gli eventi del training set per cui l'ultima CDM disponibile indica un rischio maggiore o uguale di -6 , si contano 365 eventi, tuttavia, solo 66 di questi eventi hanno l'ultima CDM ricevuta con $time_to_tca < 1$. Tutti gli altri eventi risultano quindi inutilizzabili per un approccio supervisionato, poiché non è possibile sapere quale sia effettivamente il rischio finale ad essi associato in assenza di una stima aggiornata. Questa limitazione, i cui effetti sono illustrati nella Tabella 2.3, riduce drasticamente la quantità di casi ad alto rischio realmente disponibili, accentuando il problema dello sbilanciamento del dataset e rendendo più complesso addestrare modelli generalizzabili.

	CDM	Total events	High risk	Low risk
Train set	138462	8293	66	8227
Test set	24484	2167	150	2017

Tabella 2.3. Composizione dei dataset utilizzabili.

2.2 Criticità del problema: squilibrio, salti di rischio e rappresentatività

Come descritto nella Sezione 2.1 il dataset di training fornito durante la Challenge presenta una serie di criticità che rendono particolarmente complesso l'addestramento di modelli di apprendimento supervisionato.

La distribuzione della variabile target *last_cdm_risk* (Figura 2.3) mostra una chiara prevalenza di valori a basso rischio, con un andamento fortemente asimmetrico e schiacciato su valori prossimi a -30 .

Inoltre essendo la challenge ormai conclusa e disponendo del rischio vero degli eventi del test set, è possibile effettuare un'analisi ex post per quantificare il disallineamento tra il dataset di train e quello di test. Nostante le distribuzioni illustrate in Figura 2.3 e 2.4, appaiano simili tra training e test set, questa apparente simmetria nasconde una criticità strutturale più profonda.

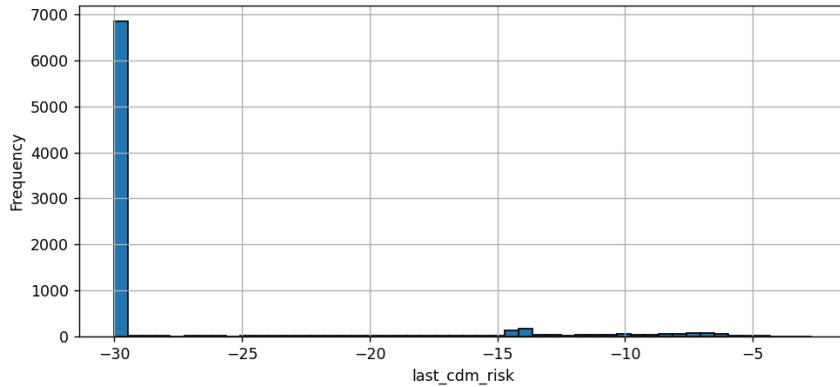


Figura 2.3. Distribuzione del rischio finale degli eventi del train set

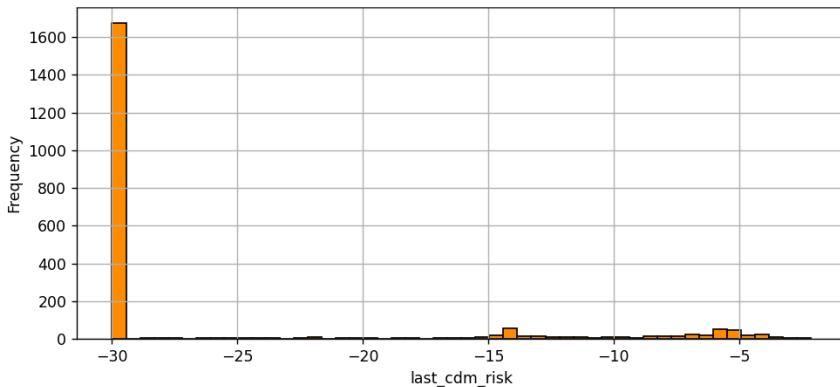


Figura 2.4. Distribuzione del rischio finale degli eventi del test set

In particolare, il grafico seguente confronta l'ampiezza massima dei salti di rischio negli eventi ad alto rischio presenti nel training set e nel test set: il test set contiene diversi eventi caratterizzati da salti improvvisi di rischio (Figura 2.6), ovvero situazioni in cui il valore stimato di rischio rimane basso per gran parte dell'evento e subisce un incremento repentino solo nelle CDM finali. Questo tipo di eventi sono stati assegnati per la maggior parte al test set, senza un'adeguata rappresentazione nel training (Figura 2.5), il che introduce una forma di asimmetria artificiale tra train e test set, che compromette la capacità di generalizzazione dei modelli.

In un contesto ideale, sarebbe stato certamente preferibile effettuare uno shuffle casuale dell'intero insieme di eventi, suddividendolo poi in training e test set in modo da mantenere proporzionale il rapporto tra il numero di eventi a basso e ad alto rischio nei due dataset. In questo modo, anche le situazioni più complesse o anomale

sarebbero state presenti nei dati di training, permettendo ai modelli di apprendere anche quei pattern meno frequenti ma rilevanti dal punto di vista operativo.

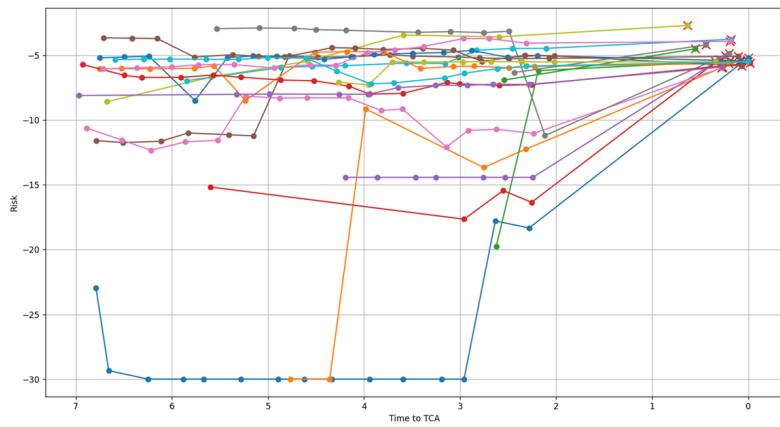


Figura 2.5. Andamento del rischio dalla ricezione della prima cdm al TCA

20 Eventi del *TRAIN SET* con massima $|risk - last_cdm_risk|$ e $last_cdm_risk \geq -6$

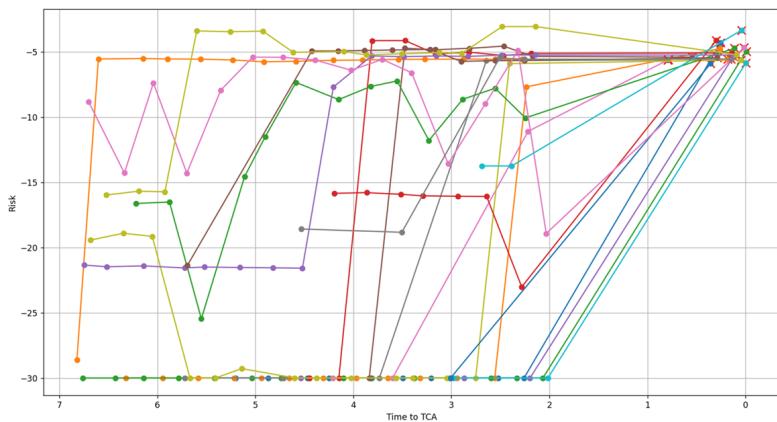


Figura 2.6. Andamento del rischio dalla ricezione della prima cdm al TCA

20 Eventi del *TEST SET* con massima $|risk - last_cdm_risk|$ e $last_cdm_risk \geq -6$

Nonostante queste criticità, nel seguito di questa tesi si è scelto comunque di utilizzare la suddivisione originale dei dataset proposti per la challenge, lasciando il test set “blind” durante tutte le fasi di sviluppo e validazione del modello. Il training set è stato suddiviso in sottoinsiemi per la validazione incrociata e il tuning degli iperparametri dei modelli sviluppati, mantenendo invariato il rapporto tra il numero di casi a basso rischio rispetto a quelli ad alto rischio su ogni sottoinsieme di validazione, mentre il test set è stato utilizzato esclusivamente per la valutazione finale delle prestazioni, al fine di preservare la comparabilità con i risultati ufficiali della Challenge.

Capitolo 3

Analisi della letteratura

Il lavoro di questa tesi ha preso avvio da un'analisi approfondita della letteratura disponibile e dei contributi pubblici dei team che hanno partecipato alla *Spacecraft Collision Avoidance Challenge* organizzata da ESA nel 2019.

La funzione obiettivo da minimizzare, nel contesto della Challenge, era il rapporto tra l'errore quadratico medio calcolato solo sugli eventi realmente ad alto rischio (MSE_{HR}) e F_β con $\beta = 2$, che penalizza severamente i falsi negativi e valorizza la sensibilità del modello.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.1)$$

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}, \quad F_2 = \frac{5 \cdot \text{Precision} \cdot \text{Recall}}{4 \cdot \text{Precision} + \text{Recall}} \quad (3.2)$$

$$MSE_{HR} = \frac{\sum_{i \in \mathcal{H}} (r_i - \hat{r}_i)^2}{N_{HR}} \quad (3.3)$$

$$\text{Score} = \frac{MSE_{HR}}{F_2} \quad (3.4)$$

- r_i è il rischio vero dell'evento i contenuto nella CDM più vicina al TCA
- \hat{r}_i è la stima del rischio prodotta dal modello per l'evento i
- N_{HR} è il numero di eventi realmente ad alto rischio, ovvero con $r_i \geq -6$
- \mathcal{H} è l'insieme degli indici i degli eventi realmente ad alto rischio.

Questa formulazione imponeva ai partecipanti di mantenere elevata recall pur riducendo la sovrastima del rischio, in un contesto fortemente sbilanciato e con pochi dati utili.

Alla competizione hanno preso parte 97 team internazionali, provenienti da ambiti eterogenei (Professionisti, universitari e amatori) e sono state utilizzate strategie che spaziavano da modelli deterministici a ensemble complessi di modelli di machine learning. Tra questi, il team *sesc* ha costruito un modello basato su una cascata di soglie deterministiche ed ha ottenuto uno score di 0.555, il valore più basso tra quelli

dei team partecipanti, corrispondente alla migliore prestazione secondo la metrica definita.

Un punto di riferimento importante era rappresentato dalla baseline ufficiale, denominata *Last Risk Prediction (LRP)* [32], la quale assegna la previsione \hat{r}_i in maniera deterministica per ciascun evento i -esimo:

$$\hat{r}_i = \begin{cases} r_{i_last_obs} & \text{se } r_{i_last_obs} \geq -6 \\ -6.001 & \text{se } r_{i_last_obs} < -6 \end{cases}$$

dove $r_{i_last_obs}$ rappresenta il valore di rischio riportato nell'ultima CDM disponibile con $time_to_tca \geq 2$ dell'evento i -esimo.

Tale baseline, nota ai partecipanti sin dall'inizio della challenge, ottiene uno score di 0.694 sul test set ufficiale; solo 12 team su 97 sono riusciti ad ottenere uno score migliore [10].

3.1 Modelli selezionati dalla Challenge ESA 2019: analisi comparativa

L'analisi dei principali lavori pubblicati ha evidenziato la diversità delle tecniche sperimentate durante la challenge, che spaziano da modelli deterministicici a reti neurali profonde, fino a sistemi ibridi ed ensemble di machine learning. Di seguito si sintetizzano alcuni dei lavori analizzati:

Sesc - 1° classificato - score 0.5553

Questo team ha adottato un approccio interamente deterministico, basato su una cascata di soglie costruita tramite analisi statistica. Pur non utilizzando tecniche di machine learning, il modello ha ottenuto il miglior risultato sulla leaderboard ufficiale, con uno score di 0.5553. Si tratta di un approccio molto efficace, ma poco flessibile e poco scalabile: in caso di cambiamenti nei dati o nel contesto, le soglie devono essere ridefinite manualmente. I modelli di machine learning, invece, sono più adatti a essere scalati nel tempo: possono integrare nuovi dati, essere riaddestrati e rivalidati, e tendono a migliorare le proprie prestazioni man mano che la quantità di dati disponibili aumenta. Questo aspetto è particolarmente rilevante in un contesto come quello della challenge, in cui il dataset iniziale è limitato ma potrebbe essere arricchito in futuro.

Magpies - 3° classificato - score 0.5849

Il team ha proposto un'architettura siamese basata su reti neurali ricorrenti (RNN) e Manhattan LSTM, progettata per distinguere eventi anomali e non anomali, ciascuno classificabile come a basso o alto rischio.

Hanno quindi suddiviso gli eventi in quattro categorie: anomalie ad alto rischio, anomalie a basso rischio, non anomalie ad alto rischio e non anomalie a basso rischio.

In particolare nel dataset di training, un evento è stato considerato anomalo dal team *Magpies* se mostrava un cambiamento improvviso del rischio tra l'ultima CDM

disponibile con $time_to_tca \geq 2$ e quella finale con $time_to_tca < 1$, passando da basso ad alto rischio o viceversa.

La rete costruita è stata addestrata su coppie di eventi simili (non anomalo, non anomalo) e dissimili (anomalo, non anomalo) testando numerose configurazioni di iperparametri attraverso k-fold cross-validation. Le configurazioni più promettenti sono state usate per costruire i modelli finali che interagiscono con un voto di maggioranza per classificare ciascun evento come anomalo o non anomalo. Infine, in base alla classificazione ottenuta e al valore del rischio nell'ultima CDM disponibile con $time_to_tca \geq 2$, viene assegnato un valore deterministico per la previsione finale \hat{r}_i del modello.

$$\hat{r}_i = \begin{cases} -6.001, & \text{se l'evento } i \text{ viene previsto come non anomalo e } r_{i_last_obs} < -6 \\ -5.35, & \text{se l'evento } i \text{ viene previsto come anomalo e } r_{i_last_obs} < -6 \\ r_{i_last_obs}, & \text{se } r_{i_last_obs} \geq -6 \end{cases}$$

DunderMifflin - 7° classificato - score 0.6276

Il team *DunderMifflin* ha adottato una strategia semplice ma efficace, utilizzando una Random Forest (RF) su tutti i dati e una rete MLP (Multi-Layer Perceptron) dedicata esclusivamente agli eventi previsti ad alto rischio dalla RF [31, 26]. Entrambi i modelli sono stati addestrati su dataset bilanciati, adottando una soglia di -8 per la classificazione con RF (potenzialmente ad alto rischio/basso rischio) e -6 per la classificazione finale basata sul valore del rischio continuo previsto dalla MLP.

Questo approccio, che non sfrutta l'intera sequenza temporale delle CDM di un dato evento ma solo l'ultima CDM disponibile con $time_to_tca \geq 2$, è di fatto stato di grande ispirazione per il lavoro sviluppato in questa tesi. Infatti nonostante la sua apparente semplicità, questo modello ha superato numerosi altri approcci più sofisticati basati su Reti Neurali Ricorrenti (RNN), Convolutional Neural Networks (CNN) o architetture siamesi, ottenendo una posizione di rilievo nella classifica finale.

Sergipm11 - 19° classificato - score 1.1161

Questo team ha esplorato diverse architetture di reti neurali, tra cui RNN e CNN [28, 29], con l'obiettivo di gestire più efficacemente l'informazione contenuta nella serie storica delle CDM di ciascun evento. Tuttavia, nonostante il significativo lavoro svolto su modelli complessi e sulla struttura del dataset, non sono riusciti a raggiungere risultati soddisfacenti. Per questo motivo, non molto prima della chiusura della challenge, questo team ha deciso di convergere su una soluzione a cascata composta da un classificatore e un regressore, simile alla soluzione del team *DunderMifflin*.

3.2 Considerazioni finali sui Modelli della Challenge

In conclusione, l'analisi dei modelli presentati dai team partecipanti alla Challenge ha evidenziato come l'efficacia di una soluzione non dipenda necessariamente dalla complessità architetturale.

In particolare, il modello proposto dal team *DunderMifflin* basato esclusivamente sull'ultima CDM disponibile con $time_to_tca \geq 2$ e su una pipeline relativamente semplice, ha ottenuto risultati competitivi. Questo dimostra che anche soluzioni focalizzate su un singolo istante temporale, se ben progettate e supportate da modelli predittivi efficaci, possono superare approcci più complessi basati sull'intera sequenza temporale di CDM — soprattutto quando questi ultimi non sono affiancati da un supporto decisionale deterministico, come nel caso del team *Magpies*.

La Tabella 3.1, presenta le performance dei primi 10 team partecipanti alla challenge, utilizzate come riferimento per valutare l'efficacia dei modelli sviluppati e proposti in questa tesi (Tabella 4.8).

Team	F₂	MSE_{HR}	Score
sesc	0.733	0.407	0.556
dietmarw	0.765	0.437	0.571
Magpies	0.753	0.441	0.585
Vidente	0.714	0.436	0.610
DeCRA	0.743	0.457	0.615
Valis	0.744	0.467	0.628
DunderMifflin	0.718	0.451	0.628
madks	0.750	0.476	0.634
vhrique	0.764	0.496	0.649
Spacemeister	0.738	0.479	0.649
LRP baseline	0.739	0.513	0.694

Tabella 3.1. Performance dei migliori 10 team partecipanti alla challenge. [32]

Capitolo 4

Approccio metodologico

4.1 Strategia generale: Sviluppo Progressivo di Modelli per la Collision Avoidance

L'approccio metodologico adottato in questo lavoro prende avvio dall'evidenza emersa dall'analisi dei modelli sviluppati dai team partecipanti alla Challenge secondo cui anche modelli con input limitati, se ben progettati, possono ottenere performance competitive. A partire da questa osservazione, sono stati implementati, validati e testati diversi ensemble con strutture e logiche differenti, utilizzando come input esclusivamente l'ultima CDM della serie storica con $time_to_tca \geq 2$, con l'obiettivo di migliorare le prestazioni in termini di F_2 (3.2) e MSE_{HR} (3.3).

Inoltre si è scelto di adottare un approccio modulare allo sviluppo del sistema, progettando componenti indipendenti con interfacce chiare e facilmente sostituibili. Questo ha permesso di sperimentare rapidamente diverse combinazioni di modelli e strategie, rendendo il processo di sviluppo più flessibile e agile.

Prima di presentare il modello finale che ha ottenuto lo score più basso (quindi migliore), analizziamo alcuni esempi significativi dei modelli esplorati durante le fasi di sviluppo.

Random Forest Filtering with MLP Regression

Il primo passo è stato quello di sviluppare un modello simile a quello di *DunderMifflin* composto da una RF [7] per la classificazione iniziale degli eventi come *potenzialmente ad alto rischio* ($\hat{r}_i \geq -8$) o a *basso rischio* ($\hat{r}_i < -8$) ed una rete MLP applicata esclusivamente agli eventi classificati come *potenzialmente ad alto rischio* dalla RF che esegue le previsioni del valore continuo del rischio di questi eventi.

Per tutti gli eventi classificati come a basso rischio dalla RF, così come per quelli a cui la MLP ha assegnato un rischio inferiore a -6 , il valore della previsione è stato impostato a $-6,001$, come previsto dalle regole della challenge, al fine di ridurre l'impatto degli eventuali falsi negativi sul calcolo della metrica MSE_{HR} .

Al modulo di classificazione è stato aggiunto l'iperparametro `probability_filter` (impostato nella maggior parte dei casi a $0,5$), che consente di classificare come ad alto rischio solo gli eventi con probabilità di appartenenza a tale classe superiore a una soglia predefinita.

Component	Hyperparameters and Values
RF	n_estimators: 100
	criterion: entropy
	min_samples_split: 20
	min_samples_leaf: 5
	threshold_classification: -8
MLP	probability_filter: 0.5
	hidden_layers: 1
	n_neurons: 200
	epochs: 200
	batch_size: 256
	learning_rate: 0.004
MLP	activation_function: ELU ($\alpha = 0.4$)
	optimizer: Adam
	loss_function: MSE _{HR}

Tabella 4.1. Iperparametri utilizzati per il modello **RF + MLP**.

SVM Filtering with Deep MLP Regression

A partire da questa prima struttura, si è deciso di incrementare progressivamente la complessità del modello, andando inizialmente a sostituire la RF con una Support Vector Machine (SVM) [9] che lavora su una soglia di classificazione più rigida della RF. Questo modello è stato il primo tra quelli sviluppati a riuscire a battere la baseline LRP ottenendo uno score di 0.6563 sul test set.

Component	Hyperparameters and Values
SVM	kernel: sigmoid
	C: 10
	gamma: 0.01
	threshold_classification: -6
	probability_filter: 0.5
MLP	hidden_layers: 2
	n_neurons_layer_1: 200
	activation_layer_1: LeakyReLU (slope = 0.3)
	n_neurons_layer_2: 100
	activation_layer_2: LeakyReLU (slope = 0.3)
	epochs: 670
MLP	batch_size: 256
	learning_rate: 0.002
	optimizer: Adam
	weight_decay (L2 regularization): 0.004
	loss_function: MSE _{HR}

Tabella 4.2. Iperparametri utilizzati per il modello **SVM + MLP**.

Come previsto, questo questo tipo di approccio ha fornito risultati soddisfacenti anche senza richiedere grossi sforzi per il tuning degli iperparametri, effettuato

con una piccola grid search accompagnata da k -cross validation [20] e seguita da una raffinamento manuale basandosi sulle performance medie ottenute sui fold di validazione costruiti.

Pure MLP Regression

Per completezza, è stato testato anche un approccio basato unicamente sulla regressione mediante rete neurale MLP [16], senza alcun classificatore a monte.

I risultati sul test set hanno evidenziato una forte sensibilità del modello a classificare gli eventi come ad alto rischio (0 falsi negativi), ma al costo di una sovrastima generalizzata del rischio, classificando solo 174 eventi del test set come a basso rischio e penalizzando enormemente l' F_2 .

Component	Hyperparameters and Values
	hidden_layers: 2
	n_neurons_layer_1: 400
	activation_layer_1: LeakyReLU (slope = 0.3)
	n_neurons_layer_2: 200
MLP	activation_layer_2: LeakyReLU (slope = 0.3)
	epochs: 200
	batch_size: 32
	learning_rate: 0.005
	optimizer: Adam
	loss_function: MSE _{HR}

Tabella 4.3. Iperparametri utilizzati per il modello MLP.

Questo risultato evidenzia come un modello regressivo puro, come l'MLP in questa configurazione, fatichi a generalizzare su un dataset fortemente sbilanciato e con un ampio range di valori di rischio. Sono state provate anche tecniche di bilanciamento, ma non hanno portato benefici e in alcuni casi hanno peggiorato le prestazioni, a causa della scarsità di esempi positivi.

Per aumentarne l'efficacia, è necessario porre un classificatore a monte, che filtri gli eventi da sottoporre alla regressione, limitandoli a quelli con un rischio potenzialmente elevato. In questo modo, la MLP può essere addestrata su un campione più informativo e nel testing il classificatore a monte può filtrare la maggior parte dei casi a basso rischio, garantendo che solo i casi più critici arrivino alla fase di regressione.

KNN Filtering with MLP Regression

Per estendere l'analisi, è stato sperimentato anche un approccio alternativo basato su un classificatore K-Nearest Neighbors (KNN) [3].

Questo schema riprende la logica di filtraggio già utilizzata negli altri ensemble con SVM e RF, ma si affida a un criterio di prossimità euclidea tra eventi per classificarli come ad alto o a basso rischio. Il classificatore KNN, così come in tutti i classificatori degli ensemble presentati precedentemente, è stato addestrato su un dataset bilanciato e la rete MLP successiva è stata addestrata esclusivamente sui casi previsti come ad alto rischio dal classificatore.

Component	Hyperparameters and Values
KNN	n_neighbors: 100 weights: uniform metric: euclidean
MLP	hidden_layers: 1 n_neurons: 600 activation_function: ELU ($\alpha = 0.4$) epochs: 200 batch_size: 256 learning_rate: 0.004 optimizer: Adam loss_function: MSE _{HR}

Tabella 4.4. Iperparametri utilizzati per il modello KNN + MLP.

HMM-RNN based Pipeline for Sequential CDM Classification

Un ulteriore esperimento è stato condotto con l'intento di sfruttare la componente temporale presente nelle sequenze di CDM associate a ciascun evento. A tal fine, è stato progettato un modello che integra Hidden Markov Models (HMM) [25], Recurrent Neural Networks (RNN) [24], SVM e MLP.

La pipeline adottata si articola in cinque fasi principali:

- La prima fase è volta ad addestrare due HMM distinte: uno utilizzando le sequenze di CDM relative ad eventi ad alto rischio, e uno utilizzando quelle relative a eventi a basso rischio.
- Successivamente, per ogni evento (sia nel train che nel test set), viene calcolato un indice basato sulla Log-Likelihood Ratio (LLR), ottenuto come differenza tra le log-verosimiglianze della sequenza temporale di CDM secondo i due HMM. Questo valore viene poi utilizzato come nuova feature in input alla fase successiva.
- Viene quindi addestrato una SVM che utilizza sia le feature originali che quella ingegnerizzata basata sulla LLR.
- A seguire, per affinare la distinzione tra veri e falsi positivi, sono state addestrate due RNN distinte: una sui veri positivi e una sui falsi positivi individuati dalla SVM eseguendo le previsioni sul train set. L'obiettivo è stimare, per ogni evento classificato come ad alto rischio dalle fasi precedenti, la probabilità che si tratti di un falso positivo. Qualora questa probabilità superi una soglia prestabilita, l'evento viene riclassificato come a basso rischio.
- Infine, per tutti gli eventi che restano classificati come ad alto rischio, viene applicato un modello MLP per stimare il valore del rischio.

Questo approccio ha portato a risultati interessanti, mostrando un miglioramento complessivo dello score rispetto alle soluzioni basate esclusivamente su Random Forest o SVM, grazie a una netta riduzione dei falsi positivi. Tuttavia, è stato

rilevato un aumento significativo dei falsi negativi, un aspetto particolarmente delicato nell’ambito del collision avoidance. Per questo motivo, nonostante le buone performance generali, si è scelto di non adottare questo modello come soluzione finale, preferendo architetture più semplici focalizzate sull’analisi dell’ultima CDM disponibile.

Component	Hyperparameters and Values
HMM - feature extractor	n_components: 10 n_iter: 500 covariance_type: diag
SVM - main classifier	kernel: sigmoid C: 10 gamma: 0.01 class_weight: balanced probability: True
RNN - false positive filter	hidden_size: 32 activation_function: Sigmoid epochs: 30 optimizer: Adam learning_rate: 0.001
MLP - final regressor	hidden_layers: 2 n_neurons_layer_1: 200 activation_layer_1: LeakyReLU (slope = 0.3) n_neurons_layer_2: 100 activation_layer_2: LeakyReLU (slope = 0.3) epochs: 670 batch_size: 256 learning_rate: 0.002 optimizer: Adam weight_decay (L2 regularization): 0.004 loss_function: MSE _{HR}

Tabella 4.5. Iperparametri utilizzati per il modello **HMM-RNN based**.

MLP Regression with SVM-Guided Training and Post-Prediction Correction

A partire dalla pipeline SVM + MLP già validata, è stata avviata una fase di sperimentazione più ampia volta a esplorare l’efficacia di ensemble di più complessi, con l’obiettivo di aumentare la precisione delle stime e ridurre la frequenza di falsi positivi e soprattutto dei falsi negativi.

Un primo esempio in questa direzione è rappresentato da un ensemble composto da 3 modelli:

- una prima SVM per dare al training dell’MLP successiva solo gli eventi previsti ad alto rischio;
- una rete MLP incaricata della stima del rischio per tutti gli eventi del test set, ma addestrata solo su eventi previsti ad alto rischio dalla fase precedente e sugli eventi previsti a basso rischio che sono support vector della SVM iniziale;

- una seconda SVM addestrata sugli stessi dati dell'MLP per confermare le previsioni superiori alla soglia dell'alto rischio e correggerle qualora venissero classificate a basso rischio da quest'ultima fase.

Questa architettura, pur mantenendo una struttura relativamente semplice, introduce una logica utile per ridurre i falsi positivi e rappresenta un primo passo verso la progettazione di ensemble più sofisticati, che verranno approfonditi nelle sezioni successive.

Component	Hyperparameters and Values
SVM - training filter	<pre> kernel: sigmoid C: 10 gamma: 0.01 threshold_classification: -6 probability_filter: 0.5 </pre>
MLP - specialized regressor	<pre> hidden_layers: 2 n_neurons_layer_1: 512 activation_layer_1: LeakyReLU (slope = 0.3) n_neurons_layer_2: 256 activation_layer_2: LeakyReLU (slope = 0.3) epochs: 670 batch_size: 256 learning_rate: 0.002 optimizer: Adam weight_decay (L2 regularization): 0.004 loss_function: MSE_{HR} </pre>
SVM - false positive filter	<pre> kernel: rbf C: 5 gamma: 0.01 class_weight: balanced threshold_classification: -6 probability_filter: 0.5 </pre>

Tabella 4.6. Iperparametri utilizzati per il modello **SVM + MLP + SVM**.

Multiphase Ensemble with Weak Multiclass SVM and Final Voting.

Un ulteriore ensemble sviluppato è stato progettato per combinare classificazione e regressione in quattro fasi distinte, ciascuna con un ruolo specifico nella selezione e raffinamento delle previsioni dei vari modelli coinvolti. Nella prima fase, vengono costruiti tre classificatori binari SVM per distinguere tre fasce di rischio (basso, medio, alto), utilizzando soglie derivate da percentili dinamici del training set al fine di garantire campioni sufficienti su ciascuna delle tre classi considerate.

Una volta individuati i campioni in ciascuna fascia si procede a costruire 3 dataset bilanciati per ogni coppia di possibili classificazioni (basso vs medio, medio vs alto, basso vs alto). Le tre SVM allenate sulle tre diverse coppie concordano la previsione finale con un voto di maggioranza attuando una selezione intenzionalmente debole, allo scopo di identificare un sottoinsieme di eventi potenzialmente ad alto rischio.

Nella seconda fase viene addestrata una seconda SVM esclusivamente su eventi delle fasce medio-alto rischio individuate nella prima fase. Il suo scopo è affinare la selezione, riducendo i falsi positivi e individuando i casi effettivamente più critici.

Segue la fase 3, in cui una rete neurale MLP con due hidden layer (attivazioni SiLU e Tanh) viene addestrata a stimare il rischio in modo continuo. L'addestramento è mirato esclusivamente su eventi classificati ad alto rischio dalle fasi precedenti, permettendo alla rete di specializzarsi sui casi più complessi.

Infine, la quarta ed ultima fase applica un ulteriore filtro: tre classificatori (Extra Trees Classifier [15], Logistic Regressor [17], Gaussian Naive Bayes [5]), allenati su tutti gli eventi veramente ad alto rischio del training set e su un numero equo di eventi a basso rischio che soddisfano $\text{last_cdm_risk} \geq \text{sampling_lower_bound}$, eseguono un check su tutti gli eventi del test set classificati ad alto rischio dalle fasi precedenti: se almeno due su tre classificano un certo evento come a basso rischio, esso viene riclassificato di conseguenza, con lo scopo di ridurre ulteriormente i falsi positivi.

Component	Hyperparameters and Values
SVM – multiclass risk separator	<pre> kernel: sigmoid C: 10 gamma: 0.01 thresholds: percentile 85 and 98 probability_filter: 0.5 </pre>
SVM – false positive filter	<pre> kernel: rbf C: 8 gamma: 0.008 threshold_classification: -6 probability_filter: 0.5 </pre>
MLP – specialized regressor	<pre> hidden_layers: 2 n_neurons: 512, 256 activation_functions: SiLU, Tanh dropout: 0.003, 0.006 epochs: 670 batch_size: 512 learning_rate: 0.002 optimizer: Adam loss_function: MSE_HR weight_decay (L2 regularization): 0.004 </pre>
final voting	<pre> Extra Trees: n_estimators = 500, criterion = entropy Logistic Regression: solver = lbfgs, penalty = 12 Gaussian Naive Bayes: default settings threshold_classification: -6 sampling_lower_bound: -10 probability_filter: 0.35 </pre>

Tabella 4.7. Iperparametri utilizzati per il modello: **Multiclass Ensemble**.

4.2 Considerazioni finali sui Modelli Sviluppati

Dopo aver analizzato e testato numerosi modelli, si è deciso di suddividere lo sviluppo in due rami paralleli. In particolare, ci si è concentrati separatamente sul modulo

di classificazione e su quello di regressione, con l’obiettivo di progettare ensemble più articolati e specializzati. Entrambi i moduli sono stati progettati per interagire in modo coerente, con particolare attenzione all’interfaccia tra classificatore (che verrà presentato in questa tesi) e regressore (presentato nella tesi del collega Rocco Salvatore).

L’idea centrale, comune a entrambi i filoni di lavoro, è stata quella di sfruttare le proprietà della Support Vector Machine, distinguendo tra:

- campioni fuori dal margine, per cui la separazione è netta e la classificazione più affidabile
- campioni all’interno del margine, considerati più ambigui e difficili da classificare.

Questa distinzione ha permesso di addestrare modelli ancora più mirati, capaci di specializzarsi su sottoinsiemi di eventi con livelli diversi di incertezza, e ha rappresentato un passaggio chiave nell’evoluzione del modello finale.

La Tabella 4.8 riassume le principali metriche di valutazione per ciascuno dei modelli sviluppati, i valori sono calcolati sul test set ufficiale della Challenge.

Il modello finale, che ha ottenuto le migliori prestazioni, sarà presentato nei capitoli successivi.

Model	TP	TN	FP	FN	F2	MSE_HR	Score
Final Model	137	1828	189	13	0.740	0.413	0.559
Multiclass Ensemble	127	1867	150	23	0.724	0.463	0.639
HMM-RNN based	124	1903	114	26	0.740	0.476	0.643
SVM + MLP	132	1804	213	18	0.698	0.458	0.656
SVM + MLP + SVM	130	1831	186	20	0.709	0.481	0.678
RF + MLP	133	1841	176	17	0.732	0.524	0.716
KNN + MLP	132	1471	546	18	0.516	0.480	0.929
MLP	150	174	1843	0	0.289	0.465	1.609
LRP baseline	115	1954	63	35	0.739	0.513	0.694

Tabella 4.8. Performance dei modelli sviluppati

Capitolo 5

Preprocessing del dataset per l’Ensemble di Classificazione

Il primo passo nello sviluppo dei modelli ha riguardato la costruzione del dataset di training a partire dai dati forniti da ESA durante la *Spacecraft Collision Avoidance Challenge*.

Come già descritto nella Sezione 2.1, è stata effettuata una prima scrematura degli eventi per selezionare solo quelli contenenti le informazioni necessarie sia per definire l’input (almeno una CDM con $time_to_tca \geq 2$ giorni) sia per assegnare correttamente il valore target (una CDM con $time_to_tca < 1$). Questo filtraggio ha permesso di costruire un dataset coerente con gli obiettivi della Challenge e idoneo allo sviluppo di modelli supervisionati.

Prima di procedere all’addestramento, il dataset così ottenuto è stato analizzato nella fase di *Feature Selection*, con l’obiettivo di individuare le variabili più rilevanti per la previsione del rischio di collisione. Tale fase ha rappresentato un passaggio fondamentale per migliorare l’efficienza e la qualità dei modelli sviluppati, riducendo enormemente la dimensionalità del problema.

5.1 Ingegnerizzazione e selezione preliminare delle feature

In questa sezione vengono presentate le feature analizzate in fase di *Feature Selection*, suddivise in due categorie principali: le feature fornite direttamente da ESA nel dataset ufficiale, e un insieme di feature ingegnerizzate, costruite appositamente per cercare di estrarre informazioni aggiuntive dalla serie storica delle CDM ricevute per ciascun evento.

Il punto di partenza per le analisi è stato rappresentato dalle 20 feature ritenute più informative da ESA al termine della Challenge. Da queste sono state selezionate tutte quelle prive di valori NaN, che hanno costituito una base solida e condivisa su per costruire modelli robusti e ridurre la dimensionalità del problema.

Infine sono state ingegnerizzate nuove feature per arricchire la rappresentazione informativa delle singole CDM: alcune di esse mirano a catturare la frequenza di ricezione delle CDM precedenti, altre sintetizzano informazioni statistiche sulla

sequenza, come il delta di rischio rispetto alla prima CDM, media, varianza e comportamento locale nei due giorni precedenti ciascuna osservazione.

Feature	Rank	Mean	Std. dev.
<i>risk</i>	1	29.275	9.557
<i>max_risk_scaling</i>	2	22.544	8.979
<i>mahalanobis_distance</i>	3	3.261	1.675
<i>c_sigma_t</i>	4	3.000	1.715
<i>max_risk_estimate</i>	5	2.624	1.367
<i>c_sigma_rdot</i>	6	2.191	1.369
<i>miss_distance</i>	7	2.089	1.112
<i>c_position_covariance_det</i>	8	1.778	1.066
<i>c_sigma_n</i>	9	1.312	0.625
<i>time_to_tca</i>	10	1.236	0.517
<i>c_sigma_r</i>	11	1.177	0.739
<i>c_obs_used</i>	12	1.164	0.554
<i>c_sigma_ndot</i>	13	0.964	0.437
<i>relative_position_n</i>	14	0.954	0.754
<i>c_recommended_od_span</i>	15	0.945	0.423
<i>relative_position_r</i>	16	0.835	0.440
<i>c_sedr</i>	17	0.779	0.486
<i>SSN</i>	18	0.773	0.372
<i>c_crdot_t</i>	19	0.718	0.468
<i>relative_speed</i>	20	0.699	0.400

Tabella 5.1. Feature ranking, media e deviazione standard delle 20 variabili più importanti secondo l'analisi ESA post-competizione. [32]

Si riporta di seguito l'elenco delle feature considerate, suddivise tra quelle selezionate dalla Tabella 5.1 (prive di valori NaN) e quelle appositamente ingegnerizzate, accompagnate da una breve descrizione del loro significato¹.

Tutte queste variabili sono state poi analizzate e valutate tramite studi di correlazione e analisi d'importanza prima di definire il set finale di feature adottato nei modelli.

Infine, in linea con l'impostazione dei modelli sviluppati, è stata mantenuta una sola osservazione per ciascun evento, selezionando esclusivamente l'ultima CDM disponibile con $time_to_tca \geq 2$ e eliminando tutte le altre.

Feature	Description
<i>risk</i>	Self-computed value at the epoch of each CDM [base 10 log]
<i>max_risk_scaling</i>	Scaling factor used to compute maximum collision probability
<i>mahalanobis_distance</i>	Statistical distance between predicted trajectories, accounting for orbital uncertainties [2]
<i>max_risk_estimate</i>	Maximum collision probability obtained by scaling combined covariance
<i>miss_distance</i>	Relative position between chaser and target at tca [m]
<i>c_position_covariance_det</i>	Determinant of the chaser's covariance (volume)
<i>time_to_tca</i>	Time interval between CDM creation and time-of-closest approach [days]
<i>relative_position_n</i>	Relative position between chaser and target: normal (cross-track) [m]
<i>relative_position_r</i>	Relative position between chaser and target: radial [m]
<i>c_sedr</i>	Specific energy dissipation rate of the chaser [W/kg]
<i>relative_speed</i>	Relative speed between chaser and target at tca [m/s]

Tabella 5.2. Feature originali selezionate dal dataset ESA, prive di valori NaN.

¹Per maggiori dettagli e per le descrizioni complete delle feature presenti nel dataset originale, si rimanda alla pagina ufficiale della Challenge: Collision Avoidance Challenge – Data – Kelvins.

Feature	Description
<i>n_CDM</i>	Number of previous CDMs for the same event
<i>risk_mean</i>	Mean risk across previous CDMs for the same event
<i>risk_std</i>	Standard deviation of risk across previous CDMs
<i>delta_p</i>	Difference between current risk and the first CDM risk of the event
<i>min_risk, max_risk</i>	Minimum and maximum risk observed so far for the event
<i>risk_range</i>	Risk variability: current max - min among previous CDMs
<i>risk_range_min</i>	Difference between current risk and minimum observed risk
<i>risk_range_max</i>	Difference between current risk and maximum observed risk
<i>n_CDM_last_2_days</i>	Number of previous CDMs with $2 \leq \text{time_to_tca} \leq 4$
<i>Risk_Mean_last_2_days</i>	Mean risk of previous CDMs with $2 \leq \text{time_to_tca} \leq 4$
<i>Risk_STD_last_2_days</i>	Standard deviation of risk from previous CDMs with $2 \leq \text{time_to_tca} \leq 4$

Tabella 5.3. Feature ingegnerizzate a partire dalla serie storica delle CDM.

5.2 Selezione delle feature e analisi di importanza

Per identificare il sottoinsieme ottimale di variabili da utilizzare nei modelli predittivi, è stata condotta un'analisi dettagliata sulle feature introdotte nella Sezione 5.1. In primo luogo, sono state esplorate le distribuzioni di ciascuna variabile (compreso il target *last_cdm_risk*) sul training set, al fine di verificare la presenza di asimmetrie o anomalie.

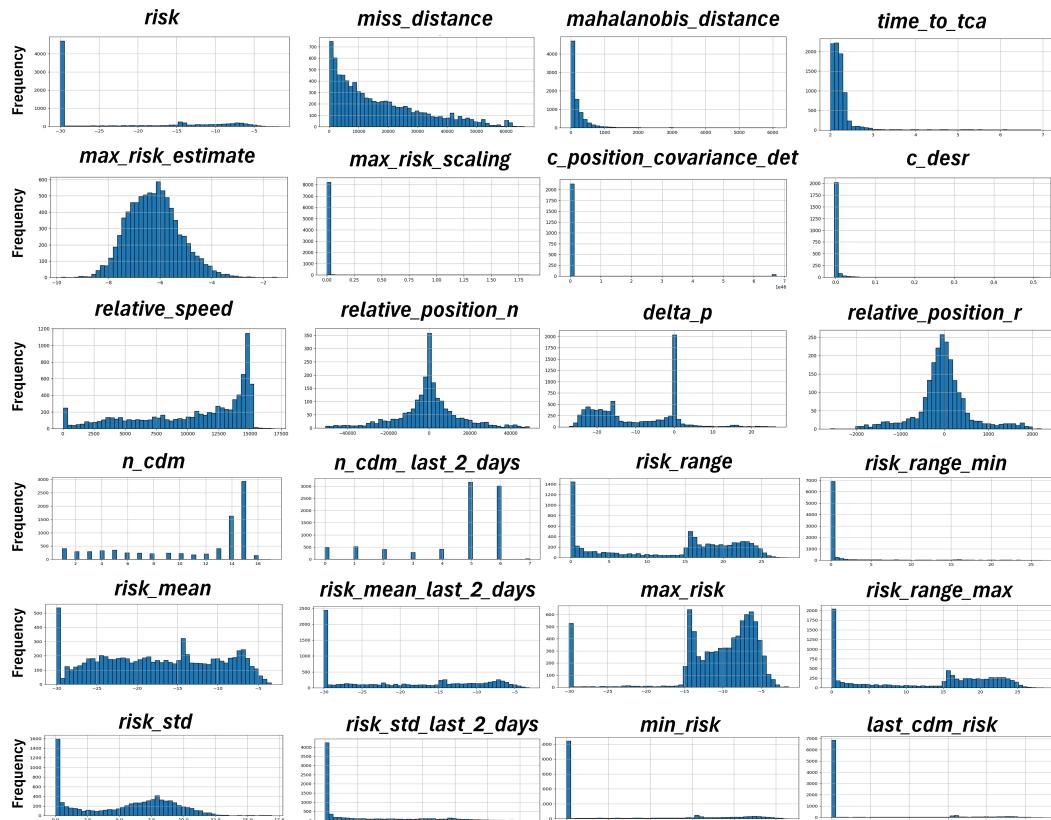


Figura 5.1. Distribuzione delle variabili analizzate

Successivamente, è stata condotta un'analisi di correlazione tra le variabili, con l'obiettivo di identificare le feature più informative rispetto al target *last_cdm_risk* e rilevare eventuali dipendenze o ridondanze tra le stesse. Come visibile in Figura 5.2, la variabile *risk* risulta fortemente correlata con molte delle feature ingegnerizzate derivate dalla sua evoluzione temporale, tra cui *min_risk*, *risk_mean*, *risk_mean_last_2_days* e *delta_p*. Di conseguenza, si è resa necessaria una scelta accurata su quali feature mantenere in fase di modellazione.

Le feature ingegnerizzate, inizialmente testate nei modelli, non hanno migliorato lo score in fase di validazione e hanno anzi contribuito ad aumentare i falsi negativi. Per questo motivo, si è deciso di conservare esclusivamente la feature *risk*, eliminando tutte le sue derivate che erano state ingegnerizzate. Questa scelta consente di evitare sovrapposizioni informative e di mantenere una descrizione più fedele e aggiornata del rischio contenuto nell'ultima CDM disponibile.

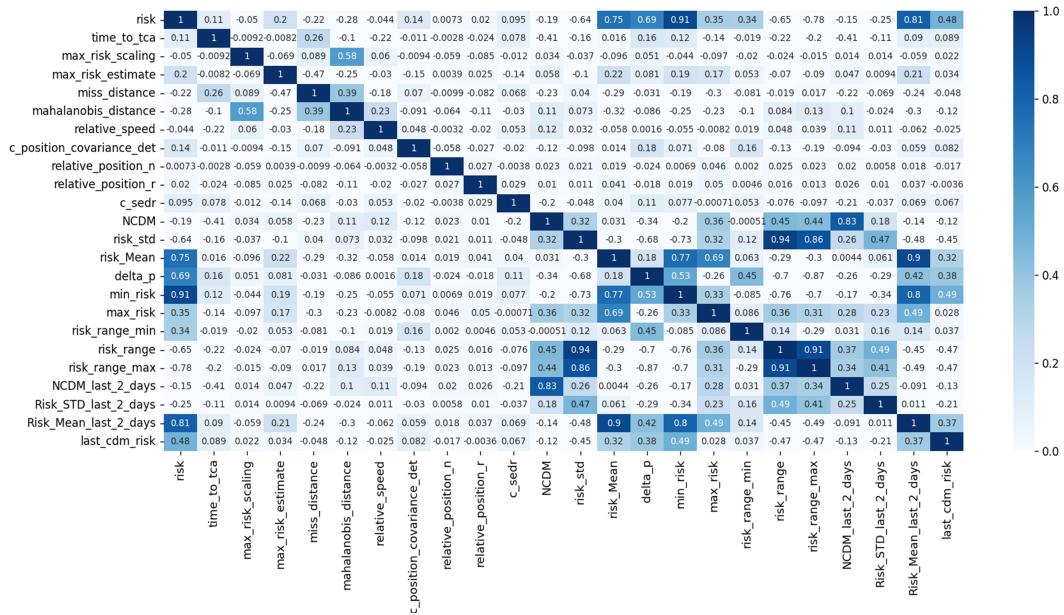


Figura 5.2. Heatmap delle variabili analizzate

Infine, dopo aver eseguito una serie di analisi con *Feature Importance* [30] di Random Forest è stato selezionato un sottoinsieme ristretto di feature da utilizzare nei modelli successivamente sviluppati. Tali feature sono riportate in Tabella 5.4 secondo l'ordine di importanza stabilito da quest'ultima analisi.

Feature	Rank	Description
<i>risk</i>	1	Self-computed value at the epoch of each CDM [base 10 log]
<i>max_risk_scaling</i>	2	Scaling factor used to compute maximum collision probability
<i>max_risk_estimate</i>	3	Maximum collision probability obtained by scaling combined covariance
<i>relative_speed</i>	4	Relative speed between chaser and target at TCA [m/s]
<i>mahalanobis_distance</i>	5	Mahalanobis distance between chaser and target
<i>miss_distance</i>	6	Relative position between chaser and target at TCA [m]
<i>time_to_tca</i>	7	Time interval between CDM creation and Time-of-Closest Approach [days]

Tabella 5.4. Feature finali selezionate.

Tale selezione ha permesso di ridurre la complessità del modello, mantenendo al contempo un buon potere predittivo: le feature selezionate rappresentano infatti un buon compromesso tra interpretabilità, rilevanza statistica e robustezza rispetto alle variazioni riscontrate tra training e test set della challenge, già evidenziate nelle Figure 2.5 e 2.6 riportate nella Sezione 2.2.

5.3 Tecnica di Bilanciamento tramite Campionamento a Bin

Uno degli aspetti più critici nella progettazione di un modello predittivo con i dataset della *Spacecraft Collision Avoidance Challenge* è la gestione dell'estremo sbilanciamento nel training set, in particolare tra eventi ad alto rischio (solo 66) e quelli a basso rischio (oltre 8.000). Per affrontare questa sfida, nel presente lavoro è stato sviluppato un metodo di campionamento mirato, volto a costruire dataset di addestramento bilanciati, che preservassero la varietà del dominio a basso rischio evitando nel contempo una prevalenza di campioni inutili.

Il metodo si basa sulla creazione di intervalli di rischio (bin) all'interno della fascia del basso rischio $[-30, -6]$, tanti quanti sono il numero di eventi ad alto rischio, in modo tale da poter estrarre nel migliore dei casi un evento a basso rischio da ciascun bin individuato.

Vediamo come funziona nel dettaglio con un esempio a scopo esemplificativo: supponiamo di avere un dataset di training con soli $N = 6$ eventi ad alto rischio (cioè con $\text{last_cdm_risk} \geq -6$)

Si selezionano tutti i $N = 6$ campioni ad alto rischio disponibili nel training set (Figura 5.3)

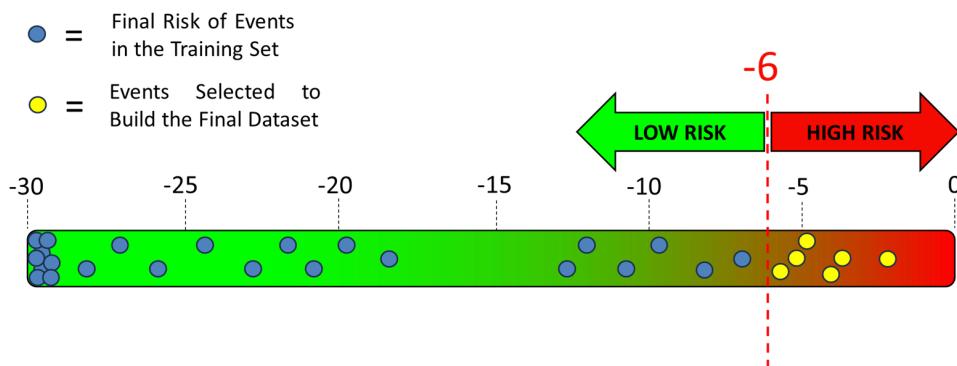


Figura 5.3. esempio bilanciamento con bin: selezione degli eventi ad alto rischio

Si suddivide l'intervallo $[-30, -6]$ in N bin di ampiezza uniforme e per ciascuno di essi si seleziona a caso un evento a basso rischio con valore di last_cdm_risk appartenente al bin preso in considerazione. Una volta esplorati tutti i bin, se alcuni di questi erano vuoti (come ad esempio il bin $[-18, -14]$ in Figura 5.4), si procede iterativamente a cercare un evento a basso rischio a partire dal bin più vicino a -6 ed escludendo gli eventi già selezionati, fino a quando non se ne seleziona un numero pari a N .

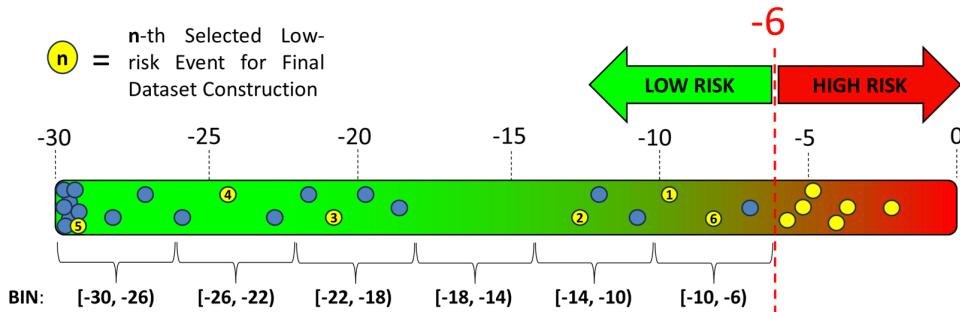


Figura 5.4. esempio bilanciamento con bin: selezione degli eventi a basso rischio

Questo approccio consente di ottenere una copertura omogenea dei valori di *last_cdm_risk* tra gli eventi a basso rischio selezionati per l'addestramento, evitando campionamenti ridondanti nei range estremamente bassi.

Per migliorare ulteriormente l'efficacia del bilanciamento, è stata introdotta una soglia dinamica minima su *last_cdm_risk*, progettata per escludere preventivamente i campioni meno informativi, come ad esempio quelli con *last_cdm_risk* = -30. Come mostrato nella distribuzione in Figura 2.3 della Sezione 2.2, tali casi costituiscono una porzione significativa del dataset e se questi campioni venissero sovraccampionati (a causa della mancanza di esempi utili in alcuni bin) si rischierebbe di ottenere un dataset dominato da eventi schiacciati su valori di rischio molto bassi, poveri di informazione discriminativa e quindi poco utili per l'addestramento efficace dei modelli.

Per capire come questa soglia si integra alla logica dei bin vediamo l'esempio precedente nel caso in cui avessimo voluto eliminare a prescindere tutti gli eventi con *last_cdm_risk* < -24 (Figura 5.5)

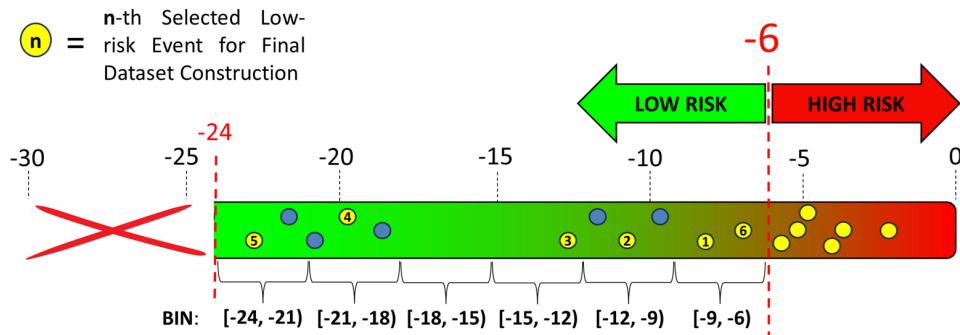


Figura 5.5. esempio bilanciamento con bin: selezione degli eventi a basso rischio con soglia

Questa soglia è stata generalmente impostata a -29.99 nelle fasi in cui si desiderava semplicemente escludere l'accumulo di casi estremi mentre settata a valori più alti (es. -8) quando si voleva costruire un modello più focalizzato sulla zona di confine tra basso e alto rischio. Questa flessibilità è risultata particolarmente utile nella progettazione modulare della pipeline, permettendo di adattare il dataset di training alle esigenze specifiche dei diversi modelli coinvolti.

5.4 Tecnica di normalizzazione adottata

Per garantire un apprendimento più stabile ed efficiente, tutti i modelli sviluppati nell'ambito di questo lavoro hanno una fase di normalizzazione dei dati in input. In particolare, si è scelto di applicare la scalarizzazione con **Min-Max Scaler** [27], che trasforma ogni feature numerica in un intervallo compreso tra 0 e 1, preservandone la distribuzione relativa.

La normalizzazione è stata sempre eseguita separatamente per il dataset di training e per quello di test, utilizzando esclusivamente le statistiche calcolate sul training set. Inoltre, le statistiche per lo scaling (valore minimo e massimo di ciascuna feature) sono state calcolate su subset del training filtrati e bilanciati, a seconda del contesto applicativo e dell'obiettivo del modello.

5.5 Considerazioni sull'uso di dati sintetici

Nonostante nelle linee guida della Challenge, l'utilizzo di dati sintetici era fortemente sconsigliato, durante la fase di sviluppo sono state testate diverse tecniche di oversampling, tra cui SMOTE [8] e altre varianti, con l'obiettivo di generare nuovi eventi ad alto rischio e migliorare la fase di addestramento. Tuttavia, come previsto, l'introduzione di dati sintetici ha sistematicamente peggiorato le performance dei modelli confermando che, in contesti come la collision avoidance, l'aggiunta artificiale di esempi può facilmente compromettere la coerenza del problema.

Si è scelto quindi di non usare dati sintetici per l'addestramento dei modelli sviluppati in questa tesi, concentrandosi invece sul campionamento controllato degli eventi (illustrato nella Sezione 5.3) per ridurre lo sbilanciamento delle classi e allo stesso tempo uniformare la distribuzione dei valori del target *last_cdm_risk*.

Capitolo 6

Ensemble di Classificazione

Questo capitolo descrive nel dettaglio il modulo di classificazione implementato e integrato nel modello finale con l'obiettivo di identificare gli eventi ad alto rischio con almeno due giorni di anticipo dal TCA. Il modello è stato progettato con particolare attenzione alla riduzione dei falsi negativi, ritenuti più critici in ambito operativo, e adotta una pipeline multilivello in grado di affinare progressivamente le decisioni.

6.1 Obiettivo e struttura dell'Ensemble di Classificazione

Per validare in maniera indipendente il modulo di classificazione non è stato possibile ottimizzare direttamente la metrica F_2 , infatti la massimizzazione diretta della F_2 può, in alcuni scenari, portare a strategie troppo poco conservative, con la tendenza a ridurre al minimo il numero di *False Positive (FP)* anche al costo di introdurre un numero significativo di *False Negative (FN)*.

Per questo motivo il modello qui proposto è stato progettato con un'attenzione prioritaria alla massimizzazione del *Recall* sugli eventi ad alto rischio cercando comunque di mantenere la *Precision* a livelli soddisfacenti al fine di tenere sotto controllo il valore dell' F_2 .

$$\text{Recall} = \frac{TP}{TP + FN}, \quad \text{Precision} = \frac{TP}{TP + FP}, \quad F_2 = \frac{(1 + 2^2) \cdot \text{Precision} \cdot \text{Recall}}{2^2 \cdot \text{Precision} + \text{Recall}}$$

Per raggiungere questo obiettivo, è stato progettato e validato un ensemble multilivello costituito da quattro fasi, ciascuna con un ruolo specifico nella progressiva raffinazione delle previsioni.

FASE 1 - SVM iniziale per separazione inside/outside

Nella prima fase della pipeline una *Support Vector Machine* (SVM) viene addestrata su un dataset bilanciato tramite la tecnica del campionamento con bin (illustrato nella Sezione 5.3). Sebbene questa SVM sia formalmente addestrata per classificare gli eventi tra alto e basso rischio, il suo scopo non è fornire una previsione finale, ma identificare la posizione relativa degli eventi rispetto all'iperpiano di separazione,

calcolandone la distanza geometrica e distinguendoli in due categorie: *inside* (distanza ≤ 1) e *outside* (distanza > 1).

Una volta definito l'iperpiano di separazione, lo stesso criterio viene applicato all'intero dataset di training e al test set, assegnando a ciascun evento l'etichetta *inside* o *outside* in base alla distanza calcolata, distinzione che consente di trattare separatamente gli eventi ambigui e quelli classificabili con maggiore certezza nelle fasi successive.

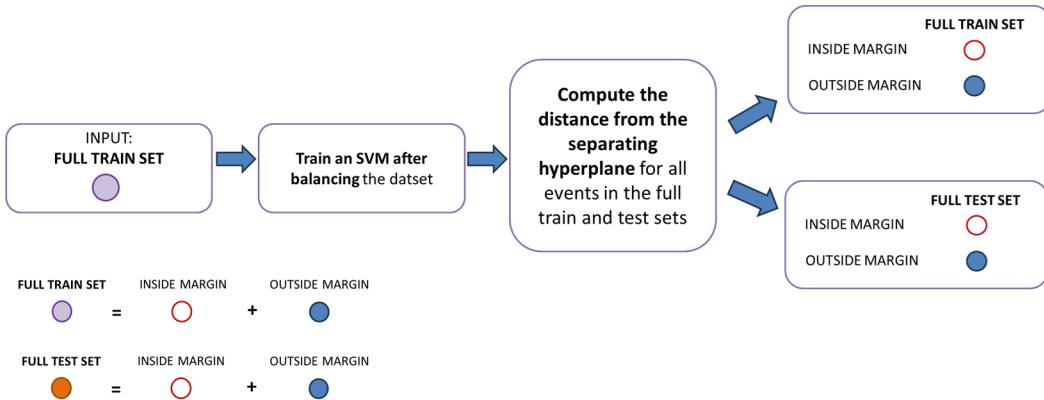


Figura 6.1. FASE 1 - SVM iniziale per separazione inside/outside

FASE 2 - Griglia di 28 SVM con voto combinato

Nella seconda fase si punta alla massima generalizzazione, allenando un totale di **56 modelli SVM**, suddivisi in due flussi paralleli:

- **28 SVM** allenate solo su eventi **outside**
- **28 SVM** allenate solo su eventi **inside**.

Le 28 configurazioni derivano da 28 combinazioni di kernel e iperparametri scelti in un intervallo ritenuto valido. I dataset utilizzati per l'addestramento di ciascuna SVM non sono bilanciati: si imposta invece `class_weight = balanced` per compensare l'asimmetria, assegnando un peso maggiore alla classe minoritaria.

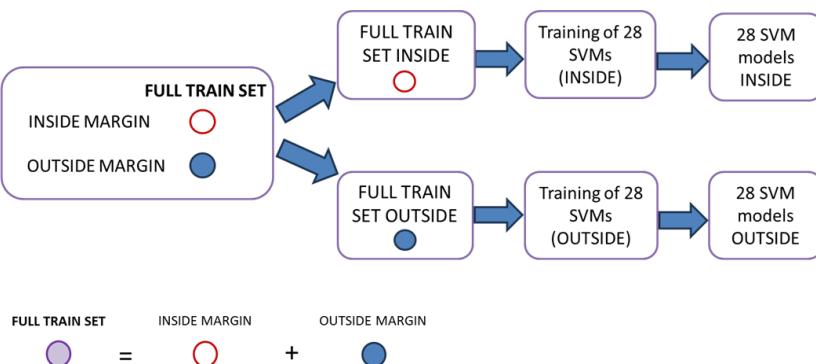


Figura 6.2. TRAIN FASE 2 - Addestramento delle 56 SVM

Ogni coppia di SVM *inside-outside* condivide la stessa configurazione e le loro predizioni vengono combinate applicando la seguente regola:

$$\text{combined}_{preds} = \begin{cases} \text{low risk} & \text{if } \text{SVM}_{in} = \text{low risk} \text{ OR } \text{SVM}_{out} = \text{low risk} \\ \text{high risk} & \text{if } \text{SVM}_{in} = \text{high risk} \text{ AND } \text{SVM}_{out} = \text{high risk} \end{cases}$$

Un aspetto peculiare di questa fase è che ogni SVM viene in realtà addestrata due volte: in un primo passaggio, la SVM viene allenata sull'intero sottoinsieme (*inside* o *outside*) per identificare i *support vectors* (*SV*), ovvero i punti più critici per la classificazione. Una volta individuati, il modello viene riallentato esclusivamente su questo sottoinsieme di *SV*, dopo aver riscalato i dati originali utilizzando le sole statistiche dei support vector stessi. Il test set viene successivamente trasformato utilizzando queste stesse statistiche, e infine vengono generate le previsioni dell'SVM corrente.

Questa strategia consente di ottenere modelli estremamente specializzati e focalizzati sulle aree più complesse dello spazio decisionale. Inoltre, associando a ciascuna coppia di SVM un diverso seed randomico e una configurazione di iperparametri differente, i *SV* selezionati variano sensibilmente da una SVM all'altra. Singolarmente, ciascuna delle previsioni combinate delle coppie di SVM può risultare poco affidabile, ma è proprio la loro diversità a rendere efficace il sistema di voto aggregato. La classificazione finale avviene solo se si raggiunge un consenso elevato: almeno l'80% delle coppie devono classificare l'evento come a basso rischio, oppure almeno il 70% come ad alto rischio. Gli eventi che soddisfano uno dei due criteri sono considerati *eventi sicuri*, e vengono classificati direttamente.

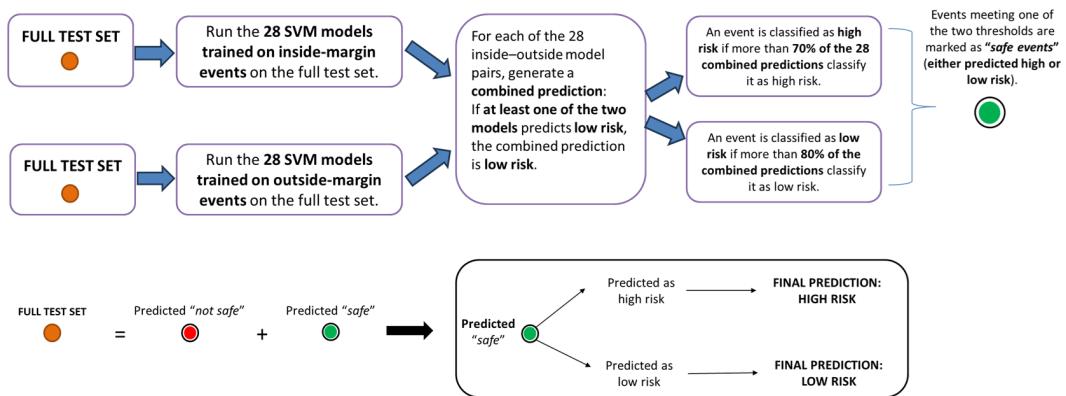


Figura 6.3. TEST FASE 2 - Voto combinato dei 56 modelli SVM

FASE 3 - Voto con classificatori specializzati (*inside/outside*)

I rimanenti eventi, considerati *non sicuri* dalla Fase 2, vengono gestiti da due flussi separati, uno per gli eventi *inside* e uno per quelli *outside*.

In ciascun flusso, vengono addestrati tre classificatori supervisionati (Extra Trees, Gradient Boosting [14], Gaussian Naive Bayes) su dataset bilanciati costruiti selezionando tutti gli eventi ad alto rischio del flusso d'appartenenza (*inside/outside*) e un numero equivalente di eventi a basso rischio appartenenti allo stesso insieme selezionati con la tecnica di bilanciamento con bin.

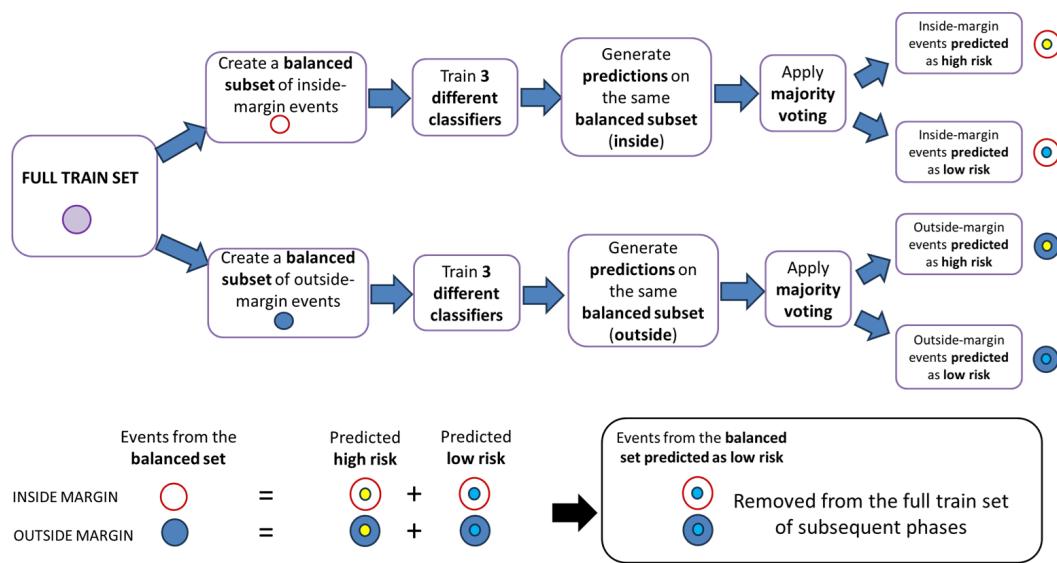


Figura 6.4. TRAIN FASE 3 - classificatori specializzati (inside/outside)

In fase di previsione, se la probabilità che un evento appartenga alla classe ad alto rischio supera una soglia prefissata, il classificatore corrispondente lo etichetta come tale. La previsione finale per ciascun evento si basa su un voto di maggioranza tra i tre modelli del rispettivo flusso (*inside* o *outside*).

Gli eventi dei training set bilanciati che vengono classificati come *low risk* da almeno due classificatori su tre vengono rimossi dal dataset di addestramento della fase successiva. Questa scelta risponde a più obiettivi strategici:

- Evitare la sovrapposizione di esempi semplici: escludendo gli eventi negativi già riconosciuti correttamente da più modelli, si prevede che la fase successiva venga addestrata su esempi troppo facili o ridondanti rispetto a quelli già visti dalla Fase 3 della pipeline, preservando così la diversità informativa tra gli eventi a basso rischio.
- Rafforzare la sensibilità agli eventi critici: la rimozione degli eventi ad alto rischio che, pur presenti nel training, non sono stati riconosciuti dalla maggioranza dei classificatori, consente di eliminare outlier o casi rumorosi, che potrebbero confondere la fase finale di classificazione e peggiorarne la generalizzazione.
- Evitare uno sbilanciamento inverso nella fase 4: il filtraggio viene applicato solo agli eventi del dataset bilanciato utilizzato nella fase corrente, e non all'intero training set della pipeline complessiva, per evitare che la successiva fase 4 si ritrovi con un numero di eventi positivi potenzialmente maggiore di quelli negativi. Ciò garantisce che la varietà dei casi a basso rischio rimanga ampia e che l'equilibrio tra classi venga mantenuto anche nelle fasi più avanzate dell'addestramento.

Dopo l'addestramento, i tre classificatori specializzati per ciascun flusso (*inside* e *outside*) vengono utilizzati per effettuare previsioni solo sugli eventi *non sicuri* del

test set. In particolare il flusso inside valuta gli eventi precedentemente etichettati come *inside* dalla Fase 1 e *non sicuri* dalla Fase 2, mentre il flusso outside valuta gli eventi del test set *outside*, anch'essi etichettati come *non sicuri* dalla fase precedente.

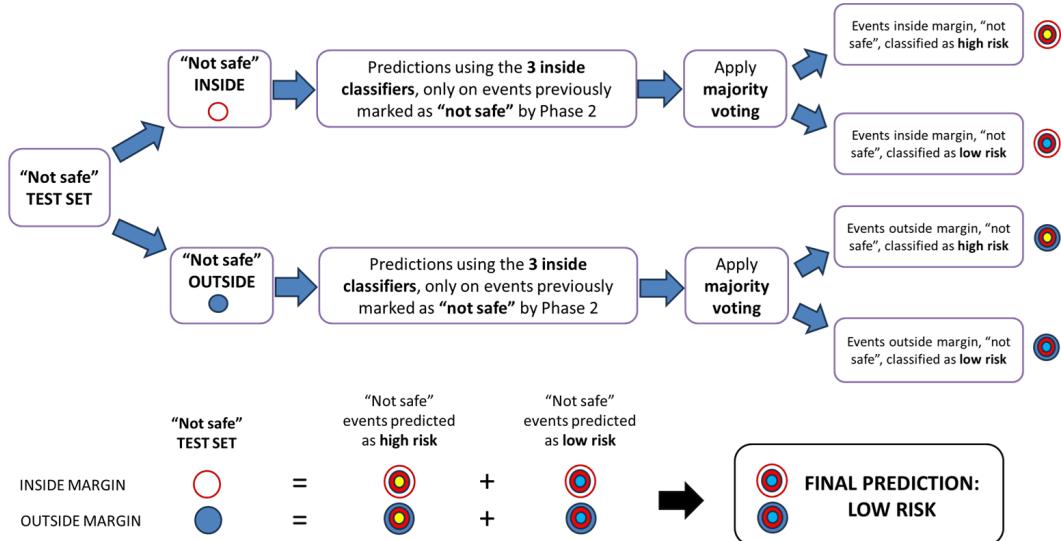


Figura 6.5. TEST FASE 3 - Voto con classificatori specializzati (inside/outside)

FASE 4 - Voto finale e riduzione dei falsi positivi

In quest'ultima fase, gli stessi tre modelli (ET, GBT, GNB) vengono riaddestrati su un nuovo dataset bilanciato che include:

- Tutti gli eventi ad alto rischio del training set non scartati nella fase precedente
- Nuovi eventi a basso rischio, selezionati dal dataset di train originale tramite il metodo di campionamento con bin (escludendo quelli scartati dalla fase precedente).

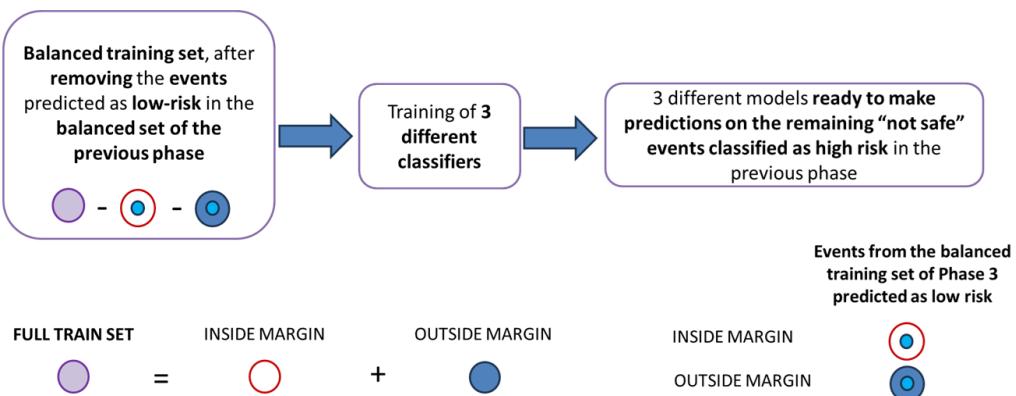


Figura 6.6. TRAIN FASE 4 - Votatori finali

Lo scopo di questa fase è effettuare una classificazione più rigida e ridurre i falsi positivi. Il voto di maggioranza in questo caso viene applicato solo agli eventi del test set classificati come *non sicuri* dalla fase due e previsti ad *alto rischio* dalla fase tre.

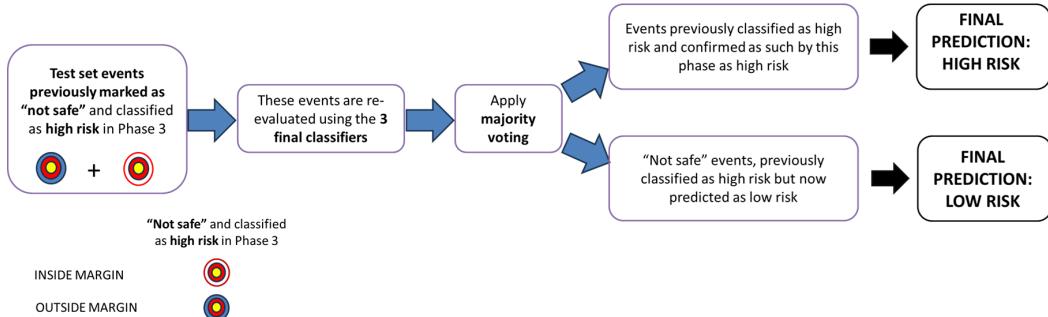


Figura 6.7. TEST FASE 4 - Voto finale e riduzione dei falsi positivi

6.2 Pseudocodice dell'Ensemble di Classificazione

Algorithm 1 Fase 1

```

1: Input:  $D_{\text{class\_train}} = \{(x_i, y_i)\}_{i=1}^N$ ,  $y_i \in \{\text{low}, \text{high}\}$ ,  $D_{\text{test}} = \{x_i\}_{i=1}^M$ 
2: Output:  $D_{\text{train\_in}}$ ,  $D_{\text{train\_out}}$ ,  $D_{\text{test\_in}}$ ,  $D_{\text{test\_out}}$ 
3: Bilancia tramite bin (soglia_minima = -8)  $D_{\text{class\_train}} \rightarrow D_{\text{train\_fase\_1}}$ 
4: Calcola le statistiche di  $D_{\text{train\_fase\_1}}$  con min-max scaler
5: Standardizza  $D_{\text{class\_train}}$  e  $D_{\text{test}}$  con le statistiche prima calcolate
6: Addestra SVM (kernel'='rbf', gamma=0.05, C=20) su  $D_{\text{train\_fase\_1}}$  scalato
7: // Localizza gli eventi del train rispetto al margine
8: for ogni evento  $x_i$  in  $D_{\text{class\_train}}$  do
9:     Calcola distanza  $d_i$  dal margine dell'iperpiano di separazione
10:    if  $d_i \leq 1$  then
11:        Assegna  $x_i$  al sottoinsieme  $D_{\text{train\_in}}$ 
12:    else
13:        Assegna  $x_i$  al sottoinsieme  $D_{\text{train\_out}}$ 
14:    end if
15: end for
16: // Localizza gli eventi del test rispetto al margine
17: for ogni evento  $x_i$  in  $D_{\text{test}}$  do
18:     Calcola distanza  $d_i$  dal margine dell'iperpiano di separazione
19:     if  $d_i \leq 1$  then
20:         Assegna  $x_i$  al sottoinsieme  $D_{\text{test\_in}}$ 
21:     else
22:         Assegna  $x_i$  al sottoinsieme  $D_{\text{test\_out}}$ 
23:     end if
24: end for
25: return  $D_{\text{train\_in}}$ ,  $D_{\text{train\_out}}$ ,  $D_{\text{test\_in}}$ ,  $D_{\text{test\_out}}$ 

```

Algorithm 2 Fase 2

```

1: Input:  $D_{\text{train\_in}}$ ,  $D_{\text{train\_out}}$ ,  $D_{\text{test}}$ 
2: Output:  $D_{\text{test\_non\_sicuri}}$ ,  $\hat{y}_i \forall x_i \in D_{\text{test}}$ 
3: // Allenamento e previsioni dei 56 modelli SVM
4: for ogni configurazione  $c_j \in \text{Tabella 6.2}$  do
5:   // Flusso outside
6:   Addestramento  $\text{SVM}_{\text{out}}^{(c_j)}$  su  $D_{\text{train\_out}}$  e class_weight='balanced'
7:   Estrai i support vector  $SV_{\text{out}}^{(c_j)}$ 
8:   Calcola le statistiche di normalizzazione Min-Max su  $SV_{\text{out}}^{(c_j)}$ 
9:   Riaddestra  $\text{SVM}_{\text{out}}^{(c_j)}$  esclusivamente su  $SV_{\text{out}}^{(c_j)}$  normalizzato
10:  Normalizza  $D_{\text{test}}$  usando le statistiche di  $SV_{\text{out}}^{(c_j)}$ 
11:  Esegui le previsioni su tutto  $D_{\text{test}}$  con  $\text{SVM}_{\text{out}}^{(c_j)} \rightarrow \text{pred}_{\text{out}}^{(c_j)}$ 
12:  // Flusso inside
13:  Addestramento  $\text{SVM}_{\text{in}}^{(c_j)}$  su  $D_{\text{train\_inside}}$  e class_weight='balanced'
14:  Estrai i support vector  $SV_{\text{in}}^{(c_j)}$ 
15:  Calcola le statistiche di normalizzazione Min-Max su  $SV_{\text{in}}^{(c_j)}$ 
16:  Riaddestra  $\text{SVM}_{\text{in}}^{(c_j)}$  esclusivamente su  $SV_{\text{in}}^{(c_j)}$  normalizzato
17:  Normalizza  $D_{\text{test}}$  usando le statistiche di  $SV_{\text{in}}^{(c_j)}$ 
18:  Esegui le previsioni su tutto  $D_{\text{test}}$  con  $\text{SVM}_{\text{in}}^{(c_j)} \rightarrow \text{pred}_{\text{in}}^{(c_j)}$ 
19:  // Predizione combinata inside-outside sui dati di test
20:  for ogni evento  $x_i$  in  $D_{\text{test}}$  do
21:    if  $\text{pred}_{\text{in}}^{(c_j)}(x_i) = \text{high}$  and  $\text{pred}_{\text{out}}^{(c_j)}(x_i) = \text{high}$  then
22:       $\text{pred}_{\text{comb}}^{(c_j)}(x_i) = \text{high}$ 
23:    else
24:       $\text{pred}_{\text{comb}}^{(c_j)}(x_i) = \text{low}$ 
25:    end if
26:  end for
27: end for
28: // Identificazione eventi previsti sicuri
29: for ogni evento  $x_i$  in  $D_{\text{test}}$  do
30:    $\#\{j \mid \text{pred}_{\text{comb}}^{(c_j)}(x_i) = \text{high}\} \rightarrow \text{voti\_high}$ 
31:    $\#\{j \mid \text{pred}_{\text{comb}}^{(c_j)}(x_i) = \text{low}\} \rightarrow \text{voti\_low}$ 
32:   if  $\frac{\text{voti\_high}}{28} \geq 0.70$  then
33:     Assegna  $\hat{y}_i = \text{high}$ 
34:   else if  $\frac{\text{voti\_low}}{28} \geq 0.80$  then
35:     Assegna  $\hat{y}_i = \text{low}$ 
36:   else
37:     Aggiungi  $x_i$  a  $D_{\text{test\_non\_sicuri}}$ 
38:   end if
39: end for
40: Inizializza  $\hat{y}_i = \text{high} \forall x_i \in D_{\text{test\_non\_sicuri}}$ 
41: return  $D_{\text{test\_non\_sicuri}}$ ,  $\hat{y}_i \forall x_i \in D_{\text{test}}$ 

```

Algorithm 3 Fase 3

```

1: Input:  $D_{\text{train\_in}}, D_{\text{train\_out}}, D_{\text{test\_in}}, D_{\text{test\_out}}, D_{\text{test\_non\_sicuri}}, \hat{y}_i \forall x_i \in D_{\text{test}}$ 
2: Output:  $D_{\text{class\_train}}, \hat{y}_i \forall x_i \in D_{\text{test}}$ 
3: // Inizializzazione dei tre classificatori
4: ETC: n_estimators=500, criterion='entropy'
5: GBC: n_estimators=200, learning_rate=0.05, max_depth=3
6: GNB: default
7: Soglia voto high risk: 0.3 per inside, 0.4 per outside
8: for  $\zeta \in \{\text{in}, \text{out}\}$  do
9:    $D_{\text{test\_}\zeta\text{\_non\_sicuri}} = D_{\text{test\_}\zeta} \cap D_{\text{test\_non\_sicuri}}$ 
10:  // Preparazione dataset di allenamento votatori  $\zeta$ 
11:  Calcola le statistiche di  $D_{\text{train\_}\zeta}$  con min-max scaler
12:  if  $\zeta = \text{in}$  then
13:    Bilancia con bin (soglia_minima = -8)  $D_{\text{train\_}\zeta} \rightarrow D_{\text{train\_fase\_3\_}\zeta}$ 
14:  else
15:    Bilancia con bin (soglia_minima = -29.99)  $D_{\text{train\_}\zeta} \rightarrow D_{\text{train\_fase\_3\_}\zeta}$ 
16:  end if
17:  Normalizza  $D_{\text{train\_fase\_3\_}\zeta}$  con le statistiche di  $D_{\text{train\_}\zeta}$ 
18:  Normalizza  $D_{\text{test\_}\zeta\text{\_non\_sicuri}}$  con le statistiche di  $D_{\text{train\_}\zeta}$ 
19:  // Addestramento e previsioni dei tre classificatori votanti
20:  Addestra classificatore GBC $^\zeta$  su  $D_{\text{train\_fase\_3\_}\zeta}$ 
21:  Addestra classificatore ETC $^\zeta$  su  $D_{\text{train\_fase\_3\_}\zeta}$ 
22:  Addestra classificatore GNB $^\zeta$  su  $D_{\text{train\_fase\_3\_}\zeta}$ 
23:  Previsioni su  $D_{\text{train\_fase\_3\_}\zeta}$  con GBC $^\zeta \rightarrow \text{Pred\_train}_{\text{GBC}}^\zeta$ 
24:  Previsioni su  $D_{\text{train\_fase\_3\_}\zeta}$  con ETC $^\zeta \rightarrow \text{Pred\_train}_{\text{ETC}}^\zeta$ 
25:  Previsioni su  $D_{\text{train\_fase\_3\_}\zeta}$  con GNB $^\zeta \rightarrow \text{Pred\_train}_{\text{GNB}}^\zeta$ 
26:  Previsioni su  $D_{\text{test\_}\zeta\text{\_non\_sicuri}}$  con GBC $^\zeta \rightarrow \text{Pred\_GBC}^\zeta$ 
27:  Previsioni su  $D_{\text{test\_}\zeta\text{\_non\_sicuri}}$  con ETC $^\zeta \rightarrow \text{Pred\_ETC}^\zeta$ 
28:  Previsioni su  $D_{\text{test\_}\zeta\text{\_non\_sicuri}}$  con GNB $^\zeta \rightarrow \text{Pred\_GNB}^\zeta$ 
29:  // Voto di maggioranza su  $D_{\text{train\_fase\_3\_}\zeta}$ 
30:  for ogni evento  $x_i$  in  $D_{\text{train\_fase\_3\_}\zeta}$  do
31:     $\#\{j \in \{\text{GBC, ETC, GNB}\} \mid \text{Pred\_train}_j^\zeta(x_i) = \text{high}\} \rightarrow \text{voti\_high}$ 
32:    if  $\text{voti\_high} < 2$  then
33:      rimuovi  $x_i$  da  $D_{\text{class\_train}}$ 
34:    end if
35:  end for
36:  // Voto di maggioranza su  $D_{\text{test\_}\zeta\text{\_non\_sicuri}}$ 
37:  for ogni evento  $x_i$  in  $D_{\text{test\_}\zeta\text{\_non\_sicuri}}$  do
38:     $\#\{j \in \{\text{GBC, ETC, GNB}\} \mid \text{Pred}_j^\zeta(x_i) = \text{high}\} \rightarrow \text{voti\_high}$ 
39:    if  $\text{voti\_high} \geq 2$  then
40:      Assegna  $\hat{y}_i = \text{high}$ 
41:    else
42:      Assegna  $\hat{y}_i = \text{low}$ 
43:    end if
44:  end for
45: end for
46: return  $D_{\text{class\_train}}, \hat{y}_i \forall x_i \in D_{\text{test}}$ 

```

Algorithm 4 Fase 4

```

1: Input:  $D_{\text{class\_train}}$ ,  $D_{\text{test\_non\_sicuri}}$ ,  $\hat{y}_i \forall x_i \in D_{\text{test}}$ 
2: Output:  $\hat{y}_i \forall x_i \in D_{\text{test}}$ 
3: // Inizializzazione dei tre classificatori
4: ETC: n_estimators=500, criterion='entropy'
5: GBC: n_estimators=200, learning_rate=0.05, max_depth=3
6: GNB: default
7: Soglia voto high risk: 0.5
8: // Preparazione dataset di allenamento votatori finali
9:  $D_{\text{test\_finali}} = \{x_i \in D_{\text{test\_non\_sicuri}} \mid \hat{y}_i = \text{high}\}$ 
10: Calcola le statistiche di  $D_{\text{class\_train}}$  con min-max scaler
11: Bilancia tramite bin (soglia_minima = -29.99)  $D_{\text{class\_train}} \rightarrow D_{\text{train\_fase\_4}}$ 
12: Normalizza  $D_{\text{train\_fase\_4}}$  con le statistiche di  $D_{\text{class\_train}}$ 
13: Normalizza  $D_{\text{test\_finali}}$  con le statistiche di  $D_{\text{class\_train}}$ 
14: // Addestramento e previsioni dei tre classificatori votanti
15: Addestra classificatore GBC su  $D_{\text{train\_fase\_4}}$ 
16: Addestra classificatore ETC su  $D_{\text{train\_fase\_4}}$ 
17: Addestra classificatore GNB su  $D_{\text{train\_fase\_4}}$ 
18: Previsioni su  $D_{\text{test\_finali}}$  con GBC  $\rightarrow Pred_{\text{GBC}}$ 
19: Previsioni su  $D_{\text{test\_finali}}$  con ETC  $\rightarrow Pred_{\text{ETC}}$ 
20: Previsioni su  $D_{\text{test\_finali}}$  con GNB  $\rightarrow Pred_{\text{GNB}}$ 
21: // Voto di maggioranza su  $D_{\text{test\_finali}}$ 
22: for ogni evento  $x_i$  in  $D_{\text{test\_finali}}$  do
23:    $\#\{j \in \{\text{GBC, ETC, GNB}\} \mid Pred_j(x_i) = \text{high}\} \rightarrow voti_{\text{high}}$ 
24:   if  $voti_{\text{high}} \geq 2$  then
25:     Conferma  $\hat{y}_i = \text{high}$ 
26:   else
27:     Assegna  $\hat{y}_i = \text{low}$ 
28:   end if
29: end for
30: return  $\hat{y}_i \forall x_i \in D_{\text{test}}$ 

```

Algorithm 5 Ensemble di Classificazione completo: richiamo delle quattro fasi

```

1: Input:  $D_{\text{class\_train}} = \{(x_i, y_i)\}_{i=1}^N$ ,  $y_i \in \{\text{low, high}\}$ ,  $D_{\text{test}} = \{x_i\}_{i=1}^M$ 
2: Output:  $\hat{y} = \hat{y}_i \forall x_i \in D_{\text{test}}$ 
3:  $D_{\text{train\_in}}, D_{\text{train\_out}}, D_{\text{test\_in}}, D_{\text{test\_out}} = \text{Fase\_1}(D_{\text{class\_train}}, D_{\text{test}})$ 
4:  $D_{\text{test\_non\_sicuri}}, \hat{y} = \text{Fase\_2}(D_{\text{train\_in}}, D_{\text{train\_out}}, D_{\text{test}})$ 
5:  $D_{\text{class\_train}}, \hat{y} = \text{Fase\_3}(D_{\text{train\_in}}, D_{\text{train\_out}}, D_{\text{test\_in}}, D_{\text{test\_out}}, D_{\text{test\_non\_sicuri}}, \hat{y})$ 
6:  $\hat{y} = \text{Fase\_4}(D_{\text{class\_train}}, D_{\text{test\_non\_sicuri}}, \hat{y})$ 
7: return  $\hat{y} = \hat{y}_i \forall x_i \in D_{\text{test}}$ 

```

6.3 Tuning e validazione dell’Ensemble di Classificazione

Complessivamente, sono stati identificati 19 iperparametri potenzialmente modificabili, distribuiti tra le varie fasi della pipeline. Tuttavia, una ricerca esaustiva su tutte le possibili combinazioni risulterebbe eccessivamente dispendiosa in termini computazionali e difficilmente gestibile.

Per questo motivo, si è scelto di fissare a priori alcuni degli iperparametri sulla base di alcune considerazioni:

- Per tutti i sistemi di voto delle fasi 3 e 4 si è deciso di utilizzare una regola di maggioranza (2 su 3), pertanto il numero minimo di voti positivi richiesti per classificare un evento come ad alto rischio è stato mantenuto fisso.
- La soglia di probabilità per classificare un evento come ad alto rischio è stata fissata a 0.5 in tutte le fasi, fatta eccezione per la fase 3 (classificatori specializzati *inside/outside*), per la quale sono stati esplorati valori più piccoli. Questo perché i classificatori di questa fase, addestrati su sottoinsiemi molto ristretti del training set, risultano altamente specializzati e particolarmente efficaci nel riconoscere eventi simili a quelli già visti all’interno della rispettiva categoria (*inside o outside*), tuttavia, proprio questa specializzazione comporta un rischio maggiore di generare falsi negativi quando si presentano eventi con comportamenti atipici rispetto a quelli osservati nel training set.
- La soglia inferiore per selezionare i sample negativi nei training set bilanciati delle fasi 3 e 4 è stata fissata a -29.99. Questo valore è stato scelto per escludere a priori gli eventi con rischio pari a -30 dall’addestramento dei classificatori presenti in queste fasi, estremamente numerosi e spesso poco informativi visto che le SVM di fase 2 nella maggior parte dei casi fileranno a monte questo tipo di eventi.

Phase	Hyperparameter	Value	Description
1	SVM_filter	0.5	Classification threshold for high risk label
3	min_sampling_lb_inside	2	Majority voting threshold (inside voters)
3	min_sampling_lb_outside	-29.99	Training sampling lower bound (outside voters)
3	min_positive_voting_outside	2	Majority voting threshold (outside voters)
4	final_min_positive_voting	2	Majority voting threshold (final voters)
4	final_min_sampling_lb	-29.99	Training sampling lower bound (final voters)
4	final_voters_filter	0.5	Probability threshold for low risk (final voters)

Tabella 6.1. Iperparametri fissati a priori e non sottoposti a tuning.

La SVM di fase 1 è stata configurata con *kernel Radial Basis Function* (RBF), selezionato al termine di una serie di test comparativi eseguiti tramite *k*-cross validation con $k = 6$. Sono stati esplorati vari kernel, tra cui lineare, polinomiale e sigmoidale, ma il kernel RBF ha mostrato le prestazioni migliori in termini di recall e stabilità delle previsioni dimostrando una maggiore capacità di modellare i confini tra le classi.

Per quanto riguarda le SVM di fase 2, queste costituiscono una componente centrale della pipeline, progettata per individuare con maggiore robustezza gli eventi

chiaramente a basso o alto rischio. In totale sono state implementate 28 SVM con configurazioni diverse:

- **8 SVM lineari**, con $C \in [0.1, 20]$;
- **8 SVM polinomiali**, con $C \in [0.5, 12]$;
- **6 SVM RBF**, con $C \in [0.25, 15]$ e $\gamma \in [0.0005, 0.02]$;
- **6 SVM sigmoid**, con $C \in [0.25, 13]$ e $\gamma \in [0.0001, 0.02]$;

Queste SVM non sono state validate singolarmente, ma valutate nel loro insieme in base alla capacità collettiva di individuare senza commettere errori eventi *sicuri*, ovvero casi classificati con consenso elevato tra le SVM. Questo approccio ha permesso di combinare in modo robusto modelli con strutture differenti, sfruttando la diversità tra kernel, parametri di regolarizzazione e funzioni di attivazione per migliorare la copertura del fronte decisionale e aumentare la probabilità che alcuni classificatori siano maggiormente efficaci su regioni particolarmente complesse dello spazio delle feature. L'efficacia di questa strategia si è manifestata fin dalle prime sperimentazioni: il sistema è riuscito a filtrare in modo stabile e affidabile una parte consistente del dataset, riducendo significativamente i falsi negativi nelle fasi successive. Questo risultato è stato ottenuto senza la necessità di un tuning spinto degli iperparametri per ciascun modello, grazie al principio su cui si basa la votazione dove l'errore di una singola SVM viene mitigato dalla presenza di altre con configurazioni complementari.

Kernel	C	γ	Kernel	C	γ
rbf	0.25	0.0005	sigmoid	0.25	0.0001
rbf	1.5	0.005	sigmoid	2	0.0005
rbf	4	0.01	sigmoid	5	0.01
rbf	6	0.02	sigmoid	7	0.02
rbf	10	0.005	sigmoid	10	0.005
rbf	15	0.001	sigmoid	13	0.001
linear	0.1	–	poly	0.5	0.1
linear	0.5	–	poly	1	0.5
linear	1	–	poly	2	2
linear	3	–	poly	4	0.01
linear	5	–	poly	6	0.2
linear	10	–	poly	8	1
linear	15	–	poly	10	0.1
linear	20	–	poly	12	0.5

Tabella 6.2. Configurazioni delle 28 SVM utilizzate nella fase 2 della pipeline.

Infine, per quanto riguarda i classificatori impiegati nelle fasi 3 e 4, è stata adottata l'entropia come criterio di suddivisione per l'Extra Trees Classifier, mentre il Gaussian Naive Bayes è stato mantenuto nelle sue impostazioni di default.

La scelta di questi tre modelli (Extra Trees, Gradient Boosting e GaussianNB) per la fase di voto finale si basa su una serie di test empirici, condotti per valutare l'efficacia di diverse combinazioni di classificatori. L'obiettivo era individuare un

insieme di modelli in grado di agire in modo complementare, minimizzando il rischio che tutti commettessero gli stessi errori sugli stessi eventi. In particolare, i tre modelli selezionati mostrano comportamenti diversi nei confronti dei casi borderline (ossia *non sicuri*), permettendo di rafforzare il processo decisionale finale attraverso una logica di voto a maggioranza.

Una volta definite le architetture da utilizzare, è stata avviata una fase esplorativa più ampia sugli iperparametri, seguendo un approccio iterativo (*trial and error*) supportato da una k -cross validation con $k = 6$, con lo scopo di selezionare un set di iperparametri in grado di garantire mediamente buone prestazioni sui sei fold di validazione. A partire da questo set di iperparametri, si è proceduto con un'analisi più dettagliata, valutando l'impatto della variazione di ciascuno di essi individualmente, mantenendo fissi tutti gli altri. Questo approccio ha permesso di isolare l'effetto specifico della variazione di ciascun iperparametro sulle prestazioni del modello, con l'obiettivo di affinare ulteriormente i valori finali selezionati e migliorarne la capacità di generalizzazione.

Nel seguito, si riportano i grafici risultanti da quest'ultima fase di validazione, focalizzandosi esclusivamente sulle curve relative agli iperparametri effettivamente adottati nella configurazione finale. Le metriche riportate (F_2 , *Recall* e *Precision*) sono state calcolate su training e validation set, al fine di evidenziare in modo chiaro il trade-off adottato. La scelta degli iperparametri è stata orientata prioritariamente alla massimizzazione del *Recall*, mantenendo comunque la *Precision* entro valori accettabili. Questo bilanciamento ha consentito di ottimizzare il valore finale di F_2 e, al tempo stesso, di limitare il numero di *False Negative* generati. Tale aspetto è particolarmente rilevante, in quanto la presenza di falsi negativi incide negativamente sul calcolo della metrica MSE_{HR} nella successiva fase di integrazione tra i due moduli del sistema (classificazione e regressione).

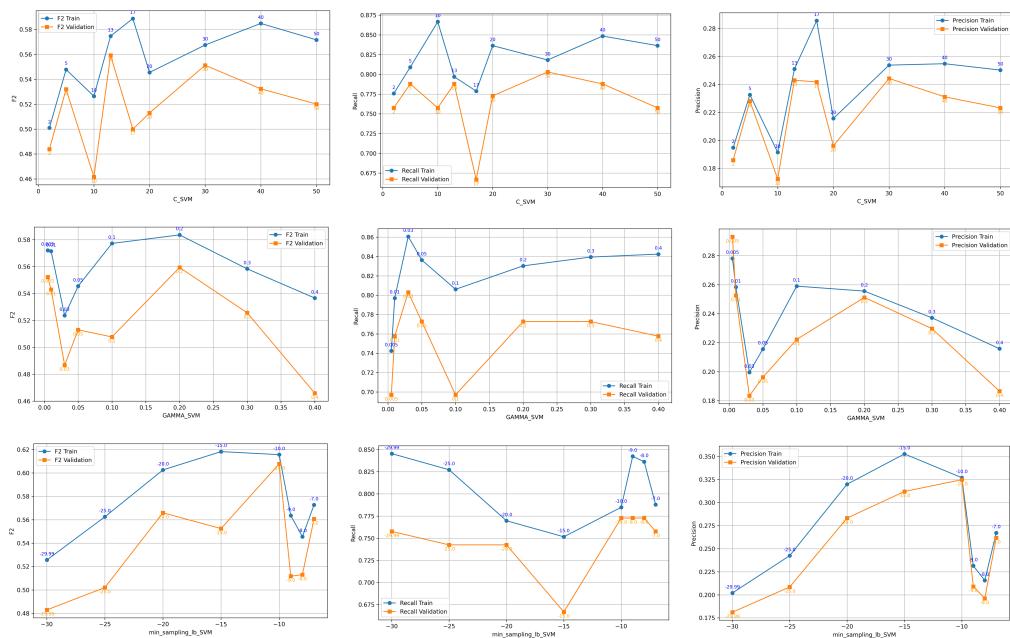


Figura 6.8. Metriche di interesse al variare degli iperparametri della FASE 1

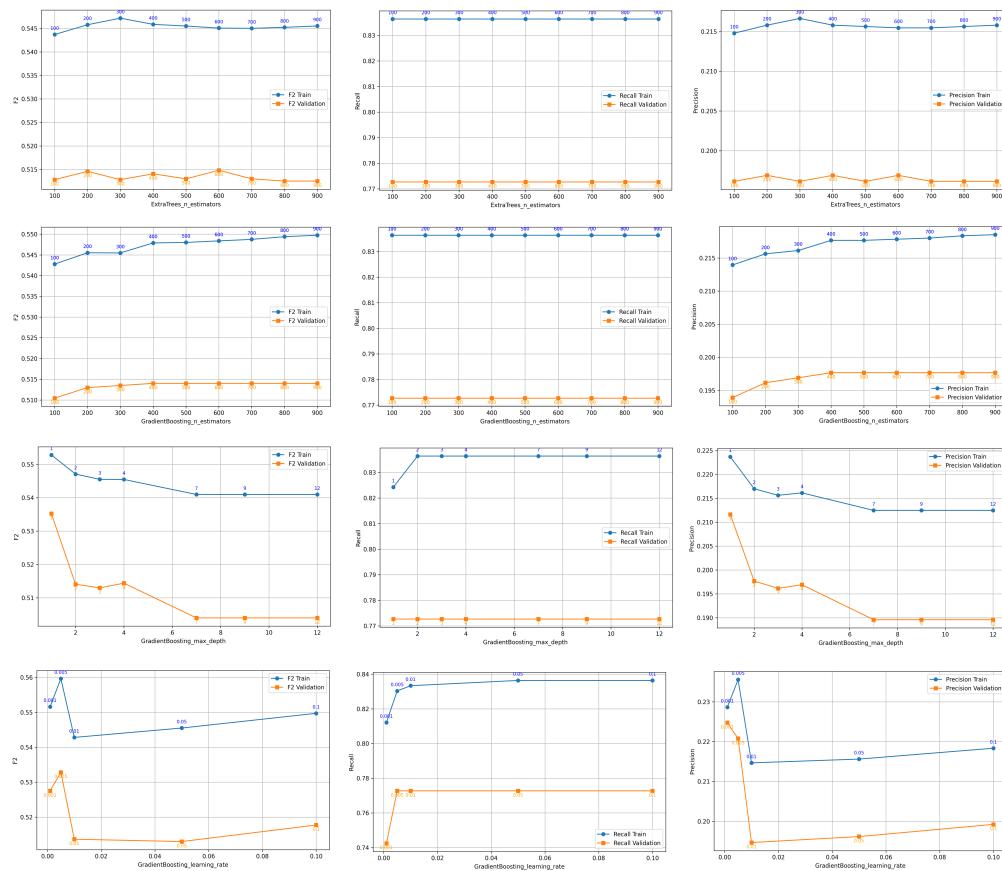


Figura 6.9. Metriche di interesse al variare degli iperparametri delle FASI 3-4

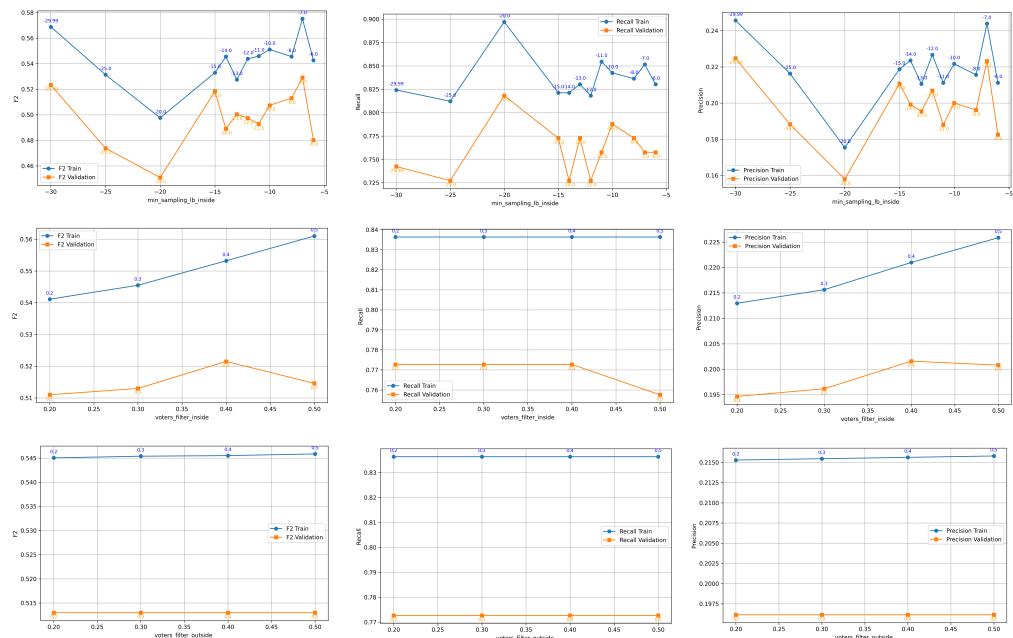


Figura 6.10. Metriche di interesse al variare degli iperparametri della FASE 3

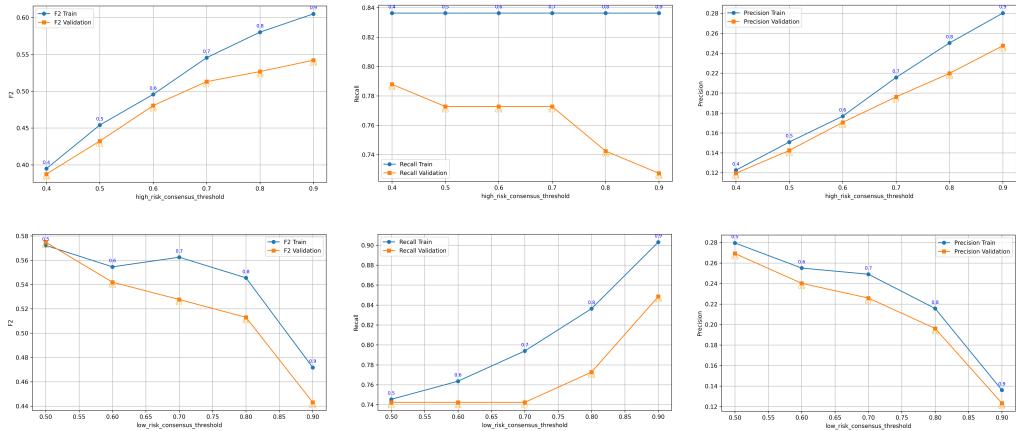


Figura 6.11. Metriche di interesse al variare degli iperparametri della FASE 2

Phase	Hyperparameter	Value	Description
1	GAMMA_SVM	0.05	Gamma value used in the initial SVM
1	C_SVM	20	Regularization parameter for the initial SVM
1	min_sampling_lb_SVM	-8	Training sampling lower bound (initial SVM)
2	high_risk_consensus_threshold	0.7	Consensus threshold for high risk prediction
2	low_risk_consensus_threshold	0.8	Consensus threshold for low risk prediction
3	min_sampling_lb_inside	-8	Min. risk for negative samples (inside voters)
3	voters_filter_inside	0.3	Probability threshold for low risk (inside)
3	voters_filter_outside	0.4	Probability threshold for low risk (outside)
3-4	ExtraTrees_n_estimators	500	Number of trees in Extra Trees classifiers
3-4	GradientBoosting_n_estimators	200	Number of boosting stages
3-4	GradientBoosting_learning_rate	0.05	Learning rate for boosting steps
3-4	GradientBoosting_max_depth	3	Max depth of each decision tree

Tabella 6.3. Iperparametri finali selezionati

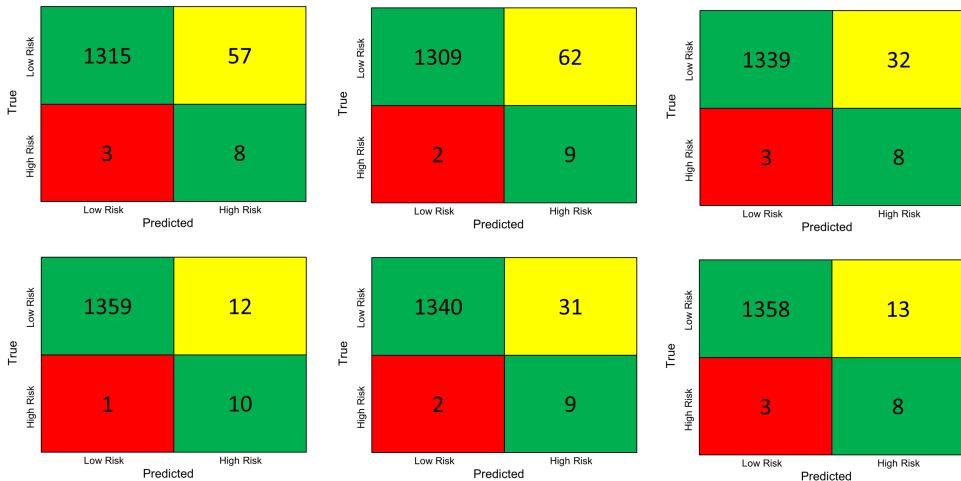


Figura 6.12. Matrici di confusione sui sei fold di validazione

In Figura 6.12 vengono riportate, a scopo illustrativo, le prestazioni ottenute dal modello finale sui diversi fold di validazione, utilizzando la configurazione di iperparametri selezionata. Questi risultati permettono di osservare la coerenza del comportamento del modello su suddivisioni differenti del dataset e di confermare visivamente la stabilità delle metriche principali.

I risultati ottenuti confermano la capacità del sistema di generalizzare correttamente anche su subset non visti del training set, mantenendo un livello basso di falsi negativi e un numero controllato di falsi positivi. In particolare, l' F_2 medio e la *Recall* media si mantengono su valori soddisfacenti e coerenti con gli obiettivi progettuali.

È importante sottolineare che l' F_2 osservato in validazione (Figura 6.13), così come quello che si otterrà calcolando l'errore sul training set, risulta fisiologicamente inferiore rispetto a quello che si osserverà sul test set ufficiale della challenge. Questo effetto è direttamente riconducibile alla diversa composizione dei due dataset.

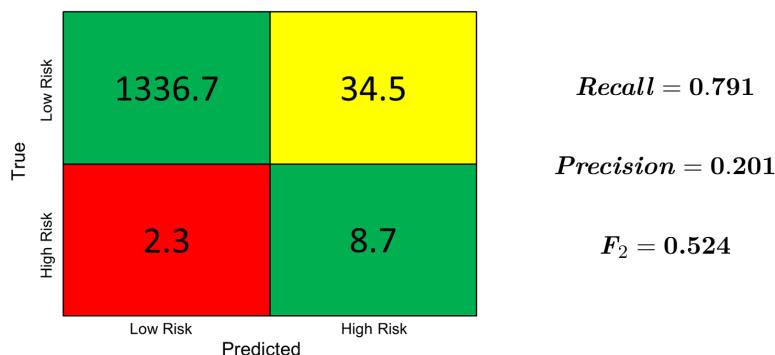


Figura 6.13. Risultati medi sui sei fold di validazione

Per chiarire questo concetto, si consideri un esempio semplificato con due dataset differenti per composizione riportati in Tabella 6.4 e si supponga di avere un modello che produca identici tassi di errore sui due dataset:

- Sbaglia il 20% degli eventi ad alto rischio \Rightarrow Recall = 0.80
- Sbaglia il 10% degli eventi a basso rischio \Rightarrow False Positive Rate = 0.1

Il calcolo delle metriche in Tabella 6.5 dimostra che, a parità di *Recall* e *False Positive Rate* (*FPR*), il valore dell' F_2 può variare in modo significativo in funzione dello sbilanciamento tra le classi.

Dataset	Total events	Low risk	High risk
A	10.100	10.000	100
B	2.150	2.000	150

Tabella 6.4. Esempio: due dataset sbilanciati simili a train e test set della challenge

Nel dataset di training della challenge, la quantità molto elevata di eventi a basso rischio determina un elevato numero di falsi positivi, che riduce drasticamente la

Precision e quindi l' F_2 . Al contrario nel dataset di test, la presenza di un numero minore di eventi a basso rischio rispetto al training set rende il contesto più favorevole e porta a un risultato finale più elevato.

Dataset	TP	FN	FP	TN	Precision	F_2
A	80	20	1 000	9 000	0.074	0.270
B	120	30	200	1 800	0.375	0.652

Tabella 6.5. Confronto delle metriche a parità di *Recall* e *FPR*

Capitolo 7

Integrazione della Regressione nella Previsione Finale

Il modulo di classificazione multilivello descritto nel Capitolo 6 consente di distinguere con alta precisione gli eventi ad alto rischio da quelli a basso rischio. Tuttavia, in assenza di un modulo di regressione complementare, la sua utilità operativa risulta limitata: tutti gli eventi classificati come ad alto rischio verrebbero infatti trattati in modo uniforme, assegnando l'etichetta alto o basso rischio o al più un valore costante, senza alcuna stima continua o graduata della reale pericolosità. Per questo motivo, è stato affiancato un modulo di regressione, con l'obiettivo comune di minimizzare lo score complessivo (Definizione 3.4) previsto dalla *Spacecraft Collision Avoidance Challenge*.

7.1 Ruolo della Regressione nel sistema predittivo combinato

Per la creazione del dataset di training del modulo di regressione è stata adottata una strategia differente rispetto a quella adottata nel modulo di classificazione: oltre ai 66 eventi che rispettano le condizioni del test set (ossia almeno una CDM con $time_to_tca \geq 2$ e una < 1), sono stati inclusi anche ulteriori 27 eventi che, pur non avendo una CDM con $time_to_tca < 1$, presentavano una CDM con $1 \leq time_to_tca < 2$. In questi casi, la CDM con $time_to_tca$ più prossima a 1 giorno è stata utilizzata come valore target per l'addestramento dei modelli.

Sebbene questa scelta possa apparire rischiosa dal punto di vista metodologico, essa ha permesso di ampliare il dataset di training per la regressione da 66 a 93 eventi, migliorando la rappresentatività dei dati e potenzialmente rafforzando la capacità di generalizzazione del modello. A supporto di questa decisione, si osserva come la baseline ufficiale LRP, che utilizza come previsione direttamente il valore di rischio dell'ultima CDM con $time_to_tca \geq 2$, ottenga comunque un errore medio quadratico sui casi ad alto rischio (MSE_{HR}) pari a 0.513.

Inoltre questa divergenza rispetto al modulo di classificazione — che invece opera rigorosamente su un dataset conforme ai requisiti della challenge — è giustificata dal ruolo subordinato assegnato al regressore nel sistema decisionale. Il suo scopo non è infatti distinguere tra eventi ad alto e basso rischio (già determinati dal

classificatore), ma fornire una stima continua del rischio per quegli eventi già valutati come pericolosi.

7.2 Integrazione dei due Moduli

Le previsioni del modulo regressivo vengono ignorate per tutti gli eventi che il classificatore ha valutato come a basso rischio. In tali casi, il rischio viene assegnato in modo costante con valore pari a -6.001 , ovvero il valore massimo previsto per la classe a basso rischio. Solo per gli eventi classificati come ad alto rischio viene richiesta al modulo regressivo una stima effettiva del rischio di collisione. Infine, in rari casi in cui il modulo di classificazione prevede un evento come ad alto rischio e il modulo di regressione fornisce una previsione inferiore alla soglia di classificazione, l'evento viene forzato al valore -6.0 (il minimo della classe ad alto rischio). Questo per garantire coerenza e preservare l'autorità della classificazione nella fase decisionale.

$$\hat{r}_i = \begin{cases} -6.001 & \text{if } \hat{r}_{i_class} = -1 \\ \hat{r}_{i_reg} & \text{if } \hat{r}_{i_class} = 1 \text{ and } \hat{r}_{i_reg} \geq -6 \\ -6 & \text{if } \hat{r}_{i_class} = 1 \text{ and } \hat{r}_{i_reg} < -6 \end{cases}$$

Il modulo regressivo utilizza una delle logiche architetturali adottate anche dal modulo di classificazione: infatti, anche in questo caso una SVM viene impiegata per suddividere gli eventi in *inside margin* e *outside margin* per poter addestrare due reti neurali MLP specializzate, ciascuna ottimizzata per stimare il rischio all'interno della propria regione di competenza¹. La figura seguente evidenzia come il modulo di regressione si integra a quello di classificazione.

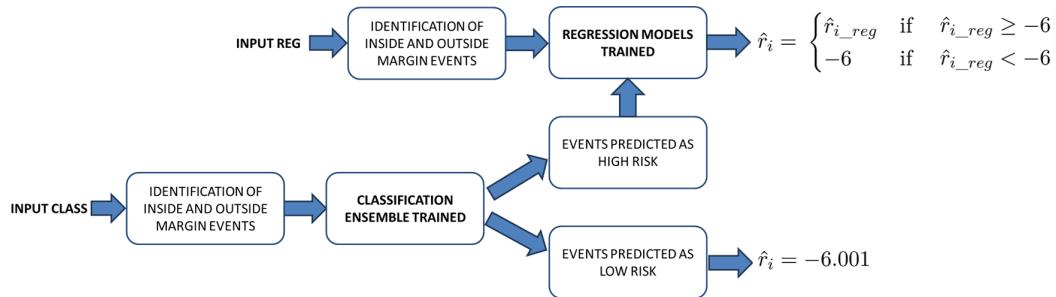


Figura 7.1. Integrazione dei due moduli sviluppati

È importante sottolineare che, sebbene sia il modulo di classificazione che quello di regressione utilizzino lo stesso meccanismo di segmentazione iniziale tra eventi *inside* e *outside*, le relative SVM usate per la segmentazione sono addestrate in modo indipendente e con iperparametri distinti in accordo anche con la diversità dei dataset adottati per l'addestramento dei due moduli.

¹Per ulteriori dettagli su implementazione, iperparametri e prestazioni di questo modulo, si rimanda alla tesi del collega Rocco Salvatore

Capitolo 8

Discussione critica

Il presente capitolo ha l'obiettivo di analizzare criticamente i risultati ottenuti dall'integrazione dei modelli sviluppati, discutendone i punti di forza, i limiti e le capacità di generalizzazione.

8.1 Valutazione del Sistema Combinato sul test set della challenge

Il modello integrato (classificazione + regressione) è stato testato sul test set ufficiale della *Spacecraft Collision Avoidance Challenge*. L'output del sistema ha prodotto la seguente matrice di confusione:

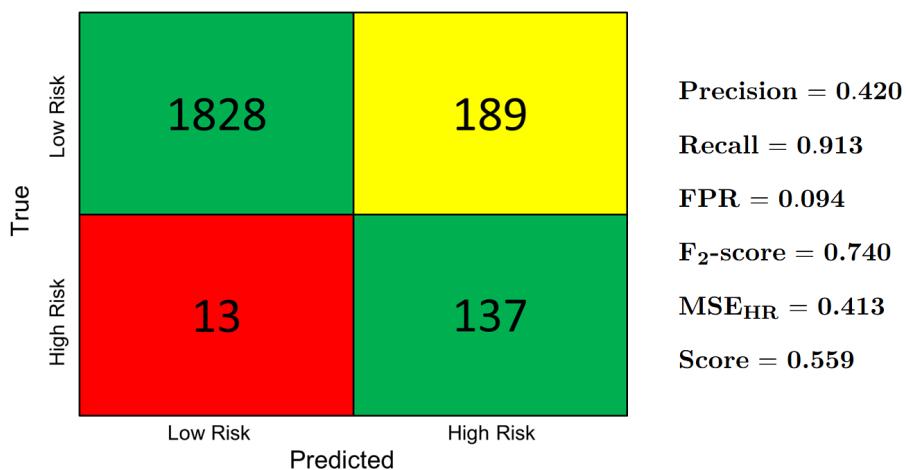


Figura 8.1. Matrice di confusione finale sul test set ufficiale e metriche di interesse

Questo risultato, se confrontato con quelli ottenuti dai team partecipanti alla challenge, si posiziona al secondo posto, superando tutti gli approcci basati su tecniche di machine learning. L'unico modello con uno score migliore è quello del team *sesc*, fondato su una cascata di soglie deterministiche ottimizzate tramite analisi statistiche. Sebbene molto efficace, questo approccio presenta una struttura rigida e, rispetto ai modelli di machine learning, una scarsa scalabilità: una volta definite, le

soglie non traggono vantaggio diretto dall'aggiunta di nuovi dati, a meno di ripetere manualmente l'intero processo di analisi. Al contrario, i modelli di machine learning possono essere riaddestrati in modo automatico e tendono a migliorare con l'aumento dei dati, cogliendo relazioni complesse e non lineari difficili da modellare con approcci deterministicici. Naturalmente, anche in questo caso è necessario rivalutare il modello aggiornato, ma il processo è molto più flessibile e adattabile a contesti dinamici.

Dopo aver analizzato le prestazioni ottenute sul test set ufficiale, è utile osservare anche il comportamento del modello sul dataset di training, al fine di valutarne la capacità di apprendimento e l'errore commesso durante la fase di addestramento. Di seguito si riporta la matrice di confusione e le metriche principali calcolate sul training set.

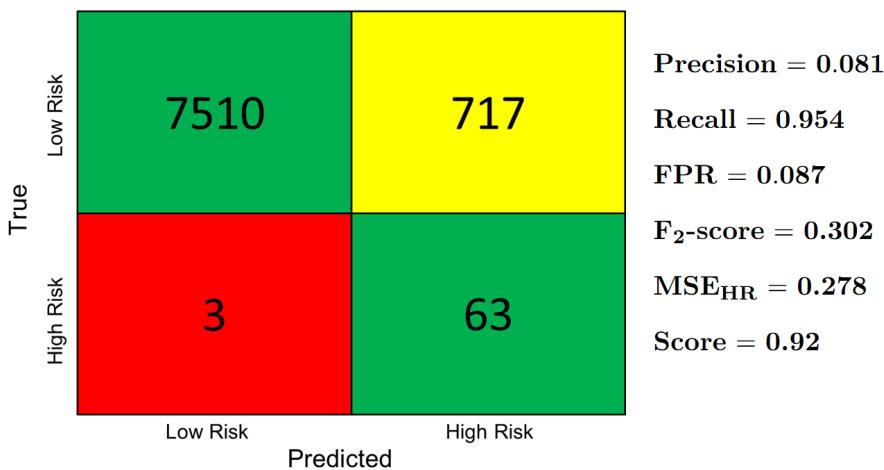


Figura 8.2. Matrice di confusione finale sul **train** set ufficiale e metriche di interesse

Le prestazioni sul training set mostrano *Recall* e *FPR* confrontabili a quelle ottenute sul test set evidenziando una discreta capacità di generalizzazione sui dati di test, le differenze nello score (0.91 sul train e 0.559 sul test) sono spiegabili unicamente dal diverso sbilanciamento dei due dataset che, come già discusso nell'esempio in Tabella 6.5, riduce drasticamente la *Precision* delle previsioni sul training set.

Per visualizzare in maniera più chiara il posizionamento del modello sviluppato rispetto alle soluzioni presentate dai team partecipanti alla Challenge, si riporta in Figura 8.3 un grafico tratto dal paper ufficiale della challenge. In esso, ciascun punto rappresenta le performance dei sistemi predittivi adottati dai partecipanti e riporta sull'asse orizzontale l'errore medio quadratico sui casi ad alto rischio e sull'asse verticale il valore di F_2 .

Nel grafico, il punto relativo all'ensemble sviluppato è stato aggiunto manualmente per evidenziare le performance ottenute sul test set ufficiale. Il modello risulta chiaramente allineato con le soluzioni migliori della challenge, e rappresenta il miglior compromesso raggiunto da un sistema completamente *data-driven* e generalizzabile. Questo confronto permette di apprezzare non solo la qualità assoluta delle prestazioni, ma anche la solidità e l'affidabilità di un approccio ibrido capace di integrare classificazione e regressione in modo sinergico.

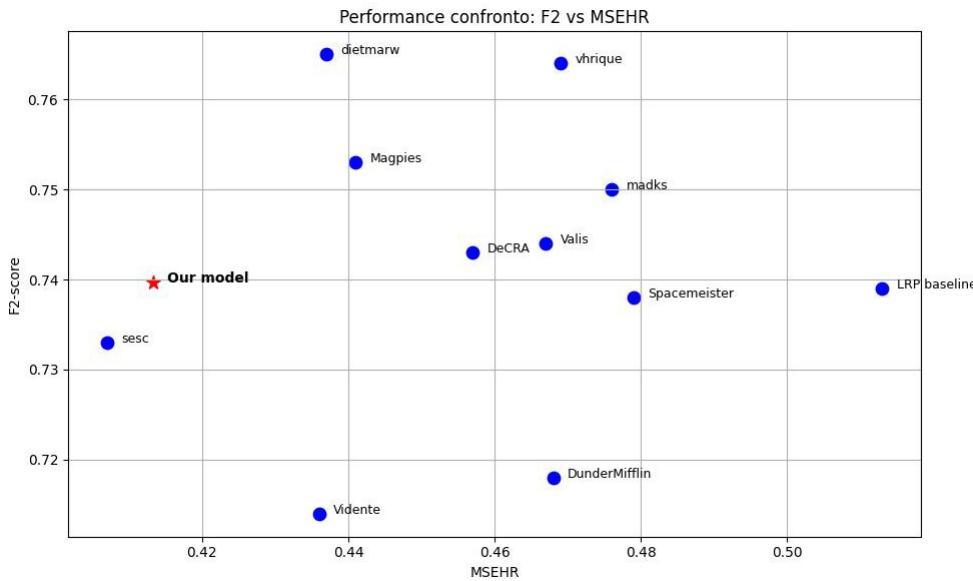


Figura 8.3. Confronto tra il modello sviluppato e i partecipanti ufficiali alla challenge

8.2 Punti di forza del Sistema Combinato

Il modello sviluppato in questa tesi presenta numerosi punti di forza, che ne giustificano le prestazioni elevate e la capacità di generalizzazione sul test set della challenge. Tali punti di forza derivano principalmente dalle scelte architetturali effettuate e dalla logica modulare su cui si basa l'intera pipeline.

Architettura modulare e scalabilità

Uno degli aspetti più rilevanti è la struttura multilivello della pipeline, che consente una scomposizione di un problema complesso in sottoproblemi più gestibili e specializzati. Ad esempio ogni sotto-modulo del modulo di classificazione (SVM iniziale, ensemble di SVM, votatori inside/outside, votatori finali), è addestrato e ottimizzato con uno scopo ben preciso, il che favorisce una maggiore scalabilità e flessibilità nel riuso dei sottomoduli o nell'estensione dell'architettura complessiva.

Un ulteriore punto di forza strategico è rappresentato dalla complementarità tra classificazione e regressione, che agiscono in modo indipendente ma sinergico. Questo tipo di struttura ha consentito di attestare il modulo di regressione su un dataset più ampio rispetto a quello utilizzato per il classificatore, includendo anche eventi con $1 \leq \text{time_to_tca} < 2$ come target. Il potenziale rischio di introdurre un leggero bias è stato gestito efficacemente grazie alla struttura gerarchica del sistema, in cui il modulo regressivo ha un ruolo subordinato alla classificazione.

Identificazione degli eventi sicuri

L'architettura proposta consente di effettuare una distinzione concettuale tra eventi previsti *sicuri* e *non sicuri*. Ad esempio sul test set della challenge, grazie al meccanismo di voto delle 28 SVM *inside/outside*, quasi il 90% degli eventi viene

etichettato come *sicuro* (1664 previsti *sicuri a basso rischio* e 273 previsti *sicuri ad alto rischio*) e classificato direttamente con elevata confidenza al termine di questa fase. Questa suddivisione introduce, di fatto, una classificazione a due livelli, in cui la percentuale di consenso associata alla classificazione può potenzialmente essere sfruttata in ambito operativo per decidere con maggiore consapevolezza in presenza di casi borderline. Una strategia conservativa potrebbe prevedere di classificare tutti gli eventi *non sicuri* come ad alto rischio, sebbene al prezzo di un maggior numero di manovre effettuate inutilmente. In particolare con questa strategia si otterebbe la seguente matrice di confusione sul test set.

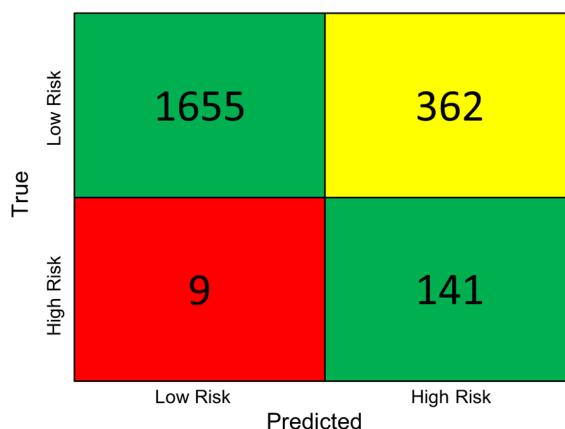


Figura 8.4. Matrice di confusione dopo la seconda fase del modulo di classificazione

Tuttavia, è bene ricordare che il sistema sviluppato non è stato pensato per sostituire l'uomo, ma per agire come strumento di supporto alle decisioni in contesti reali. Gli eventi classificati come *non sicuri* nella Fase 2 e successivamente valutati come a basso rischio dalle fasi 3 e 4 possono comunque essere sottoposti a un riesame manuale da parte di un operatore. Oltre all'analisi tecnica della serie storica completa delle CDM — che richiede competenze specialistiche non automatizzabili — l'operatore potrebbe anche tenere in considerazione la percentuale di consenso ottenuta tra le 28 SVM in fase di voto, utilizzandola come ulteriore indicatore di incertezza utile a orientare la decisione finale.

In questo senso, la distinzione tra eventi *sicuri* e *non sicuri* rappresenta un'opportunità per introdurre un livello aggiuntivo di analisi nei casi più critici, garantendo un equilibrio ottimale tra automazione e supervisione tecnica esperta.

8.3 Limiti e potenzialità di generalizzazione

Sebbene il modello ensemble sviluppato abbia raggiunto ottimi risultati, emergono alcuni limiti strutturali che vanno presi in considerazione. Per visualizzare meglio la natura e la distribuzione degli errori commessi dal sistema, si riportano i seguenti grafici, che mostrano la relazione tra la predizione finale della pipeline (asse verticale) e il valore reale del rischio (asse orizzontale), limitatamente agli eventi del test set per i quali la classificazione eseguita dal modello risulta errata:

- I falsi negativi (*False Negatives*) — eventi ad alto rischio non riconosciuti dal modello e quindi classificati come a basso rischio — sono rappresentati in rosso.

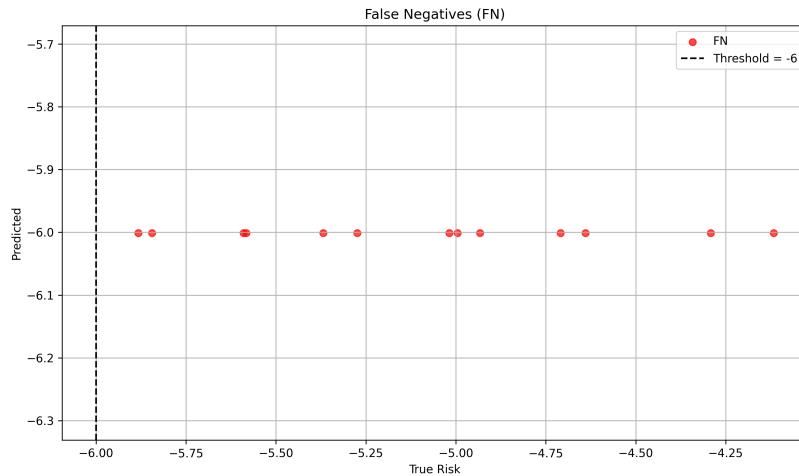


Figura 8.5. Distribuzione dei *False Negative* sul piano

- I falsi positivi (*False Positives*) — eventi a basso rischio previsti erroneamente come ad alto rischio — sono evidenziati in giallo.

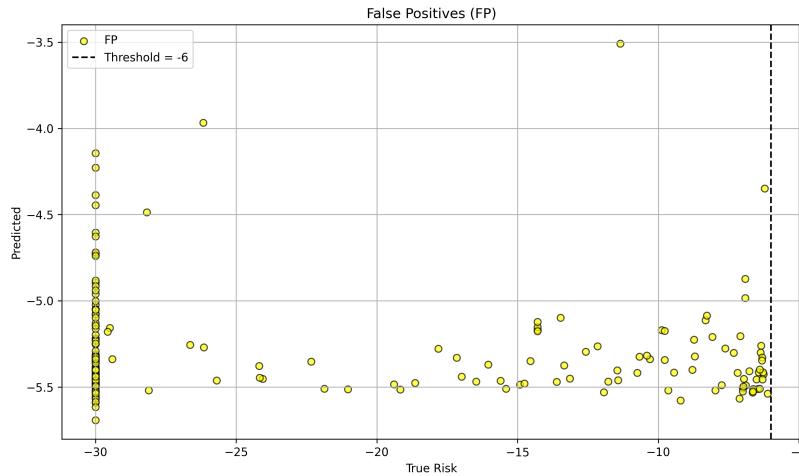


Figura 8.6. Distribuzione dei *False Positive* sul piano

Questa rappresentazione offre una visione immediata dei limiti residui del modello, mettendo in evidenza non solo la posizione dei punti erronni, ma anche l'entità dello scostamento rispetto alla soglia critica di -6. L'analisi visiva dei falsi negativi, in particolare, risulta essenziale per comprendere meglio le situazioni in cui il sistema fallisce, e per identificare potenziali margini di miglioramento nell'addestramento dei classificatori o nella struttura della pipeline.

Linee guida per un futuro miglioramento

L’analisi degli errori residui del modello, in particolare dei falsi negativi e dei falsi positivi, evidenzia alcune debolezze strutturali riconducibili in parte alla natura del dataset di addestramento. Come discusso nel Capitolo 2, il training set presenta due problematiche principali:

- Squilibrio estremo tra classi: solo 66 eventi su oltre 8.000 sono ad alto rischio, rendendo complessa la fase di apprendimento e aumentando la probabilità di falsi negativi.
- Distribuzione distorta del rischio: la maggior parte degli eventi a basso rischio assume un valore costante pari a -30 , offrendo una varietà molto limitata degli eventi a basso rischio effettivamente utili per l’addestramento.

Questo contesto riduce la capacità del modello di apprendere pattern generali e lo costringe a basarsi su un numero basso di esempi informativi. Tali problematiche si aggiungono anche all’asimmetria del training set rispetto al test set ufficiale, che, pur essendo più compatto (circa 2.000 eventi), presenta una proporzione significativamente più alta di eventi ad alto rischio (150 casi) e include alcuni eventi con forti salti di rischio tra l’ultima CDM disponibile e quella target (già mostrati in Figura 2.6). Questa asimmetria strutturale rende poco efficace ogni strategia basata unicamente sul training set, soprattutto nei confronti dei casi più critici, caratterizzati da evoluzioni anomale nel tempo.

Alla luce di queste considerazioni, emergono alcune linee guida fondamentali per potenziare la generalizzazione e la validità futura del modello:

- Ricostruzione dei dataset: unire i dataset di training e test, e ricostruire da essi due nuovi set bilanciati con proporzioni coerenti tra le classi, su cui rieseguire tuning, validazione e test finale.
- Incremento del dataset ad alto rischio: esplorare banche dati alternative o storiche per individuare eventi ad alto rischio reali non mitigati da manovre da poter includere nel dataset di allenamento.
- Diversificazione dei casi negativi: includere eventi a basso rischio con rischio diverso da -30 per arricchire l’apprendimento sui negativi.

Ruolo dell’operatore umano

Per concludere, è importante sottolineare che, allo stato attuale, nessun modello può sostituire il giudizio tecnico di un esperto del dominio. La pipeline sviluppata rappresenta un valido supporto decisionale, capace di filtrare automaticamente fino all’80 – 85% dei casi mantenendo un numero contenuto di falsi negativi, ma la decisione finale su un’eventuale manovra resta comunque responsabilità dell’operatore. Tale scelta deve tenere conto di informazioni contestuali, valutazioni dinamiche e considerazioni strategiche che non sono direttamente modellabili da un sistema automatico.

Capitolo 9

Conclusioni

Il lavoro svolto in questa tesi ha contribuito allo sviluppo di un sistema predittivo avanzato per la valutazione del rischio di collisione in orbita, integrando tecniche di classificazione e regressione in una pipeline multilivello progettata per massimizzare il recall e ridurre al minimo i falsi negativi. I risultati ottenuti nel contesto della *Spacecraft Collision Avoidance Challenge* hanno evidenziato prestazioni competitive rispetto agli approcci noti, con un'elevata capacità di generalizzazione e una struttura sufficientemente flessibile da poter essere adattata e ampliata in scenari operativi reali.

9.1 Possibili implicazioni operative per il Comando delle Operazioni Spaziali

La pipeline sviluppata non è pensata come un semplice esercizio accademico, ma come uno strumento potenzialmente applicabile in ambito operativo. In futuro, essa potrebbe essere sottoposta a una fase di sperimentazione su eventi reali.

In un simile scenario sperimentale, il sistema non verrebbe impiegato per prendere decisioni effettive, ma piuttosto come supporto “silenzioso”: gli operatori non sarebbero a conoscenza delle previsioni della pipeline, e la scelta finale in merito all’eventuale manovra resterebbe completamente a loro discrezione.

Questo approccio “cieco” consentirebbe, a posteriori, di valutare l’efficacia predittiva del modello in modo oggettivo, analizzando:

- se, nei casi in cui è stata effettivamente eseguita una manovra, la pipeline aveva correttamente classificato l’evento come ad alto rischio
- l’accuratezza della stima continua del rischio di collisione nei casi in cui quello reale ha superato la soglia critica e non è stata effettuata alcuna manovra
- il numero di falsi positivi prodotti dalla pipeline, ovvero i casi in cui il sistema prevede un rischio elevato ma il rischio finale è basso nonostante l’assenza di manovre.

Un’analisi di questo tipo rappresenterebbe un primo passo fondamentale per valutare la reale capacità di generalizzazione della pipeline su dati attuali. Se i

risultati sperimentali dovessero confermare la bontà del sistema, si potrebbe ipotizzare un suo impiego come strumento di supporto alle decisioni, da affiancare al processo decisionale dell'operatore, e da migliorare progressivamente con nuovi esempi, in particolare eventi ad alto rischio o borderline.

9.2 Configurazione hardware per l'esecuzione locale dei moduli su Python

Durante lo sviluppo e il testing dei moduli di classificazione e regressione, i codici sono stati eseguiti in locale utilizzando l'interprete Python su due notebook con configurazioni hardware simili.

Il modulo di classificazione è stato eseguito su un *ASUS VivoBook X530FN*, equipaggiato con processore *Intel Core i7-8565U* (8 core, frequenza base 2.0GHz), *16 GB di RAM* e scheda grafica integrata *Intel UHD Graphics 620*, con sistema operativo *Windows 11 Home 64-bit*.

Il modulo di regressione è stato sviluppato su un *HP ProBook 430 G5*, anch'esso dotato di processore *Intel Core i7-8550U* (8 core, 2.0GHz), *16 GB di RAM* e *Intel UHD Graphics 620*, con sistema operativo *Windows 11 Pro 64-bit*.

Entrambe le configurazioni si sono dimostrate adeguate per l'addestramento e la validazione dei modelli, nonostante l'assenza di GPU dedicate.

Ringraziamenti

Il presente lavoro è stato svolto sotto la supervisione della Professoressa Laura Palagi, che desidero ringraziare sinceramente per la disponibilità e il supporto tecnico-scientifico dimostrato durante tutte le fasi del progetto. La sua capacità di indirizzare il lavoro con competenza e rigore ha rappresentato un punto di riferimento fondamentale, soprattutto nei momenti più delicati dello sviluppo metodologico.

Un ringraziamento particolare va anche al Dottorando Lorenzo Ciarpaglini per i suoi consigli pratici, i feedback puntuali e la sua capacità di trasmettere serenità in ogni situazione, anche nei momenti più stressanti del lavoro.

Ringrazio inoltre il *Comando delle Operazioni Spaziali* per aver reso possibile questo progetto, offrendoci la possibilità di affrontare in autonomia un caso operativo reale, con piena fiducia nelle nostre capacità. Il confronto durante gli incontri tecnici e le presentazioni ha rappresentato un momento chiave per chiarire le caratteristiche del problema e coglierne appieno le implicazioni operative.

Un sentito ringraziamento va al mio collega e amico Rocco Salvatore, autore della tesi parallela, per la costante e proficua collaborazione che ha reso possibile l'integrazione efficace dei nostri moduli in un sistema predittivo completo e affidabile. Condividere questo percorso con lui ha reso il lavoro non solo più semplice, ma anche più bello: tra idee, sfide e tante risate, questa esperienza è diventata un ricordo prezioso da conservare insieme.

Bibliografia

- [1] K. T. Alfriend et al. “Probability of Collision Error Analysis”. In: *Space Debris* 1.1 (1999), pp. 21–35. DOI: 10.1023/A:1010056509803.
- [2] K. T. Alfriend et al. “Probability of Collision Error Analysis”. In: *Journal of the Astronautical Sciences* 51.2 (2003), pp. 161–178.
- [3] Nancy S. Altman. “An Introduction to Kernel and Nearest-Neighbor Non-parametric Regression”. In: *The American Statistician* 46.3 (1992), pp. 175–185.
- [4] Luciano Anselmo e Carmen Pardini. “Analysis of the Consequences in Low Earth Orbit of the Collision Between COSMOS 2251 and IRIDIUM 33”. In: *Proceedings of the 21st International Symposium on Space Flight Dynamics*. 2009. URL: <https://hdl.handle.net/20.500.14243/62295>.
- [5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [6] Vitali Braun et al. “Operational Support to Collision Avoidance Activities by ESA’s Space Debris Office”. In: *CEAS Space Journal* 8.3 (2016), pp. 177–189. DOI: 10.1007/s12567-016-0119-3.
- [7] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32.
- [8] Nitesh V. Chawla et al. “SMOTE: Synthetic Minority Over-sampling Technique”. In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357. DOI: 10.1613/jair.953.
- [9] Nello Cristianini e John Shawe-Taylor. “An Introduction to Support Vector Machines and Other Kernel-based Learning Methods”. In: *Cambridge University Press* (2000).
- [10] European Space Agency. *ESA Collision Avoidance Challenge – Final Results*. Accessed: 11 July 2025. 2021. URL: <https://kelvins.esa.int/collision-avoidance-challenge/results/>.
- [11] European Space Agency. *ESA Collision Avoidance Challenge – Home*. Accessed: 11 July 2025. 2021. URL: <https://kelvins.esa.int/collision-avoidance-challenge/home>.
- [12] European Space Operations Centre (ESOC). *DISCOS – Space Environment Statistics*. Accessed: 21 June 2025. URL: <https://sdup.esoc.esa.int/discosweb/statistics/>.

- [13] Tim Flohrer et al. “Operational Collision Avoidance at ESOC”. In: *Proceedings of the Deutscher Luft- und Raumfahrtkongress*. 2015.
- [14] Jerome H. Friedman. “Greedy Function Approximation: A Gradient Boosting Machine”. In: *Annals of Statistics* 29.5 (2001), pp. 1189–1232.
- [15] Pierre Geurts, Damien Ernst e Louis Wehenkel. “Extremely randomized trees”. In: *Machine Learning* 63.1 (2006), pp. 3–42.
- [16] Ian Goodfellow, Yoshua Bengio e Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: <https://www.deeplearningbook.org>.
- [17] David W. Hosmer e Stanley Lemeshow. “Applied Logistic Regression”. In: *Wiley* (2000).
- [18] Nicholas Johnson. “The Collision of Iridium 33 and Cosmos 2251: The Shape of Things to Come”. In: *60th International Astronautical Congress*. NASA JSC-CN-18971. 2009. URL: <https://ntrs.nasa.gov/api/citations/20100002023/downloads/20100002023.pdf>.
- [19] Heiner Klinkrad. *Space Debris: Models and Risk Analysis*. Springer-Verlag Berlin Heidelberg, 2006. ISBN: 9783540272102.
- [20] Ron Kohavi. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In: *IJCAI* 14 (1995), pp. 1137–1145.
- [21] Holger Krag. “Consideration of Space Debris Mitigation Requirements in the Operation of LEO Missions”. In: *Proceedings of the SpaceOps 2012 Conference*. 2012. DOI: 10.2514/6.2012-1257086.
- [22] J. C. Liou e N. L. Johnson. “Instability of the present LEO satellite populations”. In: *Advances in Space Research* 41.7 (2008), pp. 1046–1053.
- [23] Klaus Merz et al. “Current Collision Avoidance Service by ESA’s Space Debris Office”. In: *Proceedings of the 7th European Conference on Space Debris*. Available at: <https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/1017>. 2017.
- [24] Tomas Mikolov et al. “Recurrent neural network based language model”. In: *INTERSPEECH* (2010), pp. 1045–1048.
- [25] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. Chapter 17: Markov and hidden Markov models. MIT Press, 2012.
- [26] Jakub Nalepa. *Collision Avoidance Challenge: DunderMifflin’s code repository*. Accessed: 11 July 2025. 2021. URL: <https://gitlab.com/jnalepa/dundermifflin>.
- [27] F. Pedregosa et al. *sklearn.preprocessing.MinMaxScaler*. Scikit-learn. 2011. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>.
- [28] Sergio Pérez Morillo, Jaime Pérez Sánchez e Carlos Andrés Ramiro. *Machine Learning Lab – ESA Collision Avoidance Challenge*. Rapp. tecn. Universidad Politécnica de Madrid, 2022.

- [29] Sergio Pérez Morillo, Jaime Pérez Sánchez e Carlos Andrés Ramiro. *Predictive and Descriptive Representation Learning – ESA Collision Avoidance Challenge*. Rapp. tecn. Universidad Politécnica de Madrid, 2022.
- [30] Carolin Strobl et al. “Bias in random forest variable importance measures: Illustrations, sources and a solution”. In: *BMC Bioinformatics* 8.1 (2007), p. 25. DOI: [10.1186/1471-2105-8-25](https://doi.org/10.1186/1471-2105-8-25).
- [31] Łukasz Tulczyjew et al. “Predicting risk of satellite collisions using machine learning”. In: *Journal of Space Safety Engineering* 8.4 (2021), pp. 339–344. DOI: [10.1016/j.jsse.2021.09.001](https://doi.org/10.1016/j.jsse.2021.09.001).
- [32] Thomas Uriot et al. “Spacecraft collision avoidance challenge: Design and results of a machine learning competition”. In: *Astrodynamicics* 6.2 (2022), pp. 121–140. DOI: [10.1007/s42064-021-0101-5](https://doi.org/10.1007/s42064-021-0101-5).