

UNIVERSITÀ DEGLI STUDI DEL SANNIO
DIPARTIMENTO DI DIRITTO, ECONOMIA, MANAGEMENT E
METODI QUANTITATIVI (DEMM)

**DOTTORATO DI RICERCA “PERSONA, MERCATO, ISTITUZIONI”
XXXVII CICLO**

*Settore scientifico disciplinare STAT-04/A (già SECS-S/06)
Metodi Matematici dell'Economia e delle Scienze Attuariali e Finanziarie*

TESI DI DOTTORATO
**Applicazione del Machine Learning nella
Valutazione del Solvency Ratio**

Tutor
Prof. Nicolino Ettore D’Ortona
Co-Tutor
Prof.ssa Gabriella Marcarelli

Candidato
Mariaelisa Pelle

INDICE

INTRODUZIONE	4
1. MACHINE LEARNING E SOLVENCY II	6
1.1 IMPORTANZA DEL SOLVENCY RATIO NEL CONTESTO ASSICURATIVO	6
1.2 IL RUOLO DELLE TECNICHE DI MACHINE LEARNING NELLA VALUTAZIONE DEL RISCHIO	7
1.3 NORMATIVA DI RIFERIMENTO	9
1.4 DEFINIZIONE SOLVENCY RATIO E APPROCCI ADOTTATI PER LA DETERMINAZIONE	11
1.5 INTRODUZIONE ALLE TECNICHE DI MACHINE LEARNING NELL'AMBITO ASSICURATIVO	15
2. MODELLIZZAZIONE E MISURE DI BONTÀ.....	18
2.1 MODELLI	18
2.1.1 GENERALIZED LINEAR MODELS (GLM)	18
2.1.2 GLM CON REGOLARIZZAZIONE (SVD, RIDGE, LASSO, ELASTICNET)	20
2.1.3 BAYESIAN RIDGE REGRESSION	25
2.1.4 SUPPORT VECTOR MACHINE (SVM)	27
2.1.5 K-NEAREST NEIGHBORS (KNN).....	31
2.1.6 MODELLI AD ALBERO	34
2.1.7 RANDOM FOREST.....	34
2.1.8 XGBOOST.....	36
2.1.9 HIST GRADIENT BOOSTING (HGB)	38
2.1.10 RETI NEURALI MLP.....	39
2.1.11 GAUSSIAN PROCESS REGRESSOR (GPR)	42
2.2 METRICHE DI BONTÀ DEI MODELLI.....	44
3. DATI, METODOLOGIA E APPROCCIO UTILIZZATO	47
3.1 DATI E CRITICITÀ	47
3.2 METODOLOGIA E RISULTATI APPROCCIO INIZIALE	54
3.2.1 INTERVALLI PREDITTIVI	60
4. APPROFONDIMENTO DEL MODELLO PREDITTIVO.....	63
4.1. FEATURE SELECTION	64
4.2. CAMPIONAMENTO.....	67
4.3. APPROCCIO MACHINE LEARNING	76
4.3.1. CROSS-VALIDATION	79
4.3.2. OTTIMIZZAZIONE DEGLI IPERPARAMETRI.....	81
4.3.3. MODEL RANKING	86
4.3.4. ENSEMBLE LEARNING.....	87
4.3.5. CALIBRAZIONE	94
4.4. RISULTATI.....	96

5. CONCLUSIONI.....	105
BIBLIOGRAFIA.....	107

INTRODUZIONE

Il settore assicurativo, nel passato decennio, ha assistito a cambiamenti significativi, largamente influenzati dalla normativa Solvency II che ha standardizzato il framework di controllo e gestione del rischio introducendo il concetto di ratio di solvibilità (anche “Solvency Ratio”).

Il processo di calcolo del ratio di solvibilità emerge come un aspetto complesso per le compagnie assicurative, coinvolgendo vari attori come ad esempio la Funzione Attuariale, la Contabilità, l’Amministrazione, il Risk Manager, il CFO, il COO e la Finanza. La complessità di questo processo richiede infatti un esame accurato, delle diverse voci di Conto Economico e Stato Patrimoniale delle compagnie, quali ad esempio le riserve tecniche, gli investimenti, la liquidità, ecc.

In tale ambito si inserisce un’ulteriore profonda trasformazione dei processi aziendali, segnata dall’integrazione di tecniche di machine learning (ML) nella gestione dei dati e dei modelli valutativi, avviando una nuova fase evolutiva orientata a rimodellare i sistemi di controllo e monitoraggio del rischio in vari settori.

Il presente lavoro si propone di analizzare, nel contesto descritto, l’impiego di metodi statistici, e in particolare di tecniche di machine learning, con l’obiettivo di sviluppare uno strumento preliminare e predittivo in grado di individuare le società il cui Solvency Ratio, calcolato internamente, presenta scostamenti significativi rispetto alle previsioni generate dal modello proposto, offrendo così un meccanismo di early warning e di indirizzamento delle analisi utili ai revisori, ai risk manager e alle autorità di vigilanza. In particolare, un approccio di questo tipo potrebbe indirizzare in modo maggiormente accurato il monitoraggio delle imprese del settore da parte del sistema di controllo e gestione dei rischi.

L’analisi si basa su modelli probabilistici e statistici e prende in esame algoritmi in grado di estrarre conoscenza dai dati, costituendo dunque strumenti di validazione efficaci sia per uso interno alle compagnie sia, come appena detto, per il controllo esterno da parte degli organi di supervisione e revisione.

Alla luce di tali considerazioni, il primo obiettivo dello studio consiste nella verifica dei calcoli del ratio di solvibilità attraverso diverse tecniche tramite l’utilizzo di dati pubblici accessibili, definendo contestualmente degli intervalli predittivi. Successivamente, l’attenzione si concentra sullo sviluppo di un modello predittivo più performante, facendo ricorso anche a metodi ensemble per migliorare accuratezza e robustezza dei modelli.

L'approccio adottato rappresenta il risultato di un percorso di analisi progressivo, i cui esiti preliminari sono stati presentati in diversi contesti congressuali. In fase iniziale, le informazioni disponibili venivano aggregate tramite la costruzione di un numero limitato di indicatori, con l'obiettivo di stimare il livello di Solvency Ratio associato a ciascuna compagnia. Successivamente, come già accennato, l'attenzione si è spostata sulla definizione di modelli predittivi più avanzati, affrontando la sfida di selezionare le caratteristiche più rilevanti all'interno di una mole considerevole di dati.

Lo studio considera un database costruito includendo bilanci, conti economici, indicatori chiave di performance, ratio di solvibilità e altre informazioni rilevanti relative a 50 compagnie assicurative danni pertanto presenta un numero di variabili superiore al numero di osservazioni e dunque richiede approcci statistici e di machine learning adeguati per gestire la complessità dei dati.

Il lavoro è articolato in quattro capitoli:

- il primo capitolo è dedicato all'analisi del ruolo e della rilevanza del Solvency Ratio nel settore assicurativo, con particolare attenzione al quadro normativo di riferimento introdotto dalla disciplina Solvency II. In questa sezione verrà fornita una panoramica sul percorso che ha portato le compagnie assicurative, nel corso degli anni, a impegnarsi sempre di più nella definizione accurata dei flussi informativi necessari al calcolo del requisito patrimoniale di solvibilità. Saranno inoltre richiamati i principali meccanismi di vigilanza e controllo che hanno reso tale processo non solo un adempimento normativo, ma anche un elemento fondamentale per la gestione interna dei rischi e per la tutela degli assicurati. Un ulteriore approfondimento sarà dedicato al ruolo delle tecniche di Machine Learning nella gestione del rischio, evidenziando come strumenti avanzati di analisi e previsione possano supportare le compagnie nella costruzione di modelli più robusti e nell'ottimizzazione dei processi decisionali;
- il secondo capitolo riporta una panoramica di tutti i modelli che vengono utilizzati ai fini delle analisi, nonché le motivazioni che hanno condotto alla scelta di tali modelli. Inoltre, vengono descritte le metriche di valutazione della bontà dei modelli che verranno adottate;
- il terzo capitolo è dedicato alla descrizione del percorso metodologico seguito per il raggiungimento del primo obiettivo, precedentemente definito. In particolare, vengono illustrati i dati utilizzati per l'analisi, approfondendone le caratteristiche, le

- fonti e le principali criticità legate alla loro raccolta ed elaborazione, nonché vengono descritti i modelli adottati per le stime del Solvency Ratio e degli intervalli predittivi;
- il quarto capitolo illustra l'approccio di machine learning adottato per il conseguimento del secondo obiettivo di ricerca e presenta i principali risultati dell'analisi empirica. In particolare, viene approfondita la tecnica dell'Ensemble Learning, paradigma del machine learning fondato sulla combinazione di molteplici modelli predittivi individuali, finalizzata al miglioramento delle prestazioni rispetto a quelle ottenibili mediante un singolo modello. Il presupposto di tale approccio risiede nella capacità di modelli differenti, addestrati su sottoinsiemi dei dati o mediante algoritmi eterogenei, di cogliere aspetti complementari della complessità informativa del dataset; la loro integrazione consente pertanto di ottenere previsioni più accurate e maggiormente generalizzabili. Questa parte conclusiva mira non solo a illustrare i risultati ottenuti, ma anche a proporre possibili spunti di riflessione e sviluppi futuri, in modo da fornire un contributo utile sia in termini accademici sia per le pratiche applicative nel settore assicurativo.

1. MACHINE LEARNING E SOLVENCY II

1.1 Importanza del Solvency Ratio nel contesto assicurativo

Nel settore assicurativo, la solidità patrimoniale delle compagnie è un elemento essenziale per garantire la tutela degli assicurati e la stabilità del sistema finanziario nel suo complesso. In questo contesto, il Solvency Ratio rappresenta uno degli indicatori chiave per misurare la capacità di un'impresa di far fronte agli impegni assunti nei confronti degli assicurati.

Il Solvency Ratio è definito come il rapporto tra i fondi propri ammissibili (Eligible Own Funds, EOF) e il requisito patrimoniale di solvibilità (Solvency Capital Requirement, SCR) previsto dalla normativa europea Solvency II. Un valore del rapporto superiore al 100% indica che l'impresa dispone di risorse patrimoniali sufficienti a coprire i rischi a cui è esposta, anche in scenari avversi.

Questo indicatore sintetizza una valutazione complessa e multidimensionale del rischio, includendo non solo il rischio di sottoscrizione (associato all'attività assicurativa vera e propria), ma anche i rischi di mercato, di credito, operativi. In tal senso, il Solvency Ratio riflette la capacità dell'impresa di resistere a shock imprevisti, proteggendo gli interessi degli assicurati e contribuendo alla stabilità del sistema finanziario.

Nel contesto attuale, caratterizzato da incertezza macroeconomica, volatilità dei mercati e incremento della frequenza di eventi estremi (catastrofi naturali, cyber risk, pandemie), l'importanza di monitorare la solvibilità delle imprese assicurative è ancora più evidente. Un Solvency Ratio elevato rappresenta non solo una misura di conformità normativa, ma anche un segnale di affidabilità per assicurati, investitori, controparti e autorità di vigilanza.

In Italia, il compito di vigilare sulla solvibilità delle compagnie è affidato all'IVASS (Istituto per la Vigilanza sulle Assicurazioni), che richiede alle imprese una comunicazione costante e trasparente sulla propria situazione patrimoniale. Le compagnie devono sottoporsi periodicamente a valutazioni quantitative (QRT - Quantitative Reporting Templates) e a redigere il Regular Supervisory Report (RSR) e il Solvency and Financial Condition Report (SFCR), documenti che illustrano la propria condizione di solvibilità e la strategia di gestione del rischio.

Oltre al rispetto del requisito minimo, le compagnie fissano all'interno del Risk Appetite Framework (anche "RAF") obiettivi di Solvency Ratio superiori al 100% anche per motivi reputazionali e strategici. Un'elevata solvibilità, infatti, facilita l'accesso al mercato dei capitali, riduce il costo del capitale stesso e può costituire un vantaggio competitivo sul mercato. Tuttavia, è importante sottolineare che un Solvency Ratio troppo elevato potrebbe indicare un impiego inefficiente del capitale, a discapito della redditività. Le compagnie sono quindi chiamate a trovare un equilibrio tra solidità e rendimento, attraverso una gestione attenta del capitale e dei rischi.

Inoltre, il Solvency Ratio non deve essere interpretato in modo isolato: va considerato insieme ad altri indicatori (come il MCR – Minimum Capital Requirement, o gli stress test interni) e inserito in una più ampia cornice di governance del rischio e di ORSA (Own Risk and Solvency Assessment), che consente alla compagnia di valutare il proprio profilo di rischio in relazione alla strategia aziendale.

1.2 Il ruolo delle tecniche di Machine Learning nella valutazione del rischio

Il settore assicurativo sta vivendo una trasformazione radicale, guidata dalla digitalizzazione, dalla crescita esponenziale dei dati e dall'introduzione di tecniche avanzate come il Machine Learning (ML). In particolare, queste tecniche stanno assumendo un ruolo sempre più rilevante nella valutazione del rischio, attività centrale per ogni compagnia assicurativa.

Tradizionalmente, la valutazione del rischio si è basata su modelli attuariali classici, costruiti su ipotesi lineari e dati aggregati. Il Machine Learning, al contrario, permette di analizzare grandi moli di dati eterogenei e non strutturati, individuando pattern complessi e fornendo stime predittive più accurate e dinamiche. Questo è particolarmente utile nella previsione della probabilità di eventi assicurativi.

Le applicazioni spaziano in tutti i rami assicurativi: nei rami vita, i modelli ML possono aiutare a stimare la longevità o la probabilità di malattia; nei rami danni, possono essere utilizzati per prevedere sinistri o il rischio associato a eventi naturali. Inoltre, tecniche avanzate come le Deep Neural Networks, le random forest, o gli algoritmi di gradient boosting vengono utilizzate per costruire ad esempio modelli di pricing o nelle analisi delle frodi.

Un punto di forza del ML è la capacità di integrare dati provenienti da fonti diverse: oltre ai dati storici assicurativi, oggi vengono utilizzati dati telematici (come quelli raccolti da dispositivi IoT o scatole nere), informazioni climatiche, immagini satellitari, dati testuali e persino social media. Questo approccio consente una valutazione del rischio più contestualizzata, personalizzata e dinamica.

Ad ogni modo, l'adozione di queste tecnologie richiede una particolare attenzione e consapevolezza. A tal proposito si fa presente che le autorità di vigilanza, sia a livello europeo (come EIOPA) sia a livello nazionale (come l'IVASS), stanno monitorando da vicino l'evoluzione dell'uso del ML nel settore assicurativo (vedi la Lettera al Mercato IVASS del 6 giugno 2022 con oggetto "Indagine sull'utilizzo di algoritmi di Machine Learning da parte delle compagnie assicurative nei rapporti con gli assicurati"). L'IVASS, in particolare, ha sottolineato in più occasioni l'importanza di garantire trasparenza, correttezza e non discriminazione nei modelli di intelligenza artificiale. Iniziative e consultazioni pubbliche promosse dall'Istituto mirano proprio a promuovere un uso etico e responsabile delle tecnologie data-driven nel settore assicurativo italiano.

Inoltre, IVASS sta collaborando attivamente a livello europeo per definire linee guida comuni sull'uso di strumenti di AI e ML, ponendo particolare enfasi sul principio di explainability (spiegabilità) dei modelli, sulla governance dei dati e sulla valutazione del rischio algoritmico.

Le compagnie assicurative, da parte loro, sono chiamate a sviluppare competenze interne e strutture di controllo capaci di gestire non solo la complessità tecnica dei modelli ML, ma anche gli aspetti normativi e reputazionali connessi.

1.3 Normativa di riferimento

La direttiva Solvency II è un quadro normativo dell'Unione Europea che disciplina il settore delle assicurazioni e rappresenta uno dei più significativi sviluppi nel settore delle assicurazioni in Europa. Essa è progettata per garantire la stabilità e la solvibilità delle compagnie di assicurazione, proteggendo così i contraenti e i beneficiari. La Solvency II è entrata in vigore il 1° gennaio 2016 e rappresenta un significativo aggiornamento rispetto alle precedenti normative, con un'enfasi maggiore sul rischio e sulla gestione patrimoniale.

Prima degli anni 70, il settore delle assicurazioni in Europa era caratterizzato da un'ampia diversità di regolamentazioni nazionali, con poche armonizzazioni a livello comunitario. La crescita economica del dopoguerra e l'integrazione europea richiedevano una maggiore uniformità nelle normative assicurative per facilitare il funzionamento del mercato comune.

Nel 1973, la CEE (Comunità Economica Europea) emanò le prime direttive sull'assicurazione non vita, seguite da quelle per l'assicurazione vita nel 1979. Queste direttive furono i primi passi verso l'armonizzazione della regolamentazione assicurativa.

Introdotta negli anni 90, la Solvency I, stabilì requisiti minimi di capitale per le compagnie di assicurazione, basati principalmente su semplici misure quantitative come il volume dei premi e le riserve tecniche. Sebbene la Solvency I abbia rappresentato un progresso significativo, essa era limitata da un approccio statico e non teneva adeguatamente conto dei rischi variabili a cui le compagnie erano esposte.

Nello specifico, l'approccio risultava semplicistico, ossia la normativa utilizzava formule standardizzate che non consideravano la variabilità e la complessità dei rischi specifici di ciascuna compagnia di assicurazione. Inoltre, non veniva richiesta una valutazione dettagliata del rischio di mercato, di credito e operativo, limitando la capacità delle autorità di vigilanza di identificare e gestire potenziali vulnerabilità nel settore.

Ma una delle principali limitazioni era che la Solvency I non forniva linee guida dettagliate per la governance interna e la gestione del rischio delle compagnie di assicurazione.

Tali limitazioni, nonché la crisi finanziaria asiatica del 1997 e la crisi dei mercati emergenti nel 1998 misero in luce la necessità di un sistema più robusto per garantire la solvibilità delle compagnie di assicurazione.

Di conseguenza, nel 2000, la Commissione Europea avviò un processo di revisione della normativa Solvency I per affrontarne le carenze. Fu istituito un gruppo di esperti per esaminare le best practice a livello globale e proporre una nuova normativa.

Il processo di sviluppo di Solvency II ha visto un ampio coinvolgimento di esperti del settore, autorità di vigilanza nazionali, compagnie di assicurazione e altri stakeholder. Ciò ha garantito che la normativa fosse ben informata e ampiamente accettata.

La nuova normativa ha incorporato standard internazionali come quelli dell'International Association of Insurance Supervisors (IAIS) e del Basel Committee on Banking Supervision, per garantire una coerenza globale e promuovere la stabilità del settore assicurativo.

Dopo numerose discussioni e revisioni, la direttiva Solvency II (2009/138/CE) è stata adottata dal Parlamento Europeo e dal Consiglio dell'Unione Europea nel 2009.

Gli Stati membri dell'UE hanno recepito la direttiva nei rispettivi ordinamenti giuridici nazionali, adattando le proprie normative per conformarsi ai nuovi standard europei.

La normativa è entrata ufficialmente in vigore il 1° gennaio 2016, dopo un periodo di transizione che ha consentito alle compagnie di assicurazione di adeguarsi ai nuovi requisiti.

Solvency II si basa su tre pilastri fondamentali:

Pillar I: Requisiti Quantitativi

Capitale di Solvibilità: la direttiva introduce requisiti di capitale basati su una valutazione del rischio. Le compagnie devono mantenere un Capitale di Solvibilità Minimo (Minimum Capital Requirement, MCR) e un Capitale di Solvibilità (Solvency Capital Requirement, SCR). Il MCR è il livello minimo di capitale che un assicuratore deve detenere per evitare il fallimento immediato, mentre il SCR è più complesso e include una valutazione del rischio che la compagnia può affrontare a lungo termine.

Le riserve tecniche devono essere calcolate utilizzando i principi di mercato, riflettendo il valore attuale degli impegni assicurativi, gli attivi devono essere valutati al fair value e devono esserci adeguate garanzie di liquidità per coprire i passivi.

Pillar II: Requisiti di Supervisione e Governance

Sistema di Governance: le compagnie devono adottare un robusto sistema di governance che include funzioni chiave come gestione del rischio, controllo interno, attuariale e audit interno.

Processo di Valutazione del Rischio: ogni compagnia deve condurre una Valutazione Interna del Rischio e della Solvibilità (Own Risk and Solvency Assessment, ORSA), che valuta l'adeguatezza del capitale rispetto ai rischi aziendali.

Requisiti di Supervisione: le autorità di vigilanza hanno il compito di valutare la solidità finanziaria delle compagnie, effettuare ispezioni regolari e prendere misure correttive quando necessario.

Pillar III: Requisiti di Trasparenza e Reporting

Trasparenza: le compagnie di assicurazione devono fornire informazioni dettagliate sul loro profilo di rischio, la gestione del capitale e la situazione finanziaria.

Reporting: la normativa richiede la compilazione di rapporti regolari che devono essere presentati sia al pubblico che alle autorità di vigilanza. Questi includono il rapporto sulla solvibilità e la condizione finanziaria (Solvency and Financial Condition Report, SFCR) e il rapporto alla vigilanza (Regular Supervisory Report, RSR).

Nonostante i benefici, Solvency II ha sollevato alcune preoccupazioni relativamente alla complessità della normativa e ai costi associati all'adeguamento, soprattutto da parte delle compagnie più piccole.

1.4 Definizione Solvency Ratio e approcci adottati per la determinazione

Come già anticipato il Solvency Ratio rappresenta uno degli indicatori fondamentali di solidità finanziaria per le compagnie assicurative. Esso misura la capacità di un'impresa di far fronte ai propri impegni nei confronti degli assicurati, rapportando il livello di capitale disponibile al requisito patrimoniale richiesto dalla normativa vigente. In termini formali, il Solvency Ratio è espresso come rapporto tra i fondi propri ammissibili (Own Funds) e il Solvency Capital Requirement (SCR). Il risultato, espresso in percentuale, fornisce un'indicazione immediata della stabilità e della resilienza dell'impresa rispetto a eventi avversi che potrebbero compromettere la sua capacità di adempiere alle obbligazioni contrattuali.

Dal punto di vista gestionale, il Solvency Ratio non è soltanto un indicatore statico, ma riflette l'equilibrio dinamico tra capitale disponibile e rischi assunti. Fattori quali l'andamento dei mercati finanziari, la volatilità dei tassi di interesse, l'esposizione ai rischi assicurativi e operativi, nonché le strategie di investimento adottate, influiscono direttamente sul valore dell'indicatore.

Per questo motivo, le compagnie assicurative dedicano risorse crescenti allo sviluppo di modelli quantitativi avanzati e all'applicazione di tecniche di Risk Management, spesso integrate con metodologie di Machine Learning e strumenti di previsione in grado di stimare scenari futuri e stress test.

L'importanza del Solvency Ratio emerge chiaramente anche dal confronto tra le principali compagnie operanti sul mercato italiano. I dati, solitamente pubblicati nella relazione di solvibilità e condizione finanziaria (SFCR – Solvency and Financial Condition Report), consentono di evidenziare le differenze tra i diversi operatori in termini di solidità patrimoniale e capacità di assorbire i rischi.

Determinazione del Solvency Ratio

La determinazione del Solvency Capital Requirement può essere effettuata secondo diversi approcci metodologici, che riflettono in misura crescente la personalizzazione del modello rispetto al profilo di rischio della compagnia. I principali approcci previsti dalla normativa¹ sono:

- la Formula Standard;
- il Modello Interno Completo (Full Internal Model);
- il Modello Interno Parziale (Partial Internal Model);
- l'utilizzo dei Parametri Specifici dell'Impresa (Undertaking Specific Parameters, USP).

Ciascuno di questi approcci risponde a esigenze differenti in termini di complessità, granularità del calcolo e adeguatezza alla realtà aziendale. La scelta tra essi dipende da diversi fattori, tra cui la natura dell'attività assicurativa, il profilo di rischio, le risorse disponibili e le aspettative della vigilanza.

FORMULA STANDARD

La formula standard rappresenta l'approccio di base e più diffuso all'interno del framework Solvency II. Si tratta di un modello regolamentare predefinito, messo a punto da EIOPA, che permette alle imprese di calcolare il proprio requisito patrimoniale utilizzando parametri predefiniti, garantendo un'applicazione uniforme a tutte le compagnie europee, a prescindere dalla loro dimensione o struttura.

¹ Si vedano i Regolamenti della Commissione Europea (2014-2015)

Questo approccio si fonda su una scomposizione modulare dei rischi. I principali rischi considerati sono quelli di sottoscrizione (nei rami vita, danni e malattia), i rischi di mercato (come la variazione dei tassi di interesse, il rischio azionario o il rischio cambio), il rischio di credito (o di controparte) e il rischio operativo. Ogni modulo viene calcolato separatamente e poi aggregato tenendo conto delle correlazioni tra i diversi rischi, secondo una matrice predefinita.

La formula standard è spesso considerata un buon punto di partenza, soprattutto per le compagnie di dimensioni medie o piccole, che non dispongono delle risorse per sviluppare un modello interno complesso. Tuttavia, proprio per la sua natura standardizzata, può non riflettere pienamente il profilo di rischio effettivo di una specifica impresa. In certi casi può risultare eccessivamente conservativa, in altri sottovalutare determinate esposizioni. Nonostante ciò, la sua semplicità, la facilità di implementazione e l'assenza di obblighi di autorizzazione ne fanno un'opzione largamente adottata nel panorama europeo.

MODELLO INTERNO COMPLETO

Per le compagnie più strutturate o con un profilo di rischio particolarmente complesso, la normativa consente di sostituire la formula standard con un modello interno completo. Questo approccio permette all'impresa di sviluppare e utilizzare strumenti propri di misurazione del rischio, costruiti in modo da riflettere con maggiore fedeltà le specificità del proprio portafoglio, dei propri processi e delle proprie strategie.

Il modello interno si basa su tecniche attuariali e finanziarie avanzate e richiede l'uso di simulazioni stocastiche, analisi di scenari e modellizzazioni probabilistiche. Può comprendere tutti i rischi a cui l'impresa è esposta o integrare metodologie diverse a seconda della natura dei rischi. L'adozione di un modello interno completo comporta però un processo rigoroso di autorizzazione da parte dell'autorità di vigilanza (in Italia rappresentata dall'IVASS). L'impresa deve dimostrare che il modello sia accurato, stabile nel tempo, coerente con l'uso gestionale interno e ben integrato nei processi decisionali.

L'adozione di un modello interno consente un calcolo più aderente al rischio effettivo e può portare alla determinazione di un requisito patrimoniale più coerente, grazie a una stima più specifica e a una migliore considerazione dei benefici di diversificazione. Tuttavia, lo sviluppo e il mantenimento di questi modelli richiedono risorse notevoli, sia in termini di competenze tecniche che di sistemi informatici e governance interna. Per questo motivo, il

modello interno completo viene adottato soprattutto da grandi gruppi o da compagnie con profili di rischio molto specifici.

MODELLO INTERNO PARZIALE

Un'alternativa intermedia tra la formula standard e il modello interno completo è rappresentata dal modello interno parziale. Questo approccio consente all'impresa di sviluppare un proprio modello solo per alcune componenti del requisito di capitale, lasciando invariate le altre secondo le regole della formula standard. È una soluzione flessibile, che permette di affrontare con maggiore precisione i rischi che l'impresa ritiene non adeguatamente catturati dalla formula standard, senza dover modellizzare l'intero SCR.

Per esempio, una compagnia che esercita principalmente rami danni potrebbe decidere di utilizzare un modello interno solo per il rischio di catastrofi naturali, che può essere particolarmente significativo per il proprio portafoglio, mantenendo la formula standard per tutti gli altri moduli.

Anche in questo caso, è necessaria l'autorizzazione dell'IVASS, che valuta la coerenza del modello parziale rispetto al profilo di rischio della compagnia e la qualità metodologica della soluzione proposta. Sebbene meno oneroso del modello completo, il modello parziale richiede comunque una struttura interna solida, processi di validazione rigorosi e un'integrazione efficace con la parte di SCR calcolata adottando la formula standard. Si tratta dunque di un compromesso tra precisione e complessità, utile soprattutto per quelle imprese che hanno particolari esigenze su aree di rischio specifiche.

UNDERTAKING SPECIFIC PARAMETERS (USP)

Un'ulteriore possibilità di personalizzazione della formula standard è offerta dall'utilizzo dei parametri specifici dell'impresa, noti come USP (Undertaking Specific Parameters). In determinati moduli della formula standard, in particolare quelli relativi ai rami danni e malattia, è possibile sostituire i parametri standard forniti da EIOPA con parametri basati sull'esperienza storica dell'impresa stessa.

Questa opzione rappresenta una forma di adattamento “leggero” rispetto allo sviluppo di un modello interno e permette di migliorare l'accuratezza del calcolo in presenza di dati affidabili e rappresentativi. Per utilizzare gli USP, l'impresa deve richiedere l'autorizzazione

all'IVASS, fornendo evidenze statistiche solide, una metodologia di calcolo robusta e una governance adeguata.

Gli USP rappresentano una soluzione intermedia: da un lato, offrono una maggiore aderenza alla realtà dell'impresa rispetto ai parametri standard; dall'altro, non richiedono lo sviluppo e la manutenzione di un intero modello interno. Tuttavia, l'impresa deve garantire l'affidabilità dei dati e la corretta applicazione dei parametri nel tempo, affinché il livello di capitale richiesto rifletta effettivamente la propria esposizione al rischio.

Alla luce delle considerazioni esposte per ciascun approccio, è opportuno sottolineare che ogni metodo – dalla formula standard ai modelli interni, fino agli USP – presenta specifici punti di forza e limiti, che devono essere attentamente valutati in funzione delle caratteristiche dell'impresa, della tipologia dei rischi assunti e della capacità di investimento in competenze e sistemi informatici.

Dunque, la scelta del metodo adottato per il calcolo del Solvency Ratio non è mai puramente tecnica, poiché influenza non solo il livello di capitale richiesto, ma anche le politiche di gestione dei rischi, le strategie dell'impresa e il modo in cui questa viene valutata dal mercato e dalle autorità di vigilanza.

In questo contesto, IVASS, in coerenza con il quadro normativo europeo, svolge un ruolo centrale nel garantire che le metodologie adottate siano robuste, trasparenti e coerenti con i principi di proporzionalità e prudenza. Spetta infatti all'autorità di vigilanza assicurare che il capitale detenuto dalle imprese sia costantemente adeguato a coprire i rischi effettivamente assunti, evitando sia sottostime potenzialmente pericolose sia eccessi di capitale inefficienti.

1.5 Introduzione alle tecniche di Machine Learning nell'ambito assicurativo

I modelli di Machine Learning (ML) offrono strumenti avanzati per affrontare fenomeni assicurativi complessi e non lineari, migliorando in modo significativo la capacità predittiva rispetto ai metodi tradizionali, in particolare per eventi rari o a elevata variabilità, come i sinistri catastrofali o il comportamento dinamico degli assicurati.

Le tecniche di ML non sostituiscono i modelli attuariali classici, bensì li completano e potenziano, introducendo:

- una maggiore flessibilità modellistica, grazie alla riduzione dei vincoli parametrici e statistici;
- una superiore efficienza computazionale, che consente di gestire una vasta mole di dati e molteplici scenari di simulazione.

Diversi contributi scientifici hanno mostrato l'efficacia delle tecniche ML anche nel contesto regolamentare di Solvency II.

Krah, Nikolić e Korn (2019) applicano modelli ML — tra cui GLM, GAM, MARS e kernel regression — per costruire proxy del SCR in ambito vita, utilizzando tecniche di Least-Squares Monte Carlo (LSMC). I modelli così ottenuti riducono significativamente i tempi di calcolo, mantenendo un'elevata capacità di generalizzazione nei test fuori campione.

Hejazi e Jackson (2016) integrano reti neurali nelle simulazioni Monte Carlo annidate, ottenendo stime affidabili del SCR su portafogli di rendite variabili (Variable Annuities), con guadagni in efficienza rispetto ai metodi tradizionali. Flaig e Junike (2022) propongono l'uso di Generative Adversarial Networks (GAN) per generare scenari per il calcolo del rischio di mercato ai fini Solvency II. I risultati ottenuti sono comparabili con quelli di modelli regolamentari, offrendo un'alternativa data-driven valida e flessibile.

Tecniche di machine learning vengono adottate da Castellani, Fiore, Marino, Passalacqua, Perla, Scognamiglio e Zanetti (2018) per stimare il Solvency Capital Requirement nell'assicurazione vita tramite l'applicazione di Deep Neural Networks e Support Vector Regressors per approssimare la distribuzione del Net Asset Value. Le prestazioni dei modelli sono confrontate con la tecnica tradizionale Least Squares Monte Carlo. I risultati evidenziano l'efficacia del machine learning come alternativa efficiente e accurata nei processi di valutazione regolamentare.

Nel campo del lapse risk management, Loisel, Piette e Tsai (2019) impiegano tecniche di Support Vector Machines (SVM) e XGBoost per modellare le probabilità di estinzione anticipata delle polizze, ottenendo performance superiori rispetto ai modelli predittivi tradizionali in termini di accuratezza e valore economico.

Un modello ensemble basato su rete neurale e stacking viene adottato da Sijie, Sia, Alfred e Mounq (2024) per prevedere i premi delle assicurazioni sanitarie. Lo studio si inserisce nel contesto delle difficoltà di pricing del settore assicurativo, influenzato da diversi fattori, quali l'inflazione, vincoli normativi, concorrenza, etc. Gli autori evidenziano come i modelli di machine learning siano ampiamente utilizzati per la stima dei premi, ma come alcune

tecniche ensemble risultino ancora poco esplorate in questo ambito. L'approccio proposto pertanto integra una rete neurale artificiale con un altro algoritmo di apprendimento automatico tramite stacking. I risultati ottenuti su un dataset statunitense di assicurazioni sanitarie, mostrano che il modello ensemble supera le prestazioni dei modelli singoli nella previsione dei premi.

2. MODELLIZZAZIONE E MISURE DI BONTÀ

Questo capitolo è dedicato alla descrizione dei modelli applicati al caso studio, con particolare attenzione alle motivazioni che hanno condotto alla scelta di tali modelli rispetto alle numerose alternative presenti in letteratura e, in particolare, nell'ambito del machine learning, nonché alle metriche di valutazione della bontà dei modelli che verranno adottate.

2.1 Modelli

I modelli qui descritti sono presentati nella variante di regressione, ovvero per la stima di target quantitativi, prevedendo che le input features siano anch'esse quantitative.

I modelli GLM rappresentano uno standard *de-facto* e, come si vedrà, presentano alcuni elementi comuni agli altri modelli impiegati, in primis il concetto di stima del valore atteso. Tuttavia, modelli sempre più complessi o basati su approcci alternativi alla stima (ossia modelli che non si fondano sulle tecniche di stima tradizionali), introducono degli elementi che aiutano a gestire il problema dell'alta dimensionalità, come la regolarizzazione, il boosting e approcci più geometrici.

2.1.1 Generalized Linear Models (GLM)

I modelli lineari generalizzati (Generalized Linear Models, GLM) rappresentano un'estensione del modello di regressione lineare classico che amplia la sua applicabilità a una vasta gamma di dati con distribuzioni diverse dalla normale e relazioni non lineari tra la media della variabile risposta e le variabili predittrici. A differenza della regressione lineare standard, che assume una distribuzione normale per la variabile risposta y e una relazione lineare diretta tra la sua media μ e i predittori X , i GLM consentono di modellare dati con distribuzioni appartenenti alla famiglia esponenziale, come dati binari, conteggi o dati positivi con distribuzione asimmetrica.

Un GLM è definito da tre componenti fondamentali che ne caratterizzano la struttura e il funzionamento. Innanzitutto, si specifica la distribuzione della variabile risposta y , che può essere una distribuzione Binomiale, di Poisson, Gamma o altre distribuzioni della famiglia esponenziale, a seconda della natura dei dati. Successivamente, si definisce il predittore lineare η , che, analogamente alla regressione lineare, è una combinazione lineare delle variabili predittrici X : $\eta = X\beta$, dove β rappresenta il vettore dei coefficienti di regressione. Infine, si introduce la funzione di legame (link function) g , una funzione monotona e

differenziabile che connette la media della variabile risposta $\mu = E(y)$ al predittore lineare η : $g(\mu) = \eta$, dove $E(y)$ è il valore atteso di y . La scelta appropriata della funzione link dipende dalla distribuzione assunta per la variabile risposta.

Alcuni esempi di GLM e delle relative funzioni link includono la regressione lineare, dove la distribuzione è normale e la funzione di legame è l'identità ($g(\mu) = \mu$); la regressione logistica, con distribuzione binomiale e funzione di legame logit ($g(\mu) = \log(\mu/(1 - \mu))$); e la regressione di Poisson, con distribuzione di Poisson e funzione di legame logaritmo ($g(\mu) = \log(\mu)$).

L'inferenza nei GLM, ovvero la stima dei coefficienti β , si basa tipicamente sul metodo della massima verosimiglianza (MLE). Questo metodo cerca di trovare i valori di β che massimizzano la probabilità di osservare i valori di y dato il modello. A differenza della regressione lineare, dove la soluzione MLE ha una forma analitica chiusa, nei GLM è spesso necessario ricorrere ad algoritmi numerici.

Nella regressione lineare standard, la soluzione dei minimi quadrati ordinari (OLS), che coincide con la soluzione MLE sotto l'assunzione di normalità degli errori, è data da:

$$\beta_{OLS} = (X^T X)^{-1} X^T y$$

Nei GLM, invece, la soluzione MLE non ha generalmente una forma chiusa e richiede l'uso di algoritmi iterativi come l'algoritmo di Newton-Raphson o l'algoritmo Iteratively Reweighted Least Squares (IRLS). L'IRLS, in particolare, consiste in una sequenza di minimizzazioni di problemi di minimi quadrati pesati, dove i pesi vengono aggiornati ad ogni iterazione in base alla funzione di legame e alla derivata della funzione di verosimiglianza. Questo algoritmo converge alla soluzione MLE per i GLM. Non esiste quindi una formula comparabile alla OLS per tutti i GLM, ma si utilizzano metodi iterativi per trovare la soluzione numerica.

I GLM offrono un framework flessibile per la modellazione di dati con diverse distribuzioni e relazioni non lineari tra la media della variabile risposta e i predittori. La scelta della distribuzione e della funzione di legame è cruciale per l'adattamento del modello alle specifiche caratteristiche dei dati. Mentre la regressione lineare standard offre una soluzione analitica per la stima dei coefficienti, i GLM richiedono l'impiego di metodi iterativi per la stima di massima verosimiglianza. Inoltre, l'interpretabilità dei coefficienti, pur dipendendo dalla funzione di link utilizzata, fornisce informazioni sulla direzione e l'entità dell'effetto

dei predittori sulla risposta. L'efficienza computazionale dei GLM li rende adatti anche ad analisi con dataset di dimensioni moderate.

2.1.2 GLM con regolarizzazione (SVD, Ridge, Lasso, ElasticNet)

In contesti di alta dimensionalità, dove il numero di predittori è comparabile o addirittura superiore al numero di osservazioni, i GLM tendono a soffrire di overfitting. Questo fenomeno si manifesta con la capacità del modello di adattarsi eccessivamente ai dati di training, memorizzando il rumore presente nei dati e compromettendo la sua capacità di generalizzare a nuovi dati non visti. Per superare questa limitazione, si può ricorrere a tecniche di regolarizzazione, che introducono un vincolo sulla complessità del modello, penalizzando i coefficienti di regressione di elevata magnitudine. In letteratura esistono alcune tecniche di regolarizzazione ampiamente utilizzate in combinazione con i GLM: Singular Value Decomposition (SVD), Ridge, Lasso ed Elastic Net.

L'idea fondamentale alla base della regolarizzazione è di aggiungere un termine di penalità alla funzione di costo che il modello cerca di minimizzare durante la fase di addestramento, che solitamente si fa corrispondere alla somma dei quadrati degli scarti (RSS). Aggiungendo un termine di penalità, si introduce un trade-off tra la bontà di adattamento ai dati di training e la semplicità del modello, misurata dalla magnitudine dei suoi coefficienti. Questo trade-off è controllato da un parametro di regolarizzazione, spesso indicato con λ , che determina l'intensità della penalità.

Singular Value Decomposition (SVD)

La decomposizione ai valori singolari, nota come Singular Value Decomposition (SVD), si configura come uno strumento matematico di notevole importanza, con applicazioni che spaziano in diversi ambiti.

Considerando un insieme di dati rappresentato da una matrice X , le cui dimensioni n e p denotano rispettivamente il numero di osservazioni e di predittori, la SVD di X consiste nella sua decomposizione nel prodotto di tre matrici: $X = U\Sigma V^T$. In questa scomposizione, U e V rappresentano matrici ortogonali, mentre Σ è una matrice diagonale contenente i valori singolari di X , che ne quantificano l'importanza relativa.

Nel contesto della regressione lineare, l'obiettivo primario consiste nella determinazione del vettore dei coefficienti β , che minimizza la discrepanza tra i valori predetti e quelli osservati. La soluzione ai minimi quadrati ottenuta mediante l'applicazione della SVD è data da $\beta =$

$V\Sigma^+U^T y$, dove Σ^+ rappresenta la matrice pseudo-inversa di Σ , calcolata invertendo i valori singolari non nulli e trasponendo la matrice risultante.

È proprio nel calcolo della matrice pseudo-inversa Σ^+ che emerge la forma implicita di regolarizzazione: i valori singolari prossimi allo zero possono introdurre instabilità numerica durante il calcolo dell'inversa, e per mitigare tale problematica, viene adottata una strategia di thresholding, trattando i valori singolari inferiori a una specifica soglia come se fossero nulli. Questa operazione equivale a una riduzione della dimensionalità dello spazio delle feature, con conseguente attenuazione dell'influenza del rumore e della multicollinearità.

Il collegamento con la definizione di Modello Lineare Generalizzato (GLM) si manifesta nel fatto che la regressione lineare standard, implementata da `LinearRegression`, rappresenta un caso specifico di GLM. La regolarizzazione implicita derivante dall'uso della SVD, attraverso il thresholding dei valori singolari, incide sulla stima dei coefficienti β all'interno del framework GLM, stabilizzando la soluzione e riducendo l'impatto di eventuali problematiche legate alla multicollinearità o alla presenza di rumore nei dati. Tuttavia, tale forma di regolarizzazione si caratterizza per la sua debolezza e automaticità, distinguendosi nettamente dalle tecniche di regolarizzazione esplicita quali Ridge, Lasso o Elastic Net, che offrono un controllo parametrico e più incisivo sulla forza della regolarizzazione stessa.

Ridge Regression

La Regressione Ridge², nota anche come regolarizzazione L_2 , rappresenta una tecnica di regolarizzazione fondamentale nell'ambito dei modelli di regressione lineare. Essa nasce dall'esigenza di affrontare problematiche legate alla multicollinearità e all'overfitting, che possono compromettere l'accuratezza e la generalizzabilità dei modelli lineari standard. A differenza della regressione lineare ordinaria, che mira a minimizzare la somma dei quadrati degli errori, la Regressione Ridge introduce un termine di penalizzazione che vincola la grandezza dei coefficienti di regressione, migliorando la stabilità e la robustezza del modello.

Il punto di partenza è il modello di regressione lineare standard $y = X\beta + \varepsilon$, dove y è il vettore delle variabili dipendenti, X è la matrice delle variabili indipendenti (o predittori), β è il vettore dei coefficienti di regressione e ε è il vettore degli errori, assunti come indipendenti e identicamente distribuiti secondo una distribuzione normale con media zero e varianza costante σ^2 .

² Si veda "Ridge Regression: Biased Estimation for Nonorthogonal Problems."

Nella regressione lineare ordinaria, la stima dei coefficienti β si ottiene minimizzando la somma dei quadrati degli errori, definita come:

$$RSS_{OLS} = \sum_{i=1}^n (y_i - x_i^T \beta)^2$$

dove n è il numero di osservazioni e x_i^T è il vettore riga delle variabili predittrici per l' i -esima osservazione. Come già enunciato in precedenza, la soluzione analitica per β è data da:

$$\beta_{OLS} = (X^T X)^{-1} X^T y$$

Tuttavia, quando le variabili predittrici sono altamente correlate (multicollinearità), la matrice $X^T X$ può diventare quasi singolare, rendendo instabile la stima di β . Inoltre, in presenza di un numero elevato di predittori rispetto al numero di osservazioni, il modello può andare in overfitting, perdendo capacità di generalizzare a nuovi dati. La Regressione Ridge affronta questi problemi introducendo un termine di penalizzazione $L2$ nella funzione obiettivo, che assume la seguente forma:

$$RSS_{Ridge} = \sum_{i=1}^n (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

dove p è il numero di predittori e λ è un iperparametro di regolarizzazione non negativo. Il termine $\lambda \sum_{j=1}^p \beta_j^2$ penalizza la somma dei quadrati dei coefficienti, restringendoli verso lo zero. La soluzione per i coefficienti nella Regressione Ridge è data da:

$$\beta_{Ridge} = (X^T X + \lambda I)^{-1} X^T y$$

dove I è la matrice identità $p \times p$. L'aggiunta di λI alla matrice $X^T X$ rende la matrice invertibile anche in presenza di multicollinearità, stabilizzando la stima dei coefficienti, mentre l'iperparametro λ controlla la forza della regolarizzazione. Per valori $\lambda = 0$, la Regressione Ridge si riduce alla regressione lineare ordinaria, mentre per valori di $\lambda > 0$, la penalizzazione sui coefficienti diventa più forte, diminuendo la varianza del modello a scapito di un leggero aumento del bias. La scelta ottimale di λ è cruciale e viene tipicamente effettuata tramite tecniche di cross-validation e ottimizzazione degli iperparametri.

Lasso

La regressione Lasso³ (Least Absolute Shrinkage and Selection Operator) è una tecnica di regolarizzazione per modelli di regressione lineare che, analogamente alla Ridge, mira a migliorare la generalizzabilità e la stabilità delle stime dei coefficienti, soprattutto in presenza di multicollinearità o di un numero elevato di predittori rispetto alle osservazioni. A differenza della Ridge, che utilizza una penalizzazione $L2$ (somma dei quadrati dei coefficienti), Lasso impiega una penalizzazione $L1$ (somma dei valori assoluti dei coefficienti). Questa differenza apparentemente sottile ha conseguenze importanti sul comportamento del modello e sulla sua interpretazione.

Come per la Ridge, il punto di partenza è il modello di regressione lineare standard $y = X\beta + \varepsilon$, ma per la regressione Lasso, la funzione obiettivo assume questa forma:

$$RSS_{Lasso} = \sum_{i=1}^n (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

dove $\sum_{i=1}^n (y_i - x_i^T \beta)^2$ è la somma dei quadrati degli errori (RSS) standard, λ è l'iperparametro di regolarizzazione non negativo e $\lambda \sum_{j=1}^p |\beta_j|$ è la somma dei valori assoluti dei coefficienti, esclusa l'intercetta. Anche in questo caso, la penalizzazione non include l'intercetta per ragioni di interpretazione, invarianza alla traslazione e stabilità numerica.

La differenza cruciale rispetto alla Ridge risiede nel termine di penalizzazione: la penalizzazione $L1$ ha la proprietà di forzare alcuni coefficienti esattamente a zero. Ciò implica che la Lasso effettua una selezione delle feature, escludendo dal modello le variabili meno importanti, rendendola particolarmente utile quando si sospetta che solo un sottoinsieme delle variabili predittive sia effettivamente rilevante per il problema.

A differenza della Ridge, la soluzione della Lasso non ha una forma analitica chiusa, ad eccezione di casi molto specifici. Per trovare i coefficienti β_{Lasso} si utilizzano algoritmi di ottimizzazione iterativi, come Least Angle Regression (LARS) e Coordinate Descent.

Quando $\lambda = 0$, il modello è equivalente alla regressione lineare ordinaria, mentre per valori $\lambda > 0$ la penalizzazione sui coefficienti diventa più forte, portando un numero crescente di coefficienti a zero e riducendo la complessità del modello. La scelta ottimale di λ è cruciale e, come per la Ridge, viene effettuata tramite tecniche di cross-validation e ottimizzazione degli iperparametri.

³ Si veda "Regression Shrinkage and Selection via the Lasso".

Elastic Net

La regressione Elastic Net⁴ rappresenta una tecnica di regolarizzazione che combina le proprietà della Ridge (regolarizzazione $L2$) e della Lasso (regolarizzazione $L1$). Questa combinazione nasce dall'esigenza di superare alcune limitazioni che i due metodi presentano individualmente, offrendo una maggiore flessibilità e robustezza nella gestione di dati complessi, in particolare in presenza di multicollinearità e di un numero elevato di predittori.

Come per Ridge e Lasso, il punto di partenza è il modello di regressione lineare standard $y = X\beta + \varepsilon$, già enunciato in precedenza. La funzione obiettivo che Elastic Net cerca di minimizzare è una combinazione lineare delle penalizzazioni $L1$ e $L2$:

$$RSS_{ElasticNet} = \sum_{i=1}^n (y_i - x_i^T \beta)^2 + \lambda \left[\alpha \sum_{j=1}^p |\beta_j| + (1 - \alpha)/2 * \sum_{j=1}^p \beta_j^2 \right]$$

dove:

- $\sum_{i=1}^n (y_i - x_i^T \beta)^2$ è la somma dei quadrati degli errori (RSS) standard
- λ è l'iperparametro di regolarizzazione non negativo che controlla l'intensità complessiva della penalizzazione
- α è un iperparametro che varia tra 0 e 1 e controlla il bilanciamento tra la penalizzazione $L1$ e $L2$
- $\sum_{j=1}^p |\beta_j|$ è la somma dei valori assoluti dei coefficienti (penalizzazione $L1$)
- $\sum_{j=1}^p \beta_j^2$ è la somma dei quadrati dei coefficienti (penalizzazione $L2$). Anche in questo caso, p rappresenta il numero di predittori, ad esclusione dell'intercetta, che non viene penalizzata.

Il comportamento di Elastic Net varia significativamente al variare del parametro α :

- $\alpha = 0$: Elastic Net equivale a un modello Ridge, ottenendo solamente la penalizzazione $L2$
- $\alpha = 1$: Elastic Net equivale a un modello Lasso, applicando solamente la penalizzazione $L1$
- $0 < \alpha < 1$: Elastic Net combina le penalizzazioni $L1$ e $L2$.

Combinando entrambe le penalizzazioni si ottengono diversi vantaggi. Primo tra tutti, è garantita la gestione della multicollinearità, riducendo quindi l'instabilità delle stime dei

⁴ Si veda "Regularization and Variable Selection Via the Elastic Net".

coefficienti. In secondo luogo, i coefficienti possono assumere valore nullo, operando pertanto una selezione delle feature e semplificando il modello. Grazie alla presenza del parametro α , diventa semplice trovare un compromesso tra Ridge e Lasso, ovviamente sempre attraverso tecniche di cross-validation e ottimizzazione degli iperparametri.

Analogamente a quanto visto per la regressione Lasso, la soluzione di Elastic Net non ha una forma analitica chiusa e richiede l'uso degli stessi algoritmi di ottimizzazione iterativi.

2.1.3 *Bayesian Ridge Regression*

La regressione Ridge Bayesiana⁵ rappresenta un approccio bayesiano alla regressione lineare che incorpora la regolarizzazione $L2$ (nota anche come regolarizzazione di Tikhonov). Come la Ridge Regression frequentista, la sua principale finalità è mitigare il problema dell'overfitting, specialmente in situazioni in cui il numero di predittori è elevato o quando esiste multicollinearità tra essi. Tuttavia, a differenza dell'approccio frequentista, che stima un singolo valore puntuale per ciascun coefficiente di regressione, la Bayesian Ridge Regression stima una distribuzione di probabilità per questi coefficienti, fornendo una quantificazione dell'incertezza associata alle stime.

Per comprendere appieno la Bayesian Ridge Regression, è utile richiamare brevemente la Ridge Regression frequentista. Quest'ultima aggiunge un termine di penalizzazione alla funzione di perdita dei minimi quadrati ordinari (OLS). Questo termine di penalizzazione è proporzionale al quadrato della norma $L2$ dei coefficienti di regressione. In altre parole, la Ridge Regression cerca di minimizzare la somma dei quadrati degli errori di predizione, ma contemporaneamente penalizza valori elevati per i coefficienti. Questo effetto di mitigazione riduce la varianza delle stime dei coefficienti, a scapito di un leggero aumento del bias, portando a una migliore generalizzazione del modello a nuovi dati.

L'approccio bayesiano alla Ridge Regression introduce una prospettiva probabilistica. Invece di considerare i coefficienti di regressione come valori fissi ma sconosciuti, li considera come variabili casuali con una distribuzione di probabilità a priori. Tipicamente, si assume una distribuzione normale (gaussiana) centrata sullo zero per i coefficienti, con una varianza controllata da un parametro, spesso indicato con λ . Questo parametro svolge un ruolo analogo al parametro di regolarizzazione $L2$ nella Ridge Regression frequentista,

⁵ Si veda "Ridge Regression: Biased Estimation for Nonorthogonal Problems" e "Applied Statistical Decision Theory"

in quanto controlla l'entità del restringimento dei coefficienti verso zero. Una varianza a priori più piccola, corrispondente a un valore λ più grande, implica una maggiore penalizzazione e quindi un maggiore restringimento.

Formalmente, si parte da un modello di regressione lineare $y = X\beta + \varepsilon$, dove y rappresenta le variabili dipendenti, X le variabili indipendenti, β i coefficienti di regressione e ε l'errore casuale, assunti distribuiti normalmente con media zero e varianza $\sigma^2 I$, dove I è la matrice identità.

La distribuzione a priori gaussiana dei coefficienti β è definita come $p(\beta | \lambda) = N(0, (\lambda^{-1}I))$, dove λ ne controlla la precisione. Di conseguenza, la funzione di verosimiglianza, che descrive la probabilità dei valori di y condizionati al modello e ai parametri β e σ^2 , assume distribuzione gaussiana $p(y | X, \beta, \sigma^2) = N(X\beta, \sigma^2 I)$. L'obiettivo è calcolare la distribuzione a posteriori dei coefficienti β , dati i valori di y e la distribuzione a priori. Tramite il teorema di Bayes, si ottiene una distribuzione a posteriori anch'essa gaussiana:

$$p(\beta | y, X, \lambda, \sigma^2) = N(\mu_{post}, \Sigma_{post})$$

dove μ_{post} e Σ_{post} rappresentano rispettivamente la media e la matrice di covarianza a posteriori. La media a posteriori stima i coefficienti β considerando sia i dati che la conoscenza a priori, mentre la matrice di covarianza a posteriori quantifica l'incertezza di queste stime. Infine, per predire in un nuovo punto x^* , si calcola la distribuzione predittiva, anch'essa gaussiana.

La media a posteriori, con σ^2 noto, corrisponde alla soluzione della Ridge regression frequentista, mostrando la stretta relazione tra i due approcci:

$$\beta_{Bayesian} = \beta_{Ridge} = (X^T X + \lambda \sigma^2 I)^{-1} X^T y$$

Nel caso in cui invece il valore di σ^2 sia sconosciuto, solitamente si ricorre a una distribuzione a priori $p(\sigma^2) = GammaInversa(\gamma_1, \gamma_2)$, dove γ_1 e γ_2 sono iperparametri che devono necessariamente essere definiti nello spazio degli iperparametri del modello. La distribuzione a posteriori dei coefficienti β diventa pertanto di tipo congiunto:

$$p(\beta, \sigma^2 | y, X, \lambda, \gamma_1, \gamma_2) \propto p(y | X, \beta, \sigma^2) p(\beta | \lambda) p(\sigma^2)$$

Non essendo disponibile una soluzione analitica a questa formula, è necessario ricorrere ad algoritmi numerici come i metodi Monte Carlo o l'approssimazione di Laplace.

La Bayesian Ridge Regression fornisce un quadro più completo rispetto alla Ridge frequentista, stimando una distribuzione a posteriori per i coefficienti, quantificando l'incertezza e permettendo di inferire σ^2 e λ . Inoltre, l'approccio bayesiano offre una maggiore flessibilità nella gestione del parametro λ : invece di doverlo scegliere attraverso tecniche di cross-validation, come avviene nella Ridge frequentista, in un contesto pienamente bayesiano si può assegnare una distribuzione a priori anche a λ e inferire su di esso.

Pur mantenendo una struttura di base affine ai GLM (relazione lineare tra predittori e outcome), la Bayesian Ridge Regression combina i vantaggi della regolarizzazione $L2$ con la flessibilità dell'inferenza bayesiana, fornendo stime dei coefficienti più robuste rispetto alla regressione lineare ordinaria in presenza di multicollinearità e una misura dell'incertezza associata a tali stime.

2.1.4 Support Vector Machine (SVM)

La Support Vector Regression⁶ (SVR) è una tecnica di apprendimento supervisionato impiegata per problemi di regressione, ovvero per la predizione di valori continui. Rappresenta un caso particolare della Support Vector Machine (SVM), la quale è stata sviluppata per i problemi di classificazione, ovvero dei casi in cui la variabile target è qualitativa.

A differenza dei metodi di regressione tradizionali, che cercano di minimizzare la somma degli errori quadratici, la SVR adotta una strategia differente, basata sulla definizione di una fascia intorno alla funzione di regressione. Questa fascia, detta fascia di insensibilità, gioca un ruolo cruciale nel determinare quali errori vengono considerati significativi per l'addestramento del modello e non è presente nella SVM per classificazione. L'idea centrale della SVR consiste nel trovare una funzione che approssimi al meglio i dati, tollerando un certo margine di errore. Immaginando di avere un insieme di punti che rappresentano i nostri dati, l'obiettivo non è trovare una curva che passi esattamente per ogni punto, ma piuttosto una fascia iper-dimensionale sufficientemente ampia da contenere la maggior parte dei punti. Il parametro ε determina l'ampiezza della fascia di insensibilità.

La fascia di insensibilità ε stabilisce un intervallo di tolleranza intorno alla funzione di regressione. In altre parole, tutti i punti che cadono all'interno di questa fascia non

⁶ Si veda “New support vector algorithms” e “Support Vector Machines”

contribuiscono all'errore del modello. Solo i punti che si trovano al di fuori di questa fascia, detti *vettori di supporto*, vengono considerati errori e influenzano la forma della funzione di regressione. Un valore elevato di ε implica una maggiore tolleranza agli errori, consentendo una fascia più ampia, mentre un valore basso di ε richiede una maggiore precisione, con una fascia più stretta.

Un altro parametro fondamentale nella SVR è C , spesso definito come costo o parametro di regolarizzazione (da non confondere con la funzione di costo). Questo parametro controlla il compromesso tra due aspetti cruciali: la complessità del modello e la tolleranza agli errori. Un valore elevato di C penalizza maggiormente gli errori, portando a un modello complesso e rigido, che tende ad adattarsi molto bene ai dati di addestramento, ma che rischia di generalizzare male i nuovi dati, causando overfitting. Al contrario, un valore basso di C permette di tollerare meglio gli errori, ottenendo così un modello più semplice, flessibile e generalizzabile, ma anche meno preciso.

Come già accennato, i vettori di supporto, un concetto chiave delle SVM/SVR, rappresentano i punti che si trovano al di fuori della fascia di insensibilità o sul suo bordo, e influenzano direttamente la forma della funzione di regressione. Un valore elevato di C tende a generare un numero maggiore di vettori di supporto, mentre un valore basso di C ne riduce il numero.

Esistono diverse varianti di SVR, che differiscono per la gestione della linearità e la formulazione del problema di ottimizzazione.

La forma più semplice è la *SVR Lineare*. In questo caso, l'algoritmo cerca una funzione lineare, rappresentata da un iperpiano (una retta in $2D$, un piano in $3D$, un iperpiano in dimensioni superiori) nello spazio delle feature, che approssimi linearmente i valori target. A differenza della regressione lineare ordinaria (OLS), che minimizza la somma dei quadrati degli errori, le SVR lineari minimizzano l'errore all'interno della fascia di insensibilità. Solo gli errori di stima che superano questa soglia vengono considerati nella funzione di costo e penalizzati linearmente. Questa caratteristica conferisce alle SVR lineari una notevole resistenza agli outlier. Il concetto di linearità presuppone la possibilità di separare linearmente le feature, cosa possibile solo in assenza di multicollinearità.

La *SVR con Kernel* estende le capacità delle SVR lineari a problemi in cui i dati non sono linearmente separabili. L'idea chiave è di mappare i dati in uno spazio di feature di dimensione superiore (anche infinita) attraverso una *funzione kernel*. In questo nuovo spazio,

i dati potrebbero essere linearmente separabili, permettendo l'applicazione delle tecniche delle SVR lineari. La funzione kernel consente di calcolare il prodotto scalare tra i dati in questo spazio di feature senza doverli esplicitamente trasformare, con un notevole vantaggio computazionale. Di seguito sono riportate alcune tra le funzioni Kernel più utilizzate:

- *Kernel lineare*: si tratta del kernel più semplice e corrisponde a una trasformazione identità, dunque non mappa i dati in uno spazio di feature di dimensione superiore. È adatto quando i dati sono linearmente separabili (o approssimativamente tali). La sua formula è: $K(x, y) = x \cdot y$, dove x e y sono i vettori di input e $\cdot$ rappresenta il prodotto scalare.
- *Kernel polinomiale*: questo kernel mappa i dati in uno spazio di feature di dimensione superiore attraverso una trasformazione polinomiale. È definito da due parametri: il grado del polinomio (d) e un coefficiente (c). La sua formula è $K(x, y) = (x \cdot y + c)^d$, dove d è un intero positivo che controlla il grado del polinomio e c è un coefficiente che influenza l'offset. Un grado più alto permette di modellare relazioni più complesse, ma aumenta anche il rischio di overfitting.
- *Kernel gaussiano (RBF - Radial Basis Function)*: rappresenta uno dei kernel più utilizzati, grazie alla sua capacità di modellare relazioni non lineari complesse. Mappa i dati in uno spazio di feature di dimensione infinita. È definito da un parametro γ che controlla la larghezza del kernel ed è definito come

$$K(x, y) = \exp(-||x - y||^2 / (2\sigma^2)) = \exp(-\gamma ||x - y||^2)$$
 dove $||x - y||^2$ è il quadrato della distanza euclidea tra x e y , e $\gamma = 1/(2\sigma^2)$. Un valore di γ piccolo rende il kernel più lasco, mentre un valore di γ grande lo restringe. Un kernel più lasco considera influenti anche i punti distanti, mentre un kernel più stretto dà maggiore importanza ai punti vicini.
- *Kernel sigmoide*: questo kernel è ispirato alla funzione sigmoide utilizzata spesso nelle reti neurali. La sua formula è $K(x, y) = \tanh(\alpha x \cdot y + c)$, dove α e c sono parametri. Sebbene sia ispirato alle reti neurali, non è comunemente utilizzato come il kernel RBF o polinomiale.
- *Kernel Matérn*: è una potente e complessa generalizzazione del kernel RBF, con un parametro aggiuntivo che controlla la "regolarità" della funzione.

Le potenzialità delle SVR sono legate anche alla possibilità di combinare kernel diversi, adattando l'algoritmo a eventuali caratteristiche intrinseche specifiche del fenomeno studiato. Infatti, la scelta del kernel dipende dal problema specifico e dalla natura dei dati.

Come già accennato, se si sospetta che i dati non siano linearmente separabili e presentino relazioni non lineari, il kernel polinomiale o RBF sono le opzioni più comuni. Il kernel RBF è spesso una buona scelta di default, grazie alla sua flessibilità e alla presenza di un solo parametro da ottimizzare (γ). Il kernel sigmoide è meno utilizzato e spesso produce risultati simili al kernel lineare.

L'utilizzo del kernel necessita di una fase ulteriore di ottimizzare dei parametri (ad esempio, il grado d per il kernel polinomiale e γ per il kernel RBF) attraverso tecniche di validazione incrociata per ottenere le migliori prestazioni del modello. Ciò complica notevolmente le operazioni di training e appesantisce ulteriormente il carico computazionale.

Una terza variante algoritmica molto utilizzata è la *Nu-SVR*, la quale introduce un nuovo parametro ν che offre un'interpretazione alternativa al parametro C delle SVM standard, fornendo un controllo più intuitivo sul numero di vettori di supporto e sulla frazione di errori di training. Questo parametro può assumere valori compresi tra 0 e 1 e possiede due importanti proprietà:

- ν è un limite superiore alla frazione di errori di training;
- ν è un limite inferiore alla frazione di vettori di supporto.

In pratica, scegliendo un valore di ν , si impone un limite alla percentuale di errori che il modello può commettere sui dati di training e, contemporaneamente, si garantisce un numero minimo di vettori di supporto.

Esiste inoltre una relazione (non diretta) tra ν e C : diminuendo ν si ottiene un effetto simile all'aumentare C , incrementando l'effetto della regolarizzazione, mentre aumentando ν si ottiene l'effetto opposto, ovvero una minore regolarizzazione e una riduzione dell'errore di stima.

Mentre i GLM si basano sulla massimizzazione della verosimiglianza di una distribuzione di probabilità, le SVR si basano sulla minimizzazione dell'errore entro una fascia di insensibilità e sulla regolarizzazione, non rendendo necessarie assunzioni distributive esplicite sulla variabile risposta o sugli errori. Inoltre, la presenza della fascia di insensibilità garantisce maggiore robustezza alle SVR. Per quanto riguarda la non linearità, le SVR permettono di definire e ottimizzare funzioni kernel (o combinazioni di funzioni kernel), mentre i GLM usano una funzione di link.

Per costruzione, le SVR rappresentano una valida alternativa ai GLM per la modellazione di variabili continue, risultando particolarmente efficaci in presenza di relazioni non lineari e

di outlier. Tuttavia, la loro interpretazione può essere più complessa rispetto ai GLM, specialmente con kernel non lineari, e la stima del modello risulta solitamente meno efficiente a livello computazionale. Ciò è reso ancora più complesso dalla necessità di individuare valori ottimali per gli iperparametri del kernel.

2.1.5 *K-Nearest Neighbors (KNN)*

Il metodo K-Nearest Neighbors⁷ (KNN) rappresenta un approccio non parametrico all'apprendimento supervisionato, impiegato sia in contesti di classificazione che di regressione. La sua semplicità concettuale e la sua versatilità lo rendono uno strumento ampiamente utilizzato. L'interesse è qui rivolto alla formulazione specifica per problemi di regressione.

A differenza di altri algoritmi che costruiscono un modello esplicito a partire dai dati di addestramento, KNN si basa su un approccio detto *lazy-learning*: il calcolo viene rimandato al momento della predizione, senza una fase di training vera e propria. Quando si desidera predire il valore di una nuova istanza, l'algoritmo ricerca i K valori più simili presenti nel dataset di addestramento e ne utilizza i valori target per effettuare la predizione.

Il concetto di similitudine tra le osservazioni è cruciale nel KNN e viene quantificato attraverso una metrica di distanza. Date due istanze x e y , dove $x = (x_1, x_2, \dots, x_n)$ e $y = (y_1, y_2, \dots, y_n)$ sono vettori n -dimensionali, le metriche più comuni includono:

- *Distanza Euclidea*: $d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$. Questa metrica calcola la distanza geometrica in linea retta tra due punti nello spazio n -dimensionale, ovvero il percorso più breve.
- *Distanza di Manhattan (o distanza L1)*: $d(x, y) = \sum_i |x_i - y_i|$. Questa metrica calcola la somma delle differenze assolute tra le coordinate dei due punti. Intuitivamente, equivale a calcolare la distanza lungo un percorso a scalini, simile al tragitto che percorrerebbe un pedone lungo i marciapiedi di una città.
- *Distanza di Minkowski*: $d(x, y) = (\sum_i |x_i - y_i|^p)^{1/p}$. Rappresenta una generalizzazione delle distanze euclidea ($p = 2$) e di Manhattan ($p = 1$).

La scelta della metrica appropriata dipende dalla natura dei dati e dal problema specifico. Ad esempio, la distanza euclidea è particolarmente adatta per dati continui in spazi

⁷ Si veda "Nearest neighbor pattern classification".

multidimensionali, mentre la distanza di Manhattan può essere preferibile in presenza di dati categoriali o quando si desidera dare meno peso agli outlier.

Una volta definita la metrica di distanza, l'algoritmo KNN procede come segue: dato un nuovo punto x' di cui si vuole predire il valore, vengono calcolate le distanze tra questo punto e tutti i punti presenti nel dataset di addestramento. Vengono quindi selezionati i K punti più vicini, ovvero quelli con le distanze minori. Per la regressione, la predizione del valore target $\hat{y}$ per il nuovo punto x' viene ottenuta calcolando la media (o la mediana) dei valori target delle K osservazioni vicine selezionate: $\hat{y} = (1/K) \sum_i y_i$, dove y_i sono i valori target delle K osservazioni prossime a x' .

Il parametro K gioca un ruolo fondamentale nelle prestazioni dell'algoritmo KNN. La scelta di un valore appropriato di K è cruciale e influenza il bilanciamento tra bias e varianza del modello. Un valore di K troppo piccolo rende il modello sensibile al rumore presente nei dati di addestramento, portando a un'elevata varianza e potenzialmente all'overfitting. Al contrario, un valore di K troppo grande può produrre stime eccessivamente imprecise, portando a un alto bias e all'underfitting.

La scelta ottimale di K dipende dalla distribuzione dei dati e dalla complessità del problema. Tecniche come la cross-validation, descritta nel prossimo capitolo, possono essere impiegate per determinare empiricamente il valore di K che minimizza l'errore di predizione su dati non visti durante l'addestramento.

Un aspetto importante da considerare nell'utilizzo del KNN è la necessità di normalizzare o standardizzare i dati, soprattutto quando le feature presentano scale differenti. Ciò si rende necessario perché le metriche di distanza sono sensibili alla scala delle variabili. Se una feature ha valori molto più grandi di altre, la sua influenza nel calcolo della distanza sarà preponderante, indipendentemente dalla sua effettiva rilevanza per il problema.

Nonostante la sua semplicità, KNN presenta alcuni limiti, tra cui l'alto costo computazionale in fase di predizione, soprattutto con dataset di grandi dimensioni, e la sensibilità alla presenza di feature irrilevanti. Tuttavia, grazie alla sua natura non parametrica e alla capacità di adattarsi a diverse tipologie di dati, KNN rimane uno strumento prezioso nel panorama dell'apprendimento supervisionato per la regressione. Rispetto ai modelli GLM, KNN è meno suscettibile alla multicollinearità e non stima coefficienti per le variabili predittive. Il suo funzionamento si basa sul calcolo della distanza tra le istanze nello spazio delle feature.

Pertanto, la multicollinearità non causa problemi di instabilità o inaffidabilità delle stime dei coefficienti, tuttavia possono presentarsi le seguenti problematiche:

- distorsione del calcolo della distanza: quando due o più feature sono altamente correlate, la loro influenza nel calcolo della distanza euclidea (o altre metriche di distanza) viene amplificata. Questo perché le feature correlate impattano maggiormente nel calcolo della distanza, distorcendo la reale vicinanza tra le istanze, discriminando l'informazione apportata dalle altre feature;
- aumento della dimensionalità effettiva: la presenza di multicollinearità aumenta la dimensionalità effettiva del problema. Le feature correlate portano informazioni ridondanti, ma vengono comunque considerate come dimensioni separate nel calcolo della distanza. Ciò può esacerbare gli effetti della dimensionalità, con le conseguenze negative che ne derivano;
- difficoltà nell'interpretazione: anche se KNN non produce coefficienti interpretabili, in presenza di multicollinearità, l'influenza delle feature correlate è sovrapposta, rendendo più difficile capire quali feature siano effettivamente rilevanti per la predizione.

Per contenere gli effetti della multicollinearità, è possibile adottare la distanza di Minkowski. Questa metrica fa uso di un vettore di pesi per le feature, introducendo tuttavia un ulteriore livello di complessità dato dalla necessità di individuare un valore ottimale per i pesi. La distanza di Mahalanobis, invece, risulta più flessibile e comunque valida per affrontare questo genere di situazioni. A differenza della distanza euclidea, che tratta tutte le dimensioni come indipendenti e con la stessa scala, la distanza di Mahalanobis tiene conto della matrice di covarianza delle variabili. La sua formula, calcolata tra due vettori x e y , è definita come $D(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)}$, dove Σ è la matrice di covarianza delle variabili.

Nonostante la mancanza di una qualche funzione di regolarizzazione e la presenza di una certa suscettibilità alla multicollinearità, KNN rappresenta un valido algoritmo per il problema in studio in quanto ha la capacità teorica di modellare le relazioni non lineari tra feature senza imporre una forma funzionale, basandosi unicamente sul concetto di similitudine.

2.1.6 Modelli ad albero

Per introdurre gli algoritmi Random Forest, XGBoost e Hist Gradient Boosting (HGB), è opportuno richiamare i concetti fondamentali relativi agli alberi decisionali. Un albero decisionale è una struttura gerarchica ad albero in cui ciascun nodo interno rappresenta un test su una variabile predittiva (feature), ogni ramo un possibile esito del test e ogni nodo foglia una decisione (classe o valore predetto). La costruzione di un albero decisionale avviene in maniera ricorsiva, selezionando ad ogni stadio la variabile che massimizza la separazione dei dati in base a una metrica di impurità (ad esempio, l'entropia o l'indice di Gini per la classificazione, o la varianza per la regressione).

Gli alberi decisionali, pur essendo intuitivi e di facile interpretazione, presentano alcune limitazioni: la tendenza all'overfitting, ovvero l'eccessivo adattamento ai dati di addestramento con conseguente scarsa generalizzazione su nuovi dati, e l'instabilità, per cui piccole variazioni nei dati di addestramento possono generare alberi decisionali notevolmente differenti.

2.1.7 Random Forest

L'algoritmo Random Forest si configura come una potente tecnica di apprendimento supervisionato, impiegata con successo in svariati contesti sia di classificazione che di regressione. Ideato da Leo Breiman⁸, questo metodo ensemble sfrutta la combinazione di molteplici alberi decisionali per ottenere previsioni più accurate e robuste rispetto a un singolo albero.

Le Random Forest superano le suddette limitazioni degli alberi decisionali singoli attraverso due meccanismi cardine: il *bagging* (Bootstrap Aggregating) e il *random subspace* (selezione casuale delle feature). Il bagging consiste nella creazione di molteplici sottoinsiemi dei dati di addestramento mediante campionamento con reinserimento (bootstrap). Per ciascun sottoinsieme, viene costruito un albero decisionale. Tale processo riduce la varianza del modello, poiché ogni albero viene addestrato su un campione differente dei dati. Il random subspace, invece, prevede che durante la costruzione di ogni albero, ad ogni nodo venga considerato solo un sottoinsieme casuale delle variabili predittive disponibili per la suddivisione. Questo introduce ulteriore diversità tra gli alberi, riducendone la correlazione e migliorando ulteriormente la capacità di generalizzazione.

⁸ Breiman (2001). "Random Forests".

Combinando questi due meccanismi, le Random Forest generano una *foresta* (un insieme) di alberi decisionali debolmente correlati tra loro. La predizione finale si ottiene aggregando le predizioni dei singoli alberi: nel caso di classificazione, si seleziona la classe con la maggioranza dei voti (*voting*); nel caso di regressione, si calcola la media (o la mediana, per una maggiore robustezza agli outlier) delle predizioni. Formalmente, per la regressione, la predizione aggregata $\hat{y}$ può essere espressa come $\hat{y} = (1/N) \sum_i y_i$, dove N è il numero di alberi nella foresta e y_i è la predizione dell' i -esimo albero.

La costruzione di una Random Forest si articola nelle seguenti fasi:

1. campionamento Bootstrap: vengono creati N sottoinsiemi di dati mediante campionamento con reinserimento dal dataset originale;
2. costruzione degli alberi. Per ogni sottoinsieme:
 - a. viene costruito un albero decisionale;
 - b. viene selezionato un sottoinsieme casuale di m variabili predittive tra le M totali (con $m \ll M$);
 - c. viene selezionata la migliore variabile tra le m considerate per la suddivisione del nodo.
 - d. aggregazione delle predizioni: la predizione finale si ottiene tramite voting (classificazione) o media/mediana (regressione) delle predizioni dei singoli alberi.

L'algoritmo Random Forest prevede alcuni parametri determinanti nelle prestazioni del modello, come ad esempio il numero di alberi totale, il numero di variabili predittive considerate ad ogni nodo per la suddivisione, la profondità massima degli alberi, il numero minimo di campioni richiesti per dividere un nodo interno e il numero minimo di campioni richiesti per appartenere a un nodo foglia.

La tecnica cross-validation, unitamente all'ottimizzazione degli iperparametri, consente di individuare la combinazione che più minimizza l'errore.

Le Random Forest presentano numerosi vantaggi, tra cui l'elevata accuratezza predittiva, la robustezza all'overfitting grazie al bagging e al random subspace, la capacità di gestire dati ad alta dimensionalità, la stima dell'importanza relativa (*feature importance*) delle variabili predittive e la gestione di dati mancanti senza richiedere pre-elaborazioni complesse. Rispetto a modelli più tradizionali e ai singoli alberi decisionali, le Random Forest annoverano una minore interpretabilità e un costo computazionale potenzialmente elevato,

soprattutto con dataset di grandi dimensioni, ma rimangono più interpretabili rispetto ad altri modelli ad albero che presentano complessità ben differenti.

2.1.8 XGBoost

XGBoost⁹ (eXtreme Gradient Boosting) si configura come un potente algoritmo di apprendimento automatico, specificamente ottimizzato per performance e velocità, che trova ampia applicazione sia in problemi di classificazione che di regressione. Basato sul principio del *gradient boosting*, XGBoost introduce significative ottimizzazioni che ne incrementano l'efficacia e l'efficienza computazionale.

Il gradient boosting è un metodo ensemble che combina molteplici modelli deboli (tipicamente alberi decisionali) per formare un modello forte. A differenza del bagging, in cui i modelli vengono addestrati in parallelo su sottoinsiemi diversi dei dati, nel gradient boosting i modelli vengono addestrati sequenzialmente, con ogni nuovo modello che cerca di correggere gli errori commessi dai modelli precedenti.

Formalmente, dato un dataset di training $\{(x_i, y_i)\}$ con $i = 1, \dots, n$, dove x_i rappresenta il vettore delle feature e y_i il valore target, l'obiettivo è apprendere una funzione $F(x)$ che minimizzi una funzione di perdita $L(y, F(x))$. Nel gradient boosting, la funzione $F(x)$ viene costruita iterativamente come somma di alberi decisionali, da cui

$$F_m(x) = F_{m-1}(x) + h_m(x)$$

dove $F_m(x)$ è il modello al m-esimo passo, $F_{m-1}(x)$ è il modello al passo precedente e $h_m(x)$ è l'albero decisionale che viene aggiunto al modello. L'albero $h_m(x)$ viene addestrato per minimizzare il gradiente della funzione di perdita rispetto alle predizioni del modello precedente.

XGBoost introduce diverse ottimizzazioni rispetto al gradient boosting tradizionale, che ne determinano le elevate prestazioni:

- *regolarizzazione*: XGBoost include termini di regolarizzazione $L1$ e $L2$ nella funzione obiettivo per prevenire l'overfitting. Questo controllo della complessità del modello è cruciale per ottenere una buona generalizzazione su dati non visti. La funzione obiettivo in XGBoost può essere espressa come

$$Obj = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$

⁹ Si veda "A Scalable Tree Boosting System Tianqi Chen"

dove L è la funzione di perdita, $\hat{y}_i$ è la predizione del modello per l' i -esimo campione, f_k rappresenta il k -esimo albero e $\Omega(f_k)$ è il termine di regolarizzazione per il k -esimo albero;

- *gestione di dati sparsi*: XGBoost gestisce nativamente i dati sparsi, ottimizzando l'apprendimento anche in presenza di valori mancanti;
- *parallelizzazione*: XGBoost sfrutta la parallelizzazione per accelerare sia la fase di training che quella di predizione e dispone di implementazioni ottimizzate per sistemi di elaborazione distribuiti ad alte prestazioni;
- *ottimizzazione della funzione obiettivo*: XGBoost utilizza un'approssimazione di secondo ordine della funzione obiettivo tramite espansione di Taylor per una convergenza più rapida.

Nell'applicazione di XGBoost alla regressione, la funzione di perdita $L(y, \hat{y})$ viene scelta arbitrariamente. Solitamente viene utilizzato l'errore quadratico medio (MSE), da cui $L(y, \hat{y}) = (y - \hat{y})^2$, oppure l'errore assoluto medio (MAE), da cui $L(y, \hat{y}) = |y - \hat{y}|$. La scelta della funzione di perdita dipende dal problema specifico e dalle caratteristiche dei dati.

Come per le Random Forest, l'algoritmo XGBoost è influenzato da svariati iperparametri, i quali devono essere opportunamente selezionati tramite cross-validation e ottimizzazione. Tra i più importanti vi sono: il numero di alberi da costruire; il tasso di apprendimento, che controlla il contributo di ogni albero alla predizione finale; la profondità massima degli alberi; la frazione di campioni di training utilizzati per addestrare ogni albero; la frazione di feature utilizzate per addestrare ogni albero; i parametri di regolarizzazione $L1$ e $L2$, formalmente definiti come α e λ .

XGBoost offre numerosi vantaggi in contesti di regressione: elevata accuratezza predittiva, robustezza all'overfitting grazie alla regolarizzazione, capacità di gestire dati ad alta dimensionalità e dati sparsi, elevata velocità di training e predizione grazie alla parallelizzazione. Come per le Random Forest, tra i potenziali svantaggi si annoverano una minore interpretabilità rispetto a modelli lineari e la necessità di ottimizzare un numero elevato di parametri per ottenere le migliori prestazioni.

2.1.9 Hist Gradient Boosting (HGB)

L'Hist Gradient Boosting¹⁰ (HGB) rappresenta una variante ottimizzata del Gradient Boosting, particolarmente efficace in termini di velocità e gestione della memoria, specialmente con dataset di grandi dimensioni. Questa tecnica si distingue per l'utilizzo di istogrammi per approssimare la distribuzione delle feature, accelerando notevolmente il processo di ricerca delle migliori suddivisioni degli alberi.

Come precedentemente discusso, il gradient boosting è un metodo ensemble che combina molteplici modelli deboli, tipicamente alberi decisionali, in modo sequenziale. Ogni nuovo albero viene addestrato per correggere gli errori commessi dagli alberi precedenti, minimizzando il gradiente della funzione di perdita rispetto alle predizioni del modello corrente. La predizione finale è ottenuta sommando le predizioni di tutti gli alberi.

Formalmente, data una funzione di perdita $L(y, F(x))$, dove y è il valore target e $F(x)$ è la predizione del modello, l'obiettivo è trovare una $F(x)$ che minimizzi L . Nel gradient boosting, $F(x)$ viene costruita iterativamente in modo tale che $F_m(x) = F_{m-1}(x) + h_m(x)$, dove $h_m(x)$ è l'albero aggiunto al m-esimo passo, addestrato per approssimare il negativo del gradiente della funzione di perdita rispetto a $F_{m-1}(x)$.

La principale innovazione di HGB risiede nell'utilizzo di istogrammi per approssimare la distribuzione delle feature. Invece di considerare tutti i possibili valori di una feature per trovare la migliore suddivisione di un nodo, HGB suddivide i valori in intervalli, detti *bin*, e costruisce un istogramma, riducendo drasticamente il numero di possibili suddivisioni da valutare e accelerando il processo di training. L'aumento di velocità è dato dal fatto che la costruzione degli istogrammi è molto più veloce rispetto all'ordinamento completo dei dati e richiedono meno memoria rispetto alla memorizzazione di tutti i valori unici delle feature.

Grazie all'utilizzo degli istogrammi avviene inoltre un processo di regolarizzazione implicita. Infatti, attraverso la discretizzazione dei valori delle feature, si riduce la complessità del modello, aumentando l'effetto di smoothing, che è una naturale conseguenza della regolarizzazione. HGB prevede anche la presenza di un termine di regolarizzazione $L2$ esplicito all'interno della funzione obiettivo, che serve ad applicare una penalizzazione diretta ai pesi degli alberi, analogamente a quanto avviene nella Ridge Regression.

¹⁰ Si veda "LightGBM: A Highly Efficient Gradient Boosting Decision Tree"

Nel contesto della regressione, e analogamente a XGBoost, HGB utilizza una funzione di perdita appropriata, come l'errore quadratico medio (MSE) o l'errore assoluto medio (MAE). La costruzione degli alberi avviene seguendo i seguenti passi:

1. per ogni feature, viene costruito un istogramma;
2. per ogni nodo dell'albero, viene valutata la riduzione della funzione di perdita per ogni possibile suddivisione basata sugli istogrammi;
3. viene scelta la suddivisione che massimizza la riduzione della funzione di perdita.
4. il processo si ripete ricorsivamente per i nodi figli fino a raggiungere un criterio di arresto (ad esempio, profondità massima dell'albero o numero minimo di campioni in un nodo foglia).

Tra gli iperparametri più importanti di HGB, si trovano: il numero massimo di iterazioni, ovvero il numero di alberi costruiti; il *learning rate*, o tasso di apprendimento; la profondità massima degli alberi; il termine di regolarizzazione *L2*; il numero minimo di campioni in un nodo foglia; il numero di intervalli (bin) per la costruzione degli istogrammi.

Riassumendo, HGB offre significativi vantaggi rispetto ad algoritmi più tradizionali, come ad esempio un'elevata velocità di training, un basso consumo di memoria, una buona accuratezza predittiva, comparabile comunque ad altri algoritmi di gradient boosting, e la gestione nativa di feature categoriche. Tuttavia, HGB presenta una potenziale minore accuratezza rispetto a gradient boosting classici su dataset molto piccoli, dove l'approssimazione con istogrammi può portare a una perdita di informazione. Si tratta in ogni caso di un algoritmo impiegato in diversi ambiti come la previsione della domanda, la modellizzazione di dati finanziari e l'analisi di dati genomici.

2.1.10 Reti Neurali MLP

Le reti neurali multistrato¹¹ (MLP, Multilayer Perceptron) rappresentano una classe fondamentale di modelli di apprendimento automatico, ampiamente utilizzate in svariati contesti, tra cui la regressione. Caratterizzate da una struttura a strati interconnessi di neuroni artificiali, le MLP sono in grado di apprendere relazioni complesse tra input e output, offrendo un'alternativa potente ai metodi di regressione tradizionali.

Una MLP è composta da almeno tre strati di neuroni: uno strato di input, uno o più strati nascosti e uno strato di output. Ogni neurone di uno strato è connesso a tutti i neuroni dello

¹¹ Si veda "Learning internal representations by error propagation".

strato successivo attraverso connessioni pesate. Lo strato di input riceve, per ciascuna osservazione, le feature in ingresso, rappresentate da un vettore $x = (x_1, x_2, \dots, x_n)$, dove n è il numero di feature. Gli strati nascosti eseguono trasformazioni non lineari sui dati di input. Ogni neurone in uno strato nascosto calcola una somma pesata degli input ricevuti dai neuroni dello strato precedente, a cui applica una funzione di attivazione non lineare. Lo strato di output produce la predizione del modello. Nel caso della regressione, lo strato di output contiene tipicamente un singolo neurone con una funzione di attivazione lineare (o identità), generando quindi un valore numerico continuo.

Il processo di calcolo di una MLP può essere descritto come segue:

- *forward propagation* (propagazione in avanti): l'input x viene propagato attraverso la rete, strato per strato. Per ogni neurone j in uno strato l , il valore di attivazione $a_j^{(l)}$ viene calcolato come $a_j^{(l)} = g(\sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)})$, dove $w_{ij}^{(l)}$ è il peso della connessione tra il neurone i nello strato $l - 1$ e il neurone j nello strato l , $b_j^{(l)}$ è il bias del neurone j nello strato l , e g è la funzione di attivazione. Per lo strato di input, ovvero il primo strato, vale la condizione $a_i^{(0)} = x_i$;
- *funzione di attivazione*: le funzioni di attivazione introducono non linearità nella rete, permettendo alla rete di apprendere relazioni complesse. Esempi comuni includono la funzione sigmoide, la tangente iperbolica ($\tanh$) e la ReLU (Rectified Linear Unit). Per problemi di regressione, nello strato si usa tipicamente una funzione lineare $g(z) = z$;
- *loss function* (funzione di perdita): per addestrare la rete è necessario definire una funzione di perdita che quantifica l'errore tra le predizioni del modello e i valori target reali. Per la regressione, si utilizzano comunemente l'errore quadratico medio $MSE = (1/N) \sum_i (y_i - \hat{y}_i)^2$ o l'errore assoluto medio $MAE = (1/N) \sum_i |y_i - \hat{y}_i|$, dove y_i è il valore target reale per l' i -esimo campione e $\hat{y}_i$ è la predizione del modello;
- *backpropagation* (retropropagazione): l'algoritmo di backpropagation calcola il gradiente della funzione di perdita rispetto ai pesi della rete. Questo gradiente viene utilizzato per aggiornare i pesi attraverso un algoritmo di ottimizzazione, come la discesa del gradiente stocastica (SGD) o varianti più avanzate come Adam.

L'addestramento di una MLP per la regressione consiste nell'ottimizzare i pesi della rete per minimizzare la funzione di perdita sui dati di training. Questo processo iterativo prevede:

1. inizializzazione dei pesi: i pesi vengono inizializzati casualmente;
2. propagazione in avanti: l'input viene propagato attraverso la rete per ottenere le predizioni;
3. calcolo della funzione di perdita: viene calcolato l'errore tra le predizioni e i valori target;
4. retropropagazione: viene calcolato il gradiente della perdita rispetto ai pesi;
5. aggiornamento dei pesi tramite un algoritmo di ottimizzazione.

L'intero processo viene ripetuto per un certo numero di iterazioni (*epochs*) sull'intero dataset di training o fino a quando non si raggiunge un criterio di convergenza.

Gli iperparametri usati nelle reti MLP solitamente sono: il numero di strati nascosti e il numero di neuroni per strato, necessari per definire le capacità del modello; la funzione di attivazione, utile per determinare la non linearità della rete; funzione di perdita, usata per misurare l'errore del modello; l'algoritmo di ottimizzazione e il learning rate, impiegati dalla rete per regolare e affinare il processo di apprendimento e convergenza.

Per quanto riguarda la regolarizzazione, è bene considerare che, per quanto concettualmente semplice, il ricorso a questa tecnica dipende dall'implementazione tecnica della rete. Solitamente è possibile introdurre le regolarizzazioni di tipo *L1* (Lasso) e *L2* (Ridge) in modo arbitrario all'interno di ciascun layer della rete. È disponibile una terza tecnica di regolarizzazione che prevede l'uso di un layer di *dropout* che ha il compito di passare al layer successivo solo una quota di neuroni selezionati casualmente, mitigando quindi la tendenza all'overfitting della rete con una sorta di degrado controllato della stessa.

Le MLP dimostrano dunque ottime capacità di apprendere relazioni non lineari complesse, un'elevata flessibilità nell'adattarsi a diversi tipi di dati e la possibilità di essere applicate a problemi di regressione con input ad alta dimensionalità. Risulta tuttavia necessario effettuare un processo di ottimizzazione degli iperparametri che aumenta in complessità computazionale all'aumentare della complessità architetturale della rete. Le reti neurali inoltre sono suscettibili all'overfitting, ma esistono tecniche potenzialmente molto adattabili che riescono a mitigare questo problema. L'evoluzione tecnologica degli ultimi anni ha portato alla comparsa di software e hardware in grado di stimare ed eseguire reti neurali con prestazioni mai viste prima.

2.1.11 Gaussian Process Regressor (GPR)

La regressione con processi gaussiani¹² (GPR, Gaussian Process Regression) rappresenta un potente approccio non parametrico per la regressione, caratterizzato dalla capacità di fornire non solo una predizione puntuale, ma anche una stima dell'incertezza associata a tale predizione. A differenza dei modelli parametrici, che assumono una forma funzionale predefinita, i GPR definiscono una distribuzione di probabilità sullo spazio delle funzioni, offrendo una flessibilità notevole nell'adattarsi a diverse tipologie di dati.

Un processo gaussiano è una collezione di variabili casuali, un numero finito delle quali ha una distribuzione congiunta gaussiana. Formalmente, dato un insieme di indici T (che può essere continuo, come nel caso della regressione, o discreto), un processo gaussiano è definito dalla sua funzione media $\mu(t)$ e dalla sua funzione di covarianza $k(t, t')$, dove t e $t' \in T$. La funzione media definisce il valore atteso del processo in ogni punto t tale che $\mu(t) = E[f(t)]$, mentre la funzione di covarianza quantifica la correlazione tra i valori del processo in due punti t e t' :

$$k(t, t') = \text{Cov}[f(t), f(t')] = E[(f(t) - \mu(t))(f(t') - \mu(t'))]$$

Si assume che le osservazioni disponibili, rappresentate da coppie di valori input-output (x_i, y_i) , con i che varia da 1 a n (dove n è il numero totale di osservazioni), derivino da una funzione sottostante, spesso indicata con $f(x)$, che descrive la relazione tra input e output. Tuttavia, le misurazioni reali sono affette da rumore, che si modella tipicamente come rumore gaussiano, indicato con ε . Ogni valore osservato y_i può essere visto come la somma del valore reale, non osservabile, della funzione $f(x_i)$ e di una componente di rumore ε_i . Formalmente, si può esprimere questa relazione come $y_i = f(x_i) + \varepsilon_i$, dove ε_i segue una distribuzione normale (gaussiana) con media zero e varianza σ_n^2 . Quest'ultima quantifica l'incertezza associata alle misurazioni.

L'obiettivo principale della Regressione con Processi Gaussiani (GPR) è quello di inferire, a partire dai dati osservati, la distribuzione di probabilità della funzione $f(x)$. In altre parole, si cerca di determinare quali funzioni sono più plausibili alla luce dei dati a disposizione. A differenza di altri metodi di regressione che cercano di trovare una singola funzione ottimale, i GPR forniscono una distribuzione di probabilità sull'intero spazio delle funzioni, permettendo di quantificare l'incertezza associata alla stima di $f(x)$ in ogni punto.

¹² Si veda “Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow” – Capitolo 5

Per effettuare predizioni in nuovi punti, indicati con X^* , si calcola la distribuzione predittiva $p(f^* | X, y, X^*)$. Questa distribuzione rappresenta la probabilità dei valori della funzione f nei nuovi punti X^* , condizionata dai dati di training X e y . Un aspetto fondamentale dei GPR è che, grazie alle proprietà delle distribuzioni gaussiane, questa distribuzione predittiva risulta essere anch'essa gaussiana. Ciò significa che la predizione in ogni nuovo punto x^* è una distribuzione normale, caratterizzata da una media e una varianza. La media rappresenta la predizione puntuale del modello, mentre la varianza quantifica l'incertezza associata a tale predizione.

In termini più intuitivi, si può pensare che i GPR, osservando i dati di training, imparino a replicare la forma della funzione $f(x)$ e, allo stesso tempo, a quantificare quanto sono certi di questa stima. Nelle regioni dello spazio delle osservazioni in cui ci sono molti dati di training, la varianza della predizione sarà bassa, indicando un'alta confidenza nella stima. Al contrario, nelle regioni dove ci sono pochi dati o dove i dati sono più sparsi, la varianza sarà alta, riflettendo una maggiore incertezza. Questo è uno dei vantaggi principali dei GPR rispetto ad altri metodi di regressione, che tipicamente forniscono solo una predizione puntuale senza una misura di incertezza.

La GPR, attraverso l'uso di processi gaussiani, permette quindi di inferire una distribuzione di probabilità sull'intero spazio delle funzioni, fornendo non solo predizioni puntuali, ma anche una quantificazione dell'incertezza associata a tali predizioni.

Un aspetto importante da sottolineare è il ruolo della funzione kernel o, similmente, della funzione di covarianza. Questa funzione definisce la similarità tra i punti dati e la forma della distribuzione sulle funzioni, ovvero quanto regolari o irregolari saranno le funzioni considerate più probabili.

Le funzioni kernel usate nei GPR sono le stesse disponibili anche per altri algoritmi di Machine Learning, come le SVM descritte precedentemente oppure le regressioni Kernel. La scelta della funzione kernel influenza notevolmente le prestazioni del modello: un kernel RBF, ad esempio, favorisce funzioni regolari e continue, mentre altri kernel possono catturare relazioni più complesse. In applicazioni avanzate è pratica comune definire nuovi kernel come funzione di kernel diversi, espandendo le possibilità di adattamento del modello a fenomeni di qualsiasi natura.

Come già accennato, i GPR hanno la caratteristica di fornire una stima dell'incertezza associata a ogni predizione, rendendoli diversi dalla maggior parte dei modelli usati in

ambito Machine Learning. Sono inoltre flessibili perché, grazie alla presenza di funzioni kernel, possono modellare relazioni non lineari complesse. Avendo inoltre un approccio di natura Bayesiana, dispongono di una base probabilistica solida e consolidata.

Le due limitazioni principali dei GPR consistono nell'elevata complessità computazionale e nell'arbitrarietà associata alla scelta della funzione kernel. Quest'ultima può influenzare significativamente la bontà del modello e richiedere un grosso sforzo nella ricerca della combinazione ottimale.

2.2 Metriche di bontà dei modelli

La valutazione della bontà di un modello di machine learning è una fase cruciale per comprendere le sue reali capacità predittive, ossia se riesce davvero a fare delle buone previsioni e se in grado di funzionare bene anche su dati nuovi, mai visti durante l'addestramento. In particolare, nei problemi di regressione, dove l'obiettivo è predire una variabile numerica continua, esistono diverse metriche che quantificano la discrepanza tra le predizioni del modello e i valori reali.

La valutazione avviene calcolando le metriche di interesse in tutte le fasi di stima e predizione del modello, dunque su tutti i set di dati a disposizione. Le metriche più comuni per la valutazione di modelli di regressione sono brevemente descritte di seguito:

- l'errore quadratico medio, noto come MSE (Mean Squared Error), calcola la media dei quadrati delle differenze tra le predizioni e i valori reali:

$$MSE = (1/n) \sum_{i=1}^n (y_i - \hat{y}_i)^2 \in [0, +\infty)$$

dove n è il numero di osservazioni nel set in uso, y_i è il valore reale osservato nel set per l' i -esima osservazione e $\hat{y}_i$ è la predizione del modello per l' i -esima osservazione. L'MSE è sensibile agli outlier, poiché gli errori vengono elevati al quadrato. Valori più bassi di MSE indicano una migliore performance del modello;

- l'errore assoluto medio, noto come MAE (Mean Absolute Error), calcola la media dei valori assoluti delle differenze tra le predizioni e i valori reali:

$$MAE = (1/n) \sum_{i=1}^n |y_i - \hat{y}_i| \in [0, +\infty)$$

Il MAE è meno sensibile agli outlier rispetto all'MSE, poiché non eleva al quadrato gli errori. Anche in questo caso, valori più bassi indicano una migliore performance;

- l'errore medio assoluto percentuale, noto come MAPE (Mean Absolute Percentage Error), calcola la media delle differenze percentuali assolute tra le predizioni e i valori reali:

$$MAPE = (1/n) \sum_{i=1}^n |y_i - \hat{y}_i| / |y_i| \cdot 100 \in [0, 100]$$

MAPE è espresso in percentuale ed è quindi facilmente interpretabile. Tuttavia, presenta delle limitazioni: non è definito quando anche solo un'osservazione $y_i = 0$ e può essere distorto da valori di y_i molto piccoli;

- il coefficiente di determinazione R^2 , noto anche come indice R^2 , misura la proporzione di varianza della variabile dipendente spiegata dal modello rispetto alla varianza totale:

$$R^2 = 1 - (SS_{res} / SS_{tot}) \in (-\infty, 1]$$

dove SS_{res} è la somma dei quadrati dei residui (RSS), SS_{tot} è la somma dei quadrati totali (TSS), che misura la variabilità totale della variabile dipendente. Il valore di R^2 varia tra 0 e 1, dove un valore pari a 1 indica un adattamento perfetto del modello ai dati e una totale assenza di errori residui, mentre un valore pari a 0 indica che il modello non spiega alcuna variabilità della variabile dipendente. Valori negativi di R^2 si verificano quando il modello in esame performa peggio di un modello *naive* definito come $\hat{y} = E[x] = \mu$, ovvero il valore medio della variabile dipendente per tutte le osservazioni. Questo può accadere, ad esempio, quando: si applica un modello non lineare a dati che presentano una relazione lineare, e il modello non riesce a catturare la linearità; si utilizzano tecniche di regolarizzazione molto forti (come valori elevati di λ nelle regressioni Ridge e Lasso) che restringono eccessivamente i coefficienti, portando a predizioni peggiori della media; il modello è stato addestrato su un dataset molto piccolo e viene testato su un dataset molto diverso, con una distribuzione differente;

- R^2 aggiustato: lavorando nel contesto dell'alta dimensionalità, è bene tenere presente che l'utilizzo di R^2 può essere fuorviante in presenza di nuove variabili non significative, poiché il suo valore tende ad aumentare anche se l'aggiunta di tali variabili non migliora realmente il modello. L'indice R^2 aggiustato corregge tale fenomeno penalizzando l'inclusione di variabili non significative:

$$R_{adj}^2 = 1 - [(n - 1) / (n - p - 1)] (SS_{res} / SS_{tot})$$

dove n è il numero di osservazioni, p è il numero di predittori nel modello. L'indice R_{adj}^2 può diminuire se si aggiungono variabili non significative al modello, fornendo una misura più accurata della bontà di adattamento. Come spiegato in seguito, tuttavia, R_{adj}^2 può assumere valori maggior di 1.

Una breve nota a margine va dedicata a due indici molto diffusi in ambito GLM, ovvero AIC (Akaike Information Criterion) e BIC (Bayesian Information Criterion). Entrambi cercano di bilanciare la bontà di adattamento ai dati compensando per la complessità del modello. Tuttavia, entrambi gli indici prevedono il ricorso alla funzione di verosimiglianza. Per la maggior parte dei modelli di Machine Learning non parametrici, come alberi decisionali, Support Vector Machine e reti neurali, non è possibile calcolare AIC e BIC perché non esiste una forma analitica della funzione di verosimiglianza. Esistono delle approssimazioni numeriche o analitiche per casi specifici come nella Random Forest e reti neurali, ma il calcolo è complesso e molto oneroso in termini computazionali, dunque non conveniente.

Infine è utile osservare come MAPE, R^2 , R_{adj}^2 , AIC e BIC siano indici adimensionali, utili sia per il confronto di modelli che presentano variabili target appartenenti a domini dimensionali differenti, oppure, cosa più comune, per fornire una misura della variabilità, o della bontà di adattamento, in presenza di variabili target con scale differenti.

3. DATI, METODOLOGIA E APPROCCIO UTILIZZATO **Dati e criticità**

Per la presente analisi, i dati sono stati estratti dal portale Infobila, un'applicazione web messa a disposizione da ANIA (Associazione Nazionale fra le Imprese Assicuratrici), operativamente distribuito tramite la piattaforma ANIASAFE.

Si tratta di un portale informativo dedicato alla raccolta e all'analisi dei dati di bilancio delle imprese di assicurazione operanti in Italia. Il database mette a disposizione informazioni dettagliate relative allo Stato Patrimoniale, al Conto Economico e ai principali indicatori tecnico-finanziari delle compagnie assicurative, articolate per impresa e per ramo di attività. Il sistema consente analisi storiche e confronti tra operatori, rappresentando uno strumento di supporto per studi economici, attività di benchmarking e valutazioni sulla solidità e sulle performance del settore assicurativo italiano. Copre l'intero mercato nazionale, includendo sia le imprese associate ad ANIA sia quelle non associate, nonché le rappresentanze di imprese extra-UE operanti in Italia. I dati sono aggiornati con cadenza annuale e permettono la ricostruzione di serie storiche di lungo periodo, a partire dalla fine degli anni Novanta.

L'accesso ai dati non è libero ma avviene previa autorizzazione. Le informazioni possono essere consultate o estratte solo attraverso specifiche richieste oppure mediante la sottoscrizione di servizi a pagamento, secondo le modalità definite dal gestore del portale.

Ai fini del lavoro, dal portale sono stati estratti i Bilanci (Stato Patrimoniale e Conto Economico per ciascuna Compagnia), i Conti Tecnici (per ogni compagnia e per ciascun ramo assicurativo è stata estratta una struttura sintetica del conto tecnico), Indici Tecnici (indicatori tecnici e di performance suddivisi per ramo e compagnia). Per le analisi è stato preso come riferimento l'anno di bilancio 2021, e le elaborazioni sono state condotte mediante il software Python.

Come già anticipato, il lavoro si è sviluppato in modo progressivo e può essere articolato in due fasi principali. Una prima fase è stata dedicata alla verifica dei calcoli del Solvency Ratio mediante l'applicazione di diverse tecniche, nonché alla definizione di intervalli predittivi.

Le osservazioni corrispondono alle imprese che trasmettono annualmente i propri dati ad ANIA nell'ambito delle indagini di mercato, mentre le variabili sono rappresentate da un insieme di indicatori tecnici ricavati dalle principali voci di bilancio e di conto economico,

estratte dalla piattaforma ANIA (dati non pubblici) e dalle informazioni contenute negli SFCR.

Di seguito si riportano le 31 variabili (indicatori) definite inizialmente al fine di catturare le informazioni necessarie per intercettare il fenomeno:

Indicatori di rischio tecnico

- **Gross Loss Ratio (a):** definito come il rapporto tra l'onere sinistri e i premi di competenza, rappresenta uno degli indicatori più consolidati per valutare la rischiosità tecnica del portafoglio. L'indicatore influenza direttamente sia i flussi di cassa in uscita per sinistri pagati sia la valutazione delle riserve sinistri, incidendo sulla Best Estimate delle passività tecniche. Nel framework Solvency II, tali dinamiche si riflettono sia sui fondi propri, attraverso il risultato tecnico, sia sullo SCR Underwriting Risk (anche "SCR UWR"), che cattura la volatilità del rischio di sottoscrizione.
- **Net Loss Ratio (b):** misura l'onere sinistri al netto della riassicurazione rispetto ai premi netti e consente di cogliere l'esposizione effettiva al rischio tecnico trattenuto dall'impresa. In linea con la letteratura sul ruolo della riassicurazione come strumento di mitigazione del rischio, questo indicatore integra le scelte di trasferimento del rischio nella valutazione della performance tecnica. Il Net Loss Ratio incide sui fondi propri attraverso il risultato tecnico netto e fornisce un'informazione chiave sulla rischiosità considerata nello SCR Underwriting Risk.
- **Gross Combined Ratio (c):** sintetizza la performance tecnica complessiva includendo sia la componente sinistri sia quella delle spese. Rappresenta un indicatore chiave di redditività e sostenibilità del modello di business assicurativo. In ottica Solvency II, il CR è strettamente connesso alla determinazione della Best Estimate delle riserve e alla capacità della compagnia di generare utili "tecnici", con effetti diretti sui fondi propri.
- **Net Combined Ratio (d):** estende l'analisi della redditività tecnica incorporando l'effetto della riassicurazione, risultando particolarmente rilevante per valutare la performance economica al netto delle politiche di trasferimento del rischio. Tale indicatore risulta più informativo rispetto alla versione lorda quando si analizzano imprese con differenti strategie riassicurative. Il Net CR incide sui fondi propri e riflette la sostenibilità del portafoglio assicurativo in termini di rischio conservato.

Indicatori di struttura dei costi

- **Management expenses / Written premiums (e):** tale indicatore misura l'incidenza delle spese di gestione sui premi emessi ed è comunemente utilizzato come proxy di efficienza operativa. Un'elevata incidenza dei costi di gestione segnala una struttura organizzativa meno efficiente, con effetti sulla redditività tecnica e, di conseguenza, sui fondi propri.
- **Acquisition expenses / Written premiums (f):** cattura il peso delle spese di acquisizione, riflettendo le caratteristiche dei canali distributivi e le strategie commerciali adottate. Tali costi possono incidere significativamente sulla performance tecnica nel breve periodo, incidendo sulla capacità di generare margini e quindi fondi propri.
- **Other administrative expenses / Written premiums (h):** tale indicatore completa l'analisi dei costi facendo riferimento alle spese di amministrazione.

Indicatori di adeguatezza delle riserve tecniche

- **Sufficiency/Insufficiency of claim provisions prior years / Earned premiums (i):** l'indicatore misura gli aggiustamenti delle riserve sinistri relativi a esercizi precedenti rispetto ai premi di competenza ed è frequentemente utilizzato per valutare la bontà del processo di riservazione. Revisioni significative delle stime pregresse incidono direttamente sui fondi propri e rappresentano una fonte di rischio tecnico rilevante nel contesto Solvency II, in quanto segnalano incertezza nella valutazione delle passività.
- **Technical provisions / Written premiums (j):** tale rapporto fornisce una misura del tasso di riservazione rispetto al volume di business e riflette la prudenza delle valutazioni. Il livello delle riserve tecniche risulta centrale nella determinazione dei fondi propri e nel profilo di rischio considerato dallo SCR di underwriting.
- **Claims reserve / Written premiums (k):** l'indicatore misura il peso delle riserve sinistri rispetto ai premi emessi ed è strettamente legato all'esposizione ai sinistri non ancora liquidati rispetto al volume premi generato. Esso influisce sia sulla valutazione delle passività sia sul capitale assorbito per il rischio tecnico.
- **Premium reserve / Written premiums (l):** il rapporto cattura l'incidenza delle riserve premi e riflette la durata delle coperture e la struttura temporale delle obbligazioni assicurative. In ottica Solvency II, tale indicatore contribuisce alla

comprensione della composizione delle riserve tecniche e della loro sensibilità al rischio di sottoscrizione.

- **Claims reserve / Technical provisions (m):** tale indicatore descrive la composizione “interna” delle riserve tecniche, distinguendo tra componenti legate a sinistri già avvenuti e premi non maturati. Una maggiore incidenza delle riserve sinistri è associata solitamente a rami long-tail e a una maggiore incertezza valutativa, rilevante per lo SCR di underwriting.
- **Premium reserve / Technical provisions (n):** l'indicatore, complementare al precedente, evidenzia la quota di riserve premi sul totale delle riserve tecniche. Tale composizione influisce sul profilo temporale delle passività e sulla loro volatilità.

Indicatori di redditività tecnica

- **Technical balance / Earned premiums (o):** questo rapporto misura il margine tecnico in relazione ai premi di competenza ed è un indicatore diretto della capacità della compagnia di creare valore attraverso l'attività di underwriting.

Indicatori di riassicurazione e rischio di controparte

- **Ceded premiums / Written premiums (p):** l'indicatore misura l'intensità della riassicurazione e riflette le scelte di trasferimento del rischio adottate dalla compagnia. Un maggiore ricorso alla riassicurazione modifica il profilo di rischio dell'impresa, riducendo l'esposizione al rischio tecnico ma aumentando la rilevanza del rischio di controparte, come esplicitamente previsto nel framework Solvency II.
- **Ceded technical provisions / Technical provisions (s):** questo rapporto cattura la quota di riserve tecniche trasferite ai riassicuratori ed è rilevante sia per la valutazione delle passività sia per la determinazione del capitale assorbito per il rischio di controparte.
- **Cash / Investments (q):** l'indicatore è costruito per catturare l'esposizione al rischio di controparte associato ai depositi. Nel framework Solvency II, il cash detenuto presso controparti bancarie è soggetto a requisiti patrimoniali specifici, rendendo questo rapporto rilevante per spiegare la componente di SCR legata al rischio di default.
- **Credit Type 2 / Total assets (t):** tale indicatore misura l'esposizione verso controparti classificate come Type 2, caratterizzate da un profilo di rischio più elevato. La normativa Solvency II attribuisce a tali esposizioni un maggiore

assorbimento di capitale, rendendo il rapporto rilevante per la spiegazione dello SCR di controparte.

Indicatori patrimoniali e di leva

- **Net assets / Total technical provisions (v)**: il rapporto fornisce una misura della copertura delle riserve tecniche tramite il patrimonio netto ed è spesso utilizzato come indicatore di solidità patrimoniale.
- **Net assets / Total liabilities (cc)**: l'indicatore misura il grado di capitalizzazione complessiva rispetto alle passività totali e riflette la leva finanziaria della compagnia.

Indicatori di asset allocation e rischio di mercato

- **Equity / Investments (r)**: questo indicatore cattura l'esposizione al rischio azionario e consente di valutare l'importanza della componente equity nel portafoglio investimenti.
- **Investments / Total assets (dd)**: il rapporto descrive il peso degli investimenti sul totale dell'attivo e fornisce indicazioni sulla struttura patrimoniale della compagnia. La composizione degli attivi risulta un determinante fondamentale sia della redditività sia del profilo di rischio complessivo.
- **Property / Investments (ff)**: l'indicatore misura l'incidenza degli investimenti immobiliari sul totale degli investimenti e consente di analizzare l'esposizione al rischio immobiliare, esplicitamente modellato nello SCR di mercato secondo Solvency II.
- **Property / Total assets (gg)**: questo rapporto amplia l'analisi dell'esposizione immobiliare all'intero bilancio e segnala il grado di concentrazione del rischio immobiliare, rilevante per la stabilità patrimoniale.

Indicatori di mix di business

- **Motor / No Motor (w)**: la variabile dicotomica identifica le imprese operanti nel ramo Auto, caratterizzato da forte competizione, elevata frequenza dei sinistri e margini tecnici ridotti. Il ramo Motor rappresenta uno dei principali driver di volatilità dei risultati tecnici, rendendo tale variabile rilevante per spiegare differenze nel profilo di rischio e nel Solvency Ratio.

- **Surety written premiums / Total written premiums (z):** l'indicatore misura la specializzazione nel ramo cauzioni, caratterizzato da una struttura di rischio specifica e da dinamiche di sinistrosità differenti rispetto ai rami tradizionali.
- **General written premiums / Total written premiums (aa):** questo rapporto cattura il peso dei rami danni non auto e fornisce una misura della diversificazione del portafoglio. La diversificazione del business risulta un fattore chiave per la stabilità dei risultati e del capitale.
- **Health written premiums / Total written premiums (bb):** l'indicatore misura l'esposizione al ramo salute, caratterizzato da specifiche dinamiche di sottoscrizione e di riservazione.

Variabili metodologiche

- **USP / No USP (hh):** la variabile dicotomica distingue le imprese che utilizzano Undertaking Specific Parameters da quelle che adottano la Formula Standard. L'uso degli USP consente una calibrazione più aderente al profilo di rischio specifico dell'impresa, rendendo gli SCR calcolati non direttamente confrontabili con quelli basati sulla Formula Standard.

Successivamente, come verrà illustrato più nel dettaglio nel paragrafo 4, l'analisi si è concentrata sullo sviluppo di un modello predittivo più performante, facendo ricorso anche a metodi ensemble al fine di migliorare l'accuratezza e la robustezza delle stime. Il dataset utilizzato in questa fase è composto da 58 osservazioni e 788 variabili esplicative. La suddivisione dei dati in un insieme di addestramento (Train) e in uno di valutazione (Test) ha portato rispettivamente a 41 e 17 osservazioni, con un rapporto tra numero di variabili e osservazioni pari a 19 per il set di addestramento e a 46 per il set di test.

L'analisi di dataset caratterizzati da un elevato numero di variabili e un ridotto numero di osservazioni, rappresenta una sfida ricorrente in ambito statistico. Questa configurazione, comunemente definita *curse of dimensionality*, o *maledizione della dimensionalità*, incide significativamente sulla stima dei modelli statistici, compromettendone l'affidabilità e la generalizzabilità. Il caso in studio è paragonabile a situazioni che si ritrovano frequentemente in campo genetico¹³, in cui sono presenti sequenze molto lunghe di DNA (le input feature)

¹³ A titolo esemplificativo si veda il caso riportato sui dati genomici al Capitolo 6 del libro "Machine Learning with Python. Theory and Implementation" (Amin Zollanvari, Springer 2023).

Si utilizza un dataset di espressione genica (GEO GSE26549), contenente circa 19 897 variabili (19894 geni + 3 variabili cliniche) e 86 pazienti, con 35 che hanno sviluppato il cancro orale. L'esempio implementa una

per poche proteine (le osservazioni). In questo documento si farà riferimento al problema usando il termine *dimensionalità*.

La presenza di numerose variabili esplicative in rapporto al numero di osservazioni espone l'analisi a una serie di rischi. In primo luogo, si incorre nel rischio di *overfitting*, ossia di costruire modelli eccessivamente complessi che si adattano eccessivamente ai dati di training, catturandone anche il rumore casuale, o rumore di fondo, perdendo così la capacità di generalizzare a nuovi dati. Al contrario, si può verificare un fenomeno di *underfitting*, in cui il modello risulta troppo semplificato per cogliere la complessità intrinseca dei dati.

Inoltre, la *multicollinearità* tra le variabili esplicative, ovvero la presenza di forti correlazioni lineari tra di esse, può rendere instabili e inaffidabili le stime dei coefficienti, complicando l'interpretazione dei risultati. Infine, uno degli aspetti rilevanti della dimensionalità è il degrado esponenziale del modello rispetto al numero di variabili, rendendo necessaria una numerosità di record sempre maggiore per ottenere stime affidabili.

Per affrontare queste problematiche, è opportuno adottare una serie di strategie. La selezione delle variabili, attraverso tecniche univariate e multivariate, consente di ridurre la dimensionalità del problema e di identificare le variabili più rilevanti per spiegare la variabile dipendente. La *regolarizzazione*, mediante tecniche come Lasso e Ridge Regression, permette di controllare la complessità del modello e prevenire l'overfitting. Le tecniche di *cross-validation* forniscono una stima più accurata dell'errore di predizione e consentono di confrontare le prestazioni di diversi modelli. Sebbene esistano tecniche di riduzione della dimensionalità, come ad esempio la Principal Component Analysis (PCA) e le relative varianti, oppure tecniche di Manifold Learning come la t-SNE e tecniche innovative come gli auto-encoder, è bene osservare che anche queste metodologie soffrono dei problemi derivanti dall'alta dimensionalità. La spiegazione risiede nel fatto che la bontà di tali algoritmi dipende in parte dalla numerosità delle osservazioni, senza contare il fatto che passare da centinaia di feature a un numero nettamente inferiore pone delle difficoltà non indifferenti in termini interpretativi.

La scelta delle metodologie più appropriate dipende dunque fortemente dalle caratteristiche specifiche del dataset, dagli obiettivi dell'analisi e dalle conoscenze a priori sul dominio del problema. In questo lavoro, verranno applicate diverse strategie per affrontare e gestire la

logistic regression con regolarizzazione L1, illustrando l'intero processo: caricamento dati, preprocessing, training e valutazione del modello.

dimensione dei dati a disposizione, con l’obiettivo di individuare modelli di stima affidabili e interpretabili.

Metodologia e risultati approccio iniziale

Il campione è stato suddiviso in un set di addestramento e in un set di test, al fine di valutare la capacità predittiva dei modelli adottati. Si fa presente che in una prima fase preliminare (definibile come preparatoria allo sviluppo del modello più performante presentato nei capitoli successivi del presente lavoro), tenendo conto di un numero di indicatori leggermente inferiori (25 piuttosto che 31) sono state applicate diverse metodologie statistiche e di machine learning, tra cui: il modello di regressione lineare, le reti neurali, due differenti modelli di albero di regressione, tecniche di ensemble quali boosting, random forest e bagging, nonché metodi di riduzione della dimensionalità e di regressione penalizzata, quali l’Analisi delle Componenti Principali (PCA), la Partial Least Squares Regression (PLSR), la regressione Ridge e la regressione Lasso.

Con riferimento al modello di regressione lineare, è stata adottata una procedura stepwise finalizzata all’eliminazione delle variabili meno rilevanti, al fine di individuare una specificazione parsimoniosa e statisticamente significativa. L’insieme delle previsioni ottenute dai diversi modelli ha consentito di calcolare il Mean Squared Error (MSE), utilizzato come criterio di valutazione comparativa per la selezione del modello con le migliori performance predittive.

Considerando i risultati ottenuti, si è proceduto effettuando l’analisi sui 31 indicatori sopra riportati con due modelli: XG Boost e Random Forest.

Nel seguito si presentano i risultati ottenuti, precisando che, dato il numero limitato di osservazioni, come già evidenziato in precedenza, la performance dei modelli è stata valutata variando la percentuale del campione di addestramento, come illustrato nella tabella seguente. Per alcuni dei modelli stimati è stata inoltre adottata la tecnica della cross-validation, unitamente alla procedura di ottimizzazione degli iperparametri tramite algoritmo di Random Search.

Figura 1: Modelli XG Boost

Identificativo modello	Descrizione
XGBoost_Vanilla_rel	80% train

XGBoost_CV_11_rel	Bootstrap – 80% train
XGBoost_CV_03_rel	60% train
XGBoost_CV_04_rel	80% train + Gradient Based sampling
XGBoost_CV_05_rel	70% train + Gradient Based sampling
XGBoost_CV_02_rel_opt	80% train
XGBoost_CV_02_rel	80% train
XGBoost_CV_08_rel	75% train
XGBoost_CV_01_rel	70% train
XGBoost_CV_09_rel	65% train
XGBoost_CV_10_rel	50% train

Figura 2:Modelli Random Forest

Identificativo modello	Descrizione
RandomForest_Vanilla_rel	80% train
RandomForest_CV_03_rel	60% train
RandomForest_CV_02_rel	80% train
RandomForest_CV_08_rel	75% train
RandomForest_CV_01_rel	70% train
RandomForest_CV_09_rel	65% train
RandomForest_CV_10_rel	50% train

Figura 3:MSE Modelli XG Boost

Identificativo modello	MSE_TRAIN	MSE_TEST
XGBoost_Vanilla_rel	0.0000	0.3961

Identificativo modello	MSE_TRAIN	MSE_TEST
XGBoost_CV_11_rel	0.0207	0.3276
XGBoost_CV_03_rel	0.1408	0.4121
XGBoost_CV_04_rel	0.1795	0.2756
XGBoost_CV_05_rel	0.1795	0.2756
XGBoost_CV_02_rel_opt	0.1809	0.2535
XGBoost_CV_02_rel	0.2006	0.2642
XGBoost_CV_08_rel	0.2060	0.4047
XGBoost_CV_01_rel	0.2078	0.4016
XGBoost_CV_09_rel	0.2187	0.3939
XGBoost_CV_10_rel	0.2552	0.4290

Figura 4: MSE Modelli Random Forest

Identificativo modello	MSE_TRAIN	MSE_TEST
RandomForest_Vanilla_rel	0.0894	0.3300
RandomForest_CV_03_rel	0.4437	0.3855
RandomForest_CV_02_rel	0.3995	0.3053
RandomForest_CV_08_rel	0.3982	0.3793
RandomForest_CV_01_rel	0.4128	0.3681
RandomForest_CV_09_rel	0.4536	0.3630
RandomForest_CV_10_rel	0.5599	0.4129

Utilizzando il Mean Squared Error (MSE) come criterio di valutazione delle prestazioni del modello, tra i due risultano più performanti quelli ottenuti con il modello XG Boost come si può osservare nelle Figure 3 e 4.

In particolare, tra tutti i modelli definiti, il modello CV_02_rel_opt dell'XG Boost si è dimostrato il migliore nella fase di test, mostrando la minima errore quadratico medio tra i modelli considerati.

Nelle Figure 5 e 6 si riporta nuovamente il valore del MSE per tutti i modelli XG Boost sia in fase di Train che di Test, mentre nella Figura 7, si riporta il confronto tra valori osservati e valori stimati tramite il modello CV 02_rel_opt per ciascuna variabile.

Figura 5:MSE Train Modelli XG Boost

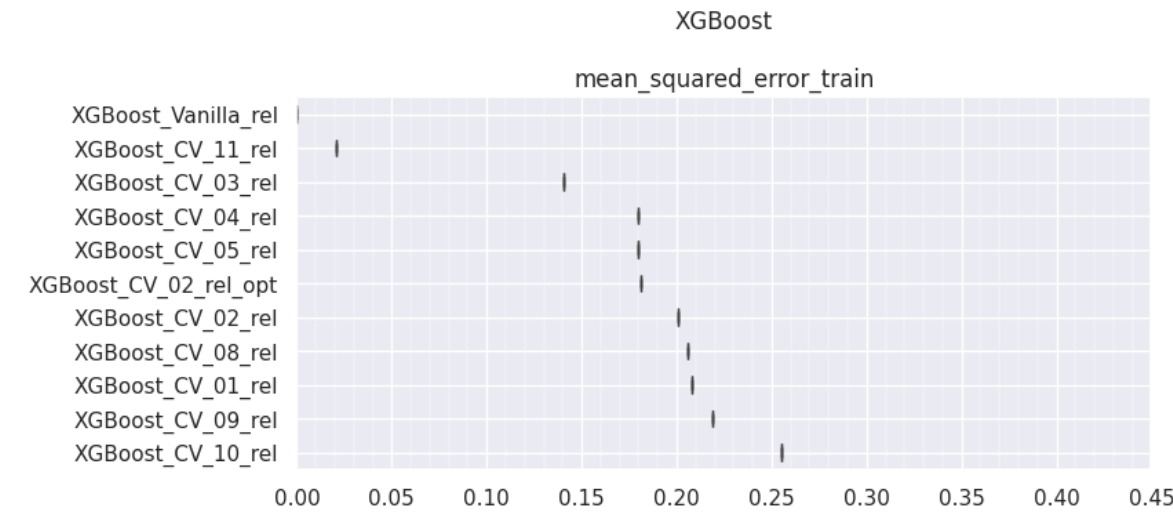


Figura 6:MSE Test Modelli XG Boost

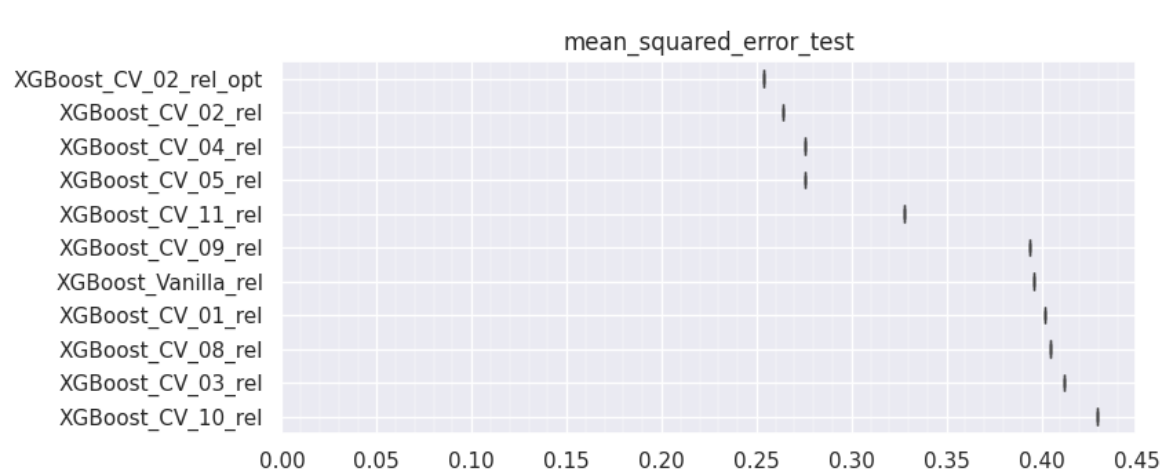
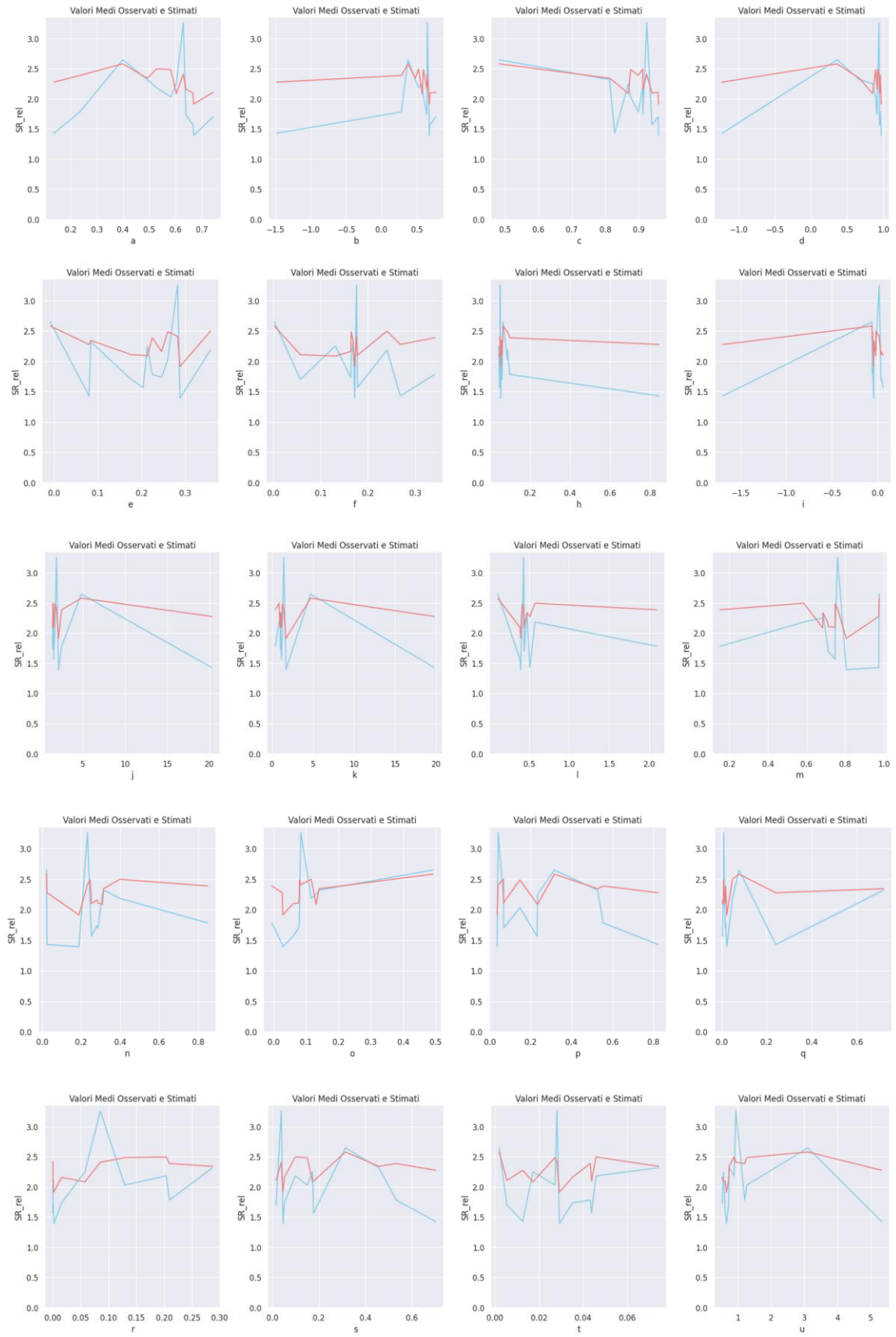
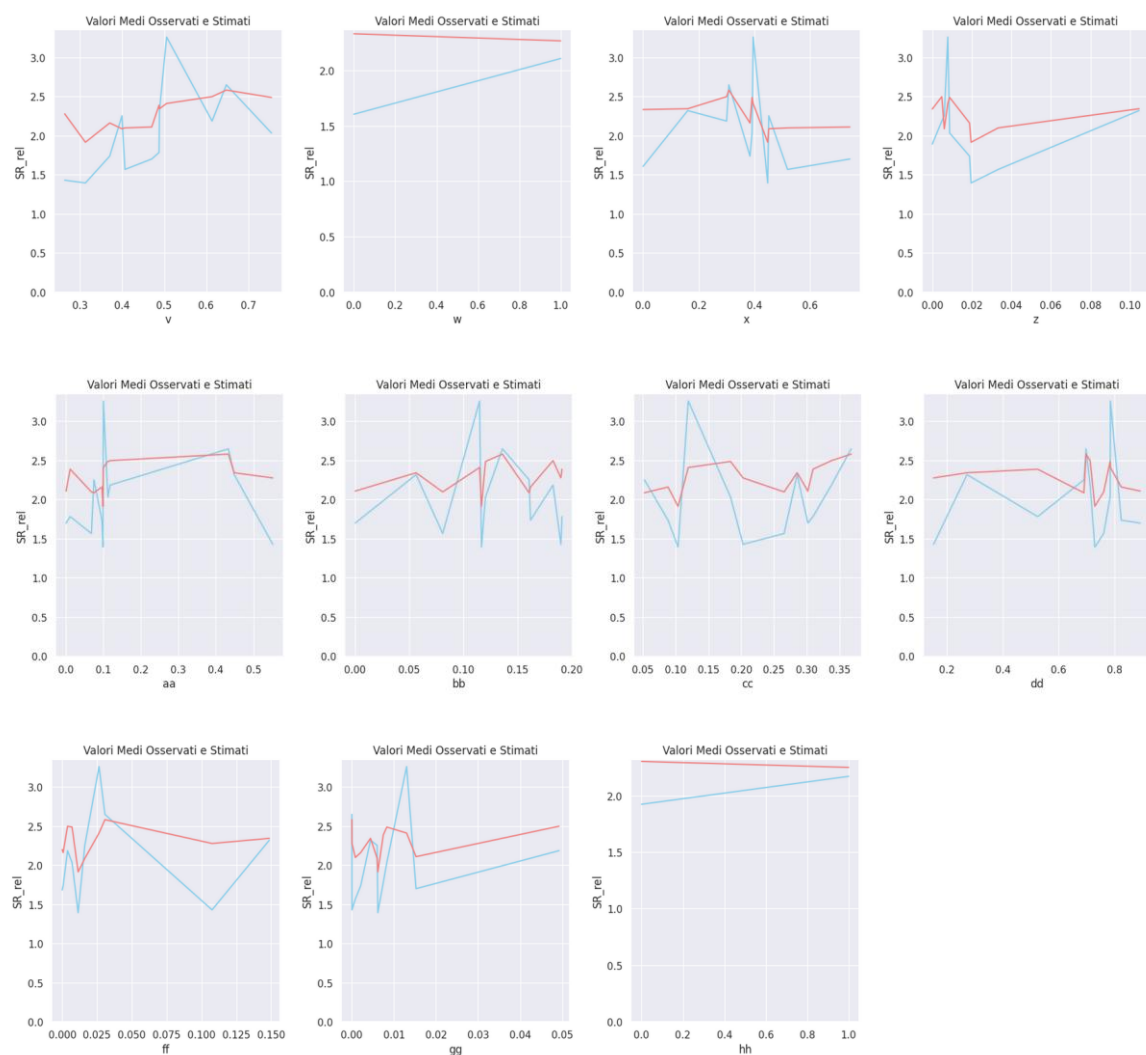


Figura 7: Valori Osservati vs Stimati - Modello XGBoost_CV_02_rel_opt





Successivamente, sulla base delle performance superiori di CV 02_rel_opt, è stata effettuata un'analisi dell'importanza delle variabili incluse nel modello mediante il metodo SHAP (SHapley Additive exPlanations). Tale approccio consente di quantificare il contributo marginale di ciascuna variabile alla predizione, fornendo un'interpretabilità locale e globale del modello, caratteristiche fondamentali nell'analisi dei modelli di Machine Learning complessi.

Nella prima visualizzazione grafica (Figura 8) sono riportate le variabili del modello completo, ordinate in base al loro impatto sulle predizioni, mentre nei grafici successivi (Figura 9 e 10) è presentato il modello ridotto, ottenuto rimuovendo alcune variabili secondo un criterio discrezionale. Tale riduzione permette di valutare l'effetto della semplificazione del modello sulla qualità predittiva e sull'interpretabilità.

Figura 8: Future Importance Modello XGBoost_CV_02_rel_opt

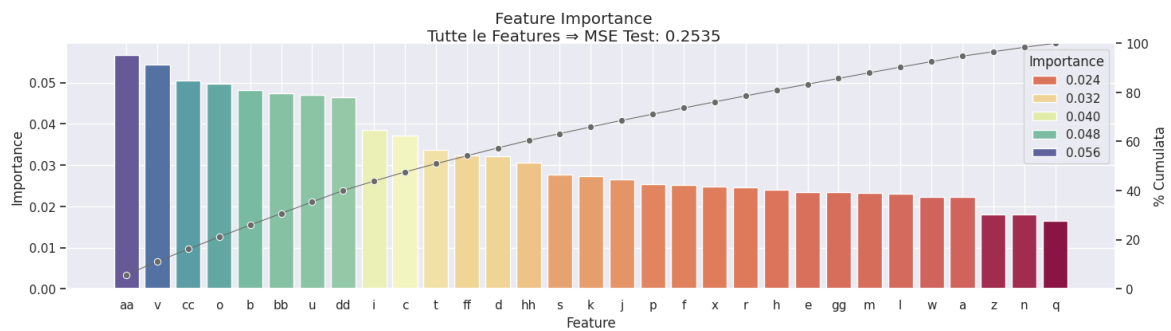


Figura 9: Future Importance Modello XGBoost_CV_02_rel_opt – rimosse Features q,n,z

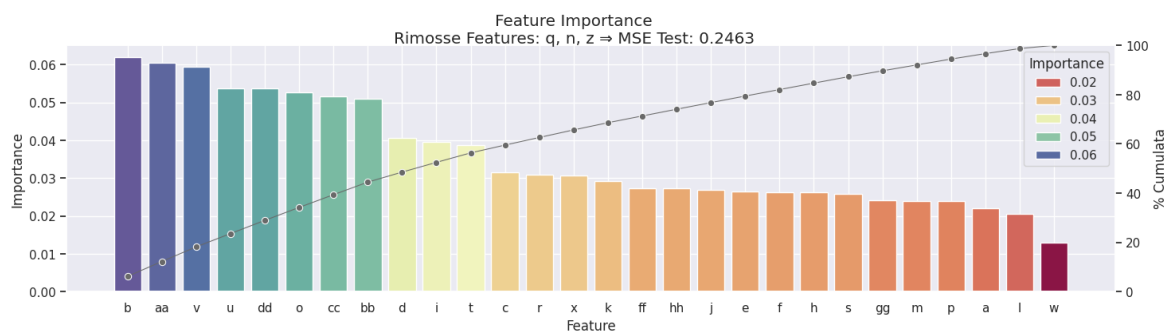
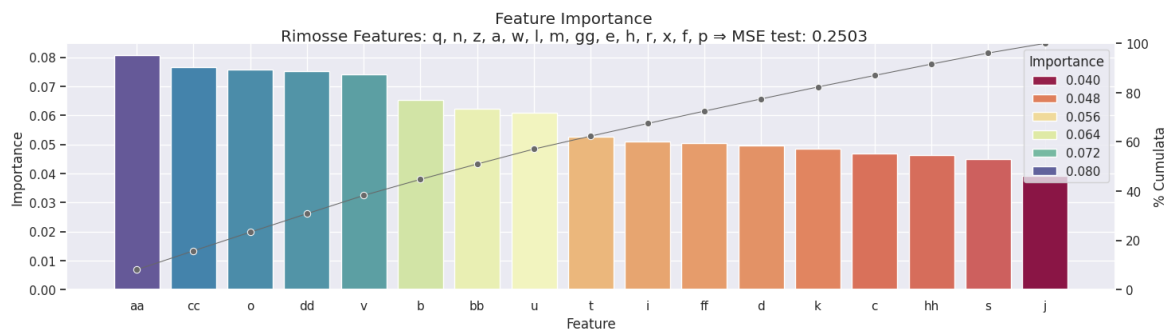


Figura 10: Future Importance Modello XGBoost_CV_02_rel_opt – rimosse Features q,n,z,a,w,l,m,gg,e,h,r,x,f,p



Come evidenziato nella Figura 10, il modello CV 02, anche nella versione ridotta, mantiene il valore di MSE più basso, confermando la sua robustezza predittiva e la consistenza del contributo delle variabili selezionate. Questo risultato sottolinea l'efficacia del modello nell'ottimizzazione della predizione e nella gestione della complessità del dataset.

3.2.1 Intervalli predittivi

Come già anticipato il lavoro non si limita alla sola stima puntuale del Solvency Ratio, ma consiste nello sviluppo di uno strumento di screening in grado di supportare i revisori, i risk

manager e l'autorità di vigilanza nell'identificazione delle imprese il cui Solvency Ratio si discosta in modo significativo dalle previsioni del modello proposto.

I modelli proposti mirano a superare le “rigidità” insite nel framework Solvency II, consentendo una selezione oggettiva delle variabili esplicative e l'individuazione di effetti di interazione in grado di migliorare la capacità predittiva, mantenendo un adeguato livello di interpretabilità, nonché la definizione di intervalli predittivi.

La costruzione di intervalli predittivi permettono di effettuare un confronto e definire un criterio per l'individuazione delle compagnie o le componenti del SCR richiedono con maggiore urgenza una revisione. Le autorità di vigilanza, ad esempio, possono avvalersi di tali intervalli per supportare la pianificazione delle attività ispettive, individuando le imprese il cui Solvency Ratio osservato si colloca al di fuori di soglie coerenti con il profilo di rischio stimato e pubblicato dalla compagnia.

La scelta del livello di significatività (α) presenta un inevitabile grado di discrezionalità e può essere calibrata tenendo conto delle risorse disponibili e dei tempi necessari per l'attività di supervisione. Analogamente, gli intervalli predittivi risultano particolarmente utili anche per i revisori, chiamati a svolgere annualmente una *limited review* del Solvency Capital Requirement (SCR).

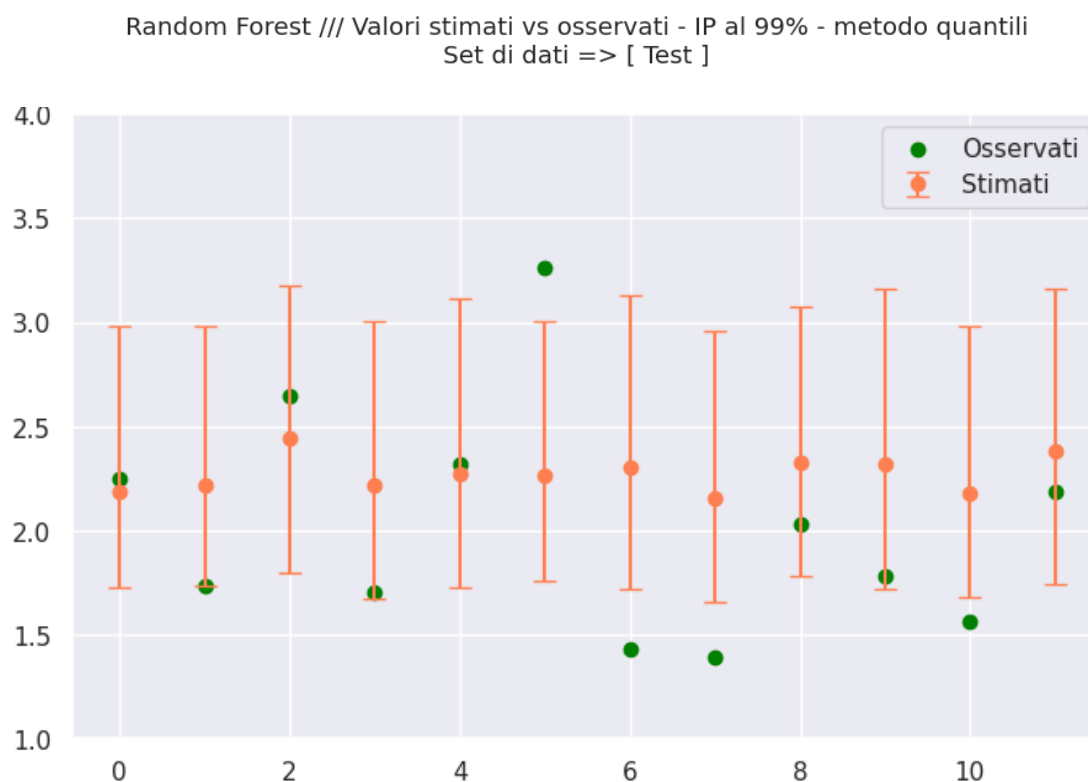
Le Random Forest, grazie alla loro struttura ensemble, forniscono una base naturale per affrontare questo problema. Oltre alla previsione puntuale del Solvency Ratio, il modello consente infatti di ricostruire una distribuzione empirica delle previsioni, derivante dall'insieme degli alberi che compongono la foresta. Tale caratteristica può essere sfruttata per stimare intervalli predittivi mediante un approccio quantilico.

Un ulteriore vantaggio di tale approccio risiede nella sua natura non parametrica, che lo rende particolarmente adatto a un contesto caratterizzato da relazioni non lineari, interazioni complesse tra le variabili e potenziali fenomeni di eteroschedasticità.

Possiamo aggiungere che l'utilizzo degli intervalli predittivi ottenuti mediante Random Forest potrebbero essere utilizzati come uno strumento operativo di early warning, capace di segnalare tempestivamente le imprese che si discostano in modo significativo dalle previsioni.

Di seguito si riporta un grafico in cui vengono confrontati i ratio di solvibilità stimati e pubblicati dalle compagnie con l'intervallo predittivo e la previsione effettuata dal modello Random Forest adottato.

Figura 11: Intervallo predittivo – Random Forest



Dal grafico emerge che in una fase di pianificazione di attività di audit, il modello ci suggerisce di dare priorità ad un controllo del calcolo del Solvency Ratio per le compagnie 5, 6, 7 e 10 in quanto, dato l'effort, le risorse e il tempo a disposizione gestiti tramite l'alpha fissata al 99%, risultano quelle al di fuori dell'intervallo.

Come si può osservare, il modello restituisce, seppure con un livello di significatività molto elevato, un numero considerevole di compagnie fuori dall'intervallo previsto. Questo risultato ha reso necessario un ulteriore approfondimento, sia in termini di variabili esplicative utilizzate sia di modello, con l'obiettivo di migliorare le capacità predittive, obiettivo che sarà sviluppato nel capitolo successivo.

4. APPROFONDIMENTO DEL MODELLO PREDITTIVO

Come già anticipato, successivamente alla fase iniziale l'analisi si è focalizzata sullo sviluppo di un modello predittivo più performante, facendo ricorso anche a metodi ensemble al fine di incrementarne l'accuratezza e la robustezza del modello. A tal proposito, nei paragrafi che seguono vengono descritte le diverse fasi che hanno condotto allo sviluppo del lavoro, articolate secondo il flusso canonico tipico delle applicazioni di machine learning.

L'obiettivo principale dell'analisi consiste nella stima dello SCR (Solvency Capital Requirement), che sarà condotta considerando separatamente cinque moduli distinti: SCR Non-Life Risk, SCR Life Risk, SCR Health Risk, SCR Market Risk e SCR Counterparty Default Risk. Al fine di rendere il processo di calcolo più agevole e garantire una maggiore coerenza metodologica, si è scelto di procedere con l'aggregazione dei singoli moduli ricorrendo alla matrice di correlazione¹⁴ prevista dalla Formula Standard. Si precisa che, come già anticipato nel paragrafo 3, il dataset utilizzato in questa fase è composto da 58 osservazioni e 788 variabili esplicative.

Pertanto, in tale scenario, viene applicata la tecnica ANOVA per la selezione delle variabili esplicative (input features). Per ciascun modulo, il metodo calcola la quota di varianza spiegata rispetto alla varianza totale, associando a questa misura un livello di significatività. All'interno del lavoro svolto, questa metodologia ha il solo scopo di quantificare indicativamente le input feature potenzialmente esplicative, ovvero le input feature che sono verosimilmente responsabili della variabilità di ciascun modulo.

Successivamente viene riportata l'analisi della bontà del campionamento casuale, tecnica che in Machine-Learning si applica durante la fase di separazione (*split*) del dataset originale nei set di Training (addestramento del modello) e Test (valutazione del modello). Il campionamento casuale è stato confrontato con il campionamento stratificato attraverso simulazioni in cui sono stati fatti variare il seme casuale, la percentuale di split e le soglie di stratificazione. Le distribuzioni del set di Train e Test sono state messe a confronto attraverso i test di Kolmogorov-Smirnov, t-test, test di Wilcoxon-Mann-Whitney e la distanza di Jensen-Shannon.

Nella fase successiva, per ciascun modulo del SCR sono stati stimati svariati modelli di Machine-Learning applicando i seguenti algoritmi:

¹⁴ Regolamento delegato (UE) 2015/35 della Commissione del 10 ottobre 2014 che integra la direttiva 2009/138/CE del Parlamento europeo e del Consiglio per quanto riguarda le norme attuative della direttiva Solvency II, pubblicato in G.U.U.E. L 12 del 17 gennaio 2015.

- Generalized Linear Models (GLM) per la definizione di un gruppo di controllo
- GLM con regolarizzazione (Lasso, ElasticNet, Ridge)
- XGBoost
- Support Vector Machine (Lineare, Kernel e Nu)
- Bayesian Ridge Regression
- Reti neurali Multi Layer Perceptron (MLP)
- Gaussian Process Regressor (GPR)
- K-Nearest Neighbor lineare (KNN)
- Hist Gradient Boosting (HGB)
- Random Forest.

La stima di tali modelli è stata condotta applicando la tecnica del campionamento stratificato su base percentile, per poi passare a tecniche di *cross-validation* e ottimizzazione degli iperparametri. Per quest'ultimo passaggio, è stata applicato l'algoritmo Bayesian Search (si veda par. 4.3.2).

Come già anticipato, per aggregare i singoli moduli del SCR è stata applicata la *formula standard* alle stime ottenute per i singoli moduli. Uno degli aspetti più importanti della metodologia qui descritta è il criterio di selezione del miglior modello per ciascun modulo. Ciò presuppone che tale scelta venga effettuata conoscendo l'effetto congiunto dei moduli sul valore finale del SCR, anziché sulla scelta del miglior modello di ciascun modulo. Sono stati quindi applicati diversi criteri di selezione, ovvero di definizione degli *ensemble* di modelli:

1. ensemble di modelli GLM, così da definire un gruppo di controllo;
2. ensemble dei migliori modelli di ciascun modulo (inclusi i GLM);
3. ensemble dei migliori modelli di ciascun modulo (esclusi i GLM);
4. ensemble definito come la combinazione di modelli (inclusi i GLM) tale da massimizzare la bontà delle stime SCR tramite opportuna tecnica di *ensemble-learning*.

4.1.Feature Selection

La selezione delle feature è una fase cruciale nell'analisi dei dati, in particolare quando si ha a che fare con dataset ad alta dimensionalità, come in questo lavoro. L'obiettivo è quello di identificare le variabili esplicative più rilevanti per spiegare la variabilità della variabile dipendente, migliorando così le prestazioni dei modelli predittivi e la loro interpretabilità. Questo processo può essere applicato: prima di stimare il modello, come nel caso del test- F

illustrato nel prossimo paragrafo; durante la stima del modello, come avviene nella regressione Lasso, Ridge e in molti modelli di Machine-Learning ad albero; in forma ibrida, confrontando i risultati sequenziali di modelli stimati con features differenti, come negli algoritmi stepwise. Tuttavia è anche possibile mescolare queste tecniche, dando origine a pipeline più o meno complesse.

test- F

Il test- F offre un metodo statisticamente rigoroso per valutare l'importanza di ciascuna feature e selezionare quelle più rilevanti. Questa metodologia è originariamente concepita per la regressione lineare. Tuttavia, la sua logica può essere estesa ad altri modelli di machine learning, in quanto molti di essi si basano su relazioni lineari tra le features e la variabile target.

Nello specifico, il test- F permette di valutare l'importanza di ciascuna feature indipendentemente dalle altre. Ciò è particolarmente utile quando si ha a che fare con un gran numero di features potenzialmente correlate. Selezionando solo le features più significative, si riduce la dimensionalità del dataset, semplificando il modello e migliorandone le prestazioni, e diminuisce anche il rischio di overfitting.

L'algoritmo prevede i seguenti passaggi:

1. calcolo della statistica F per ciascuna feature: si costruisce un modello di regressione lineare includendo tutte le features e si calcola la statistica F per ciascuna di esse;
2. calcolo del p-value: ad ogni statistica F è associato un p-value, utile per determinarne la significatività;
3. definizione di una soglia: si sceglie una soglia per il p-value (tipicamente 0,05);
4. selezione delle features: si selezionano le features con un p-value inferiore alla soglia, in quanto si ritiene che contribuiscano in modo significativo alla spiegazione della variabilità della variabile target.

Il test- F permette di comprendere quali sono i fattori più importanti che influenzano la variabile target. Si tratta di un test relativamente semplice e veloce da calcolare, dunque computazionalmente efficiente, e può essere adottato per tecniche più sofisticate di feature selection, come Lasso o Ridge.

Sono tuttavia presenti delle limitazioni alla bontà di questo metodo. Viene assunta una relazione lineare tra le features e la variabile target, mentre la presenza di multicollinearità,

ovvero di forte correlazione tra features, può influenzare i risultati. Il test- F inoltre non è in grado di rilevare o testare le interazioni tra features, anche se questo può essere ottenuto creando opportune trasformate delle feature, laddove possibile, e applicandovi il test- F in sostituzione alle feature di partenza.

Analisi dei risultati

Si osserva una importante eterogeneità nella significatività delle features per quanto riguarda i valori centrali della distribuzione delle statistiche F osservate: le feature più significative e quelle meno significative hanno una variabilità ridotta da modulo a modulo, mentre i valori centrali risultano più eterogenei. Questo aspetto introduce un livello di incertezza e distorsione rilevante nella fase di stima di un modello statistico.

Inoltre, come evidenziato in tabella (Figura 12), la percentuale di feature per le quali il test- F risulta statisticamente significativo cambia a seconda del modulo considerato. Si ricorda che il dataset presenta 788 features e che sono presenti 5 variabili target, ciascuna rappresentata dai moduli che compongono il SCR:

Figura 12: Test F per modulo di rischio

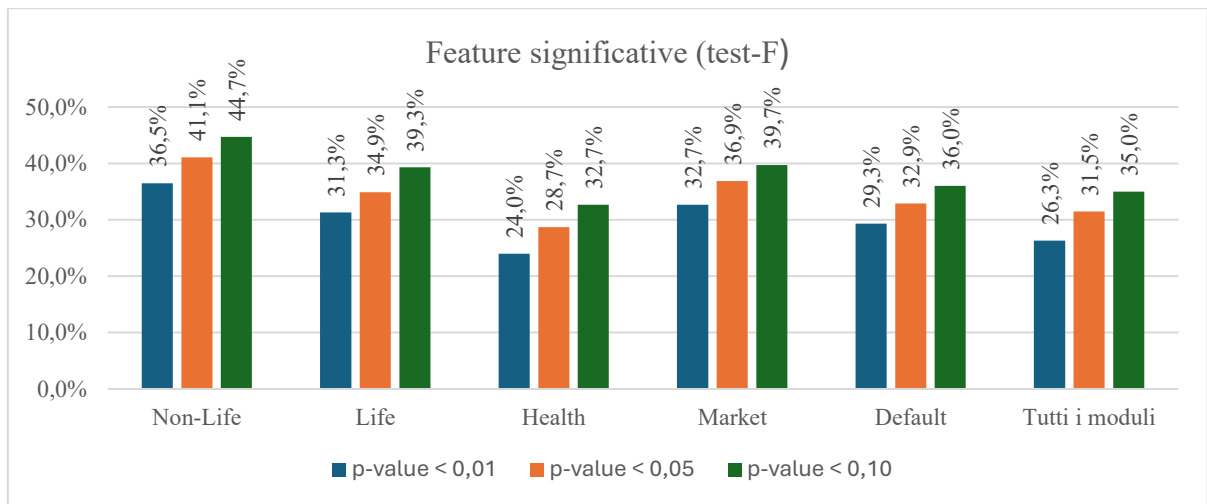
Modulo	p-value < 0,01	p-value < 0,05	p-value < 0,10
Non-Life	288 (36,5%)	324 (41,1%)	352 (44,7%)
Life	247 (31,3%)	275 (34,9%)	310 (39,3%)
Health	189 (24,0%)	226 (28,7%)	258 (32,7%)
Market	258 (32,7%)	291 (36,9%)	313 (39,7%)
Default	231 (29,3%)	259 (32,9%)	284 (36,0%)

Mediando il p-value del test- F per ciascuna feature su tutti i moduli, si osserva che 248 feature (il 31,5% del totale) ha una significatività *media* < 0,05. Di seguito i risultati completi:

Figura 13: Test F per tutti i moduli

	p-value < 0,01	p-value < 0,05	p-value < 0,10
Tutti i moduli	207 (26,3%)	248 (31,5%)	276 (35,0%)

Figura 14: Test F



Risulta evidente come la maggior parte delle feature non siano rilevanti per ciascun modulo, ma per ogni modulo cambia la significatività delle feature, dunque non è immediata una eliminazione *a-priori* di tali feature. Anzi, ignorando le interazioni e le relazioni non lineari con i moduli, è molto probabile incappare in un errore di eccessiva approssimazione del fenomeno, con conseguente underfitting dei modelli.

Inoltre, considerando una soglia $p\text{-value} < 0,05$, si osserva ad esempio come il modulo Non-Life abbia un numero di feature ritenute significativamente rilevanti superiore di circa il 43,2% rispetto al modulo Health. Un buon modello dovrà discriminare in modo ragionevolmente efficace questo tipo di eterogeneità strutturale, tenendo conto del fatto che uno degli obiettivi del lavoro è la semplificazione delle operazioni di scomposizione del dato sorgente: per ogni modulo si vuole tenere la stessa struttura in input, sebbene alcuni gruppi di feature non siano direttamente correlati con gli altri moduli e sia presente una evidente multicollinearità.

4.2. Campionamento

In questa sezione viene affrontata la tematica del campionamento da un punto di vista del suo impatto sulla rappresentatività del set di Train e del set di Test. Ciò si è reso necessario in quanto si è osservato come piccole variazioni nella percentuale di split dei due set, o addirittura del seme casuale, introducessero differenze macroscopiche nelle stime prodotte dai modelli e nella bontà stessa dei modelli. Alla base di questo comportamento vi è la scarsità di osservazioni, la quale mina la bontà di qualsiasi tecnica di campionamento, soprattutto se le variabili target hanno distribuzioni asimmetriche, come nel caso in esame.

Si è partiti dapprima con la tecnica di campionamento usata comunemente per dataset privi di particolari vincoli di autocorrelazione (come le serie temporali) o anagrafici (presenza di

variabili di stratificazione che portano alla determinazione di sotto-campioni con distribuzioni distinte), per poi passare al campionamento stratificato per soglie percentile.

Tecniche di campionamento¹⁵

Nel campionamento casuale semplice, viene definito un seme casuale (*random seed*) per inizializzare l'algoritmo di estrazione randomizzata (senza ripetizione) delle osservazioni. La probabilità di estrarre un'osservazione è generata partendo da una variabile casuale uniforme (0, 1), dunque ogni osservazione ha la stessa probabilità di essere estratta rispetto a tutte le altre osservazioni non ancora estratte. Se da un lato questa tecnica risulta semplice e anche computazionalmente efficiente, non si adatta correttamente a dati *sbilanciati*, ovvero dati in cui la variabile target, in questo caso ciascun modulo, è distribuito diversamente all'interno di strati non ben definiti del campione. Risulta quindi necessario individuare quale sia questa struttura di stratificazione del dato, ma ciò è possibile in via deterministica solo in presenza di feature qualitative, come il genere (Maschio, Femmina), le fasce di età, le fasce di reddito, etc. In ogni caso, non è garantito che, stratificando per una di queste feature, il campionamento risulti bilanciato.

La soluzione comunemente adottata per superare questo ostacolo è il ricorso al campionamento stratificato, in cui gli strati sono definiti partendo da soglie percentili. Data la natura del percentile, ovvero il valore sotto cui si trova una determinata percentuale di osservazioni, garantisce l'equidistribuzione della variabile target all'interno dei suoi livelli. La scarsa numerosità del dataset in studio, tuttavia, ha reso non immediata la determinazione di tali percentili e ha reso necessaria l'implementazione di una metodologia che consentisse di confrontare in modo sistematico la bontà delle diverse combinazioni di opzioni disponibili.

Metodologia di analisi

Per mettere a confronto le tecniche, si è provveduto a effettuare delle simulazioni facendo variare i seguenti parametri:

- il seme casuale, usato per inizializzare e rendere riproducibile la generazione casuale dell'algoritmo di campionamento;
- la percentuale di split usata per ripartire il dataset originale nei set di Train e Test;
- il modulo, ovvero la variabile target;
- il criterio di stratificazione: media, percentili, mediana, etc.

¹⁵ Si veda "Advanced Sampling Methods"

- le soglie del criterio di stratificazione nel caso dei percentili: 3 o 4 percentili.

Ciascuna simulazione consiste dunque nel generare una sequenza di $J = 100$ semi casuali, e per ciascuno di essi campionare secondo una combinazione fissa degli altri parametri. Così facendo si perviene a una sequenza di J coppie di set Train-Test.

Per valutare la bontà dei risultati di campionamento, ovvero se la distribuzione della variabile target sia ugualmente rappresentata nella distribuzione della variabile target originale e in quella campionata nel set di Train, è necessario effettuare una serie congiunta di valutazioni:

- tramite il confronto grafico degli istogrammi della variabile target all'interno dei due set;
- confrontando alcune statistiche descrittive come media, varianza, percentili, etc.
- usando opportuni test statistici come Kolmogorov-Smirnov, t-test, test di Wilcoxon-Mann-Whithney e la distanza di Jensen-Shannon.

La scelta di non confrontare il set di Train con il set di Test è che il set di Test viene usato come validazione esterna alla cross-validation e non incide sull'addestramento del modello. Avendo pochissimi record a disposizione, si deve dare massima priorità alla rappresentatività del campione rispetto alla popolazione di partenza, ovvero il dataset originale. Il set di Test viene determinato in modo del tutto complementare.

Analisi dei risultati

La complessità intrinseca del problema analizzato prevede la produzione di centinaia di grafici e statistiche. A fini esemplificativi, sono riportati i grafici del modulo Non-Life. Sull'asse delle ascisse è riportata la percentuale di split del dataset, mentre sull'asse delle ordinate è rappresentata in blu il valore della statistica test, in rosso il p-value. Trattandosi di più osservazioni, è stato anche possibile costruire un intervallo di confidenza al 95% e una traiettoria media.

Campionamento casuale semplice

La significatività dei test è molto ampia, indicando una casualità nell'osservare campionamenti validi: il fatto che, selezionando un particolare seme casuale, si osservi un set di Train rappresentativo del dataset completo è frutto del caso.

Figura 15: Test di KS (Campionamento casuale semplice) – SCR Non Life

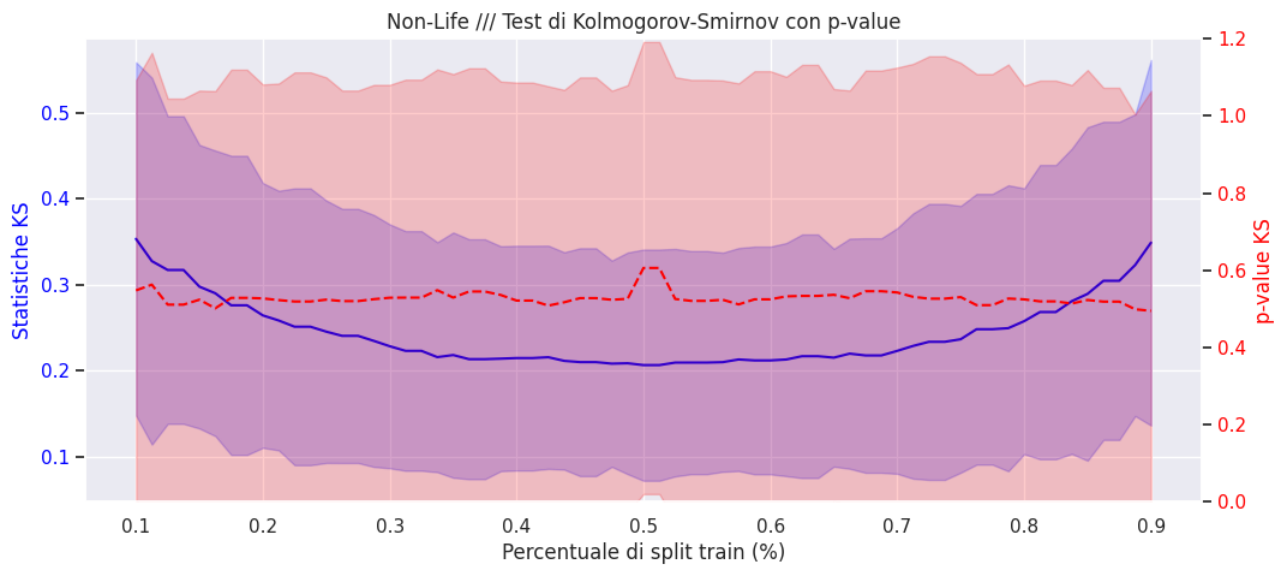
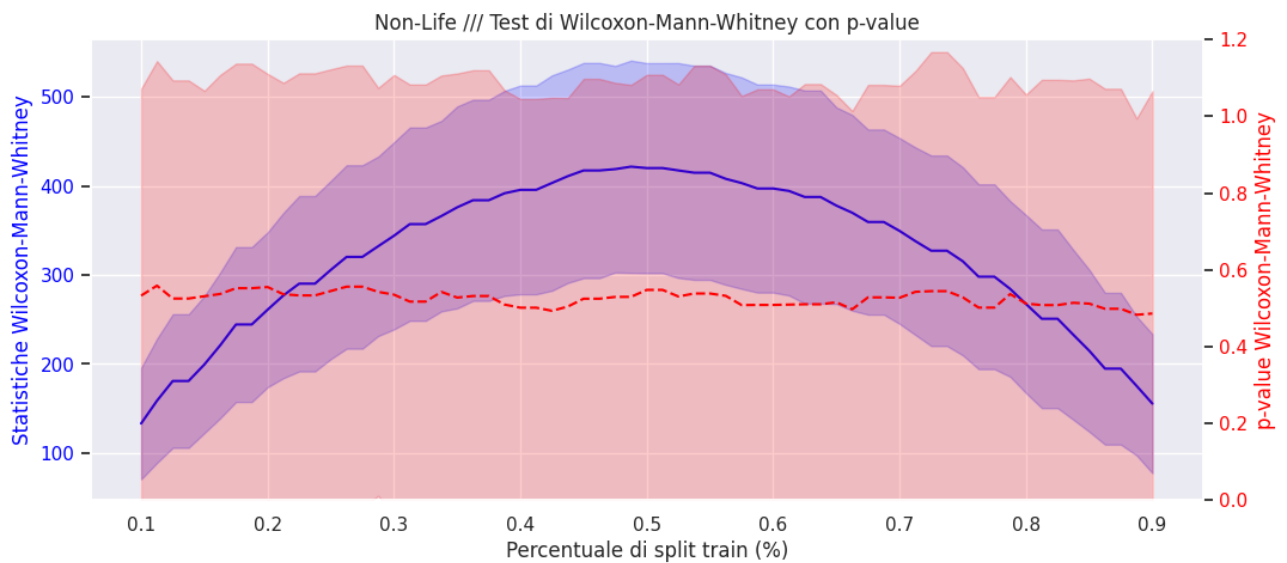
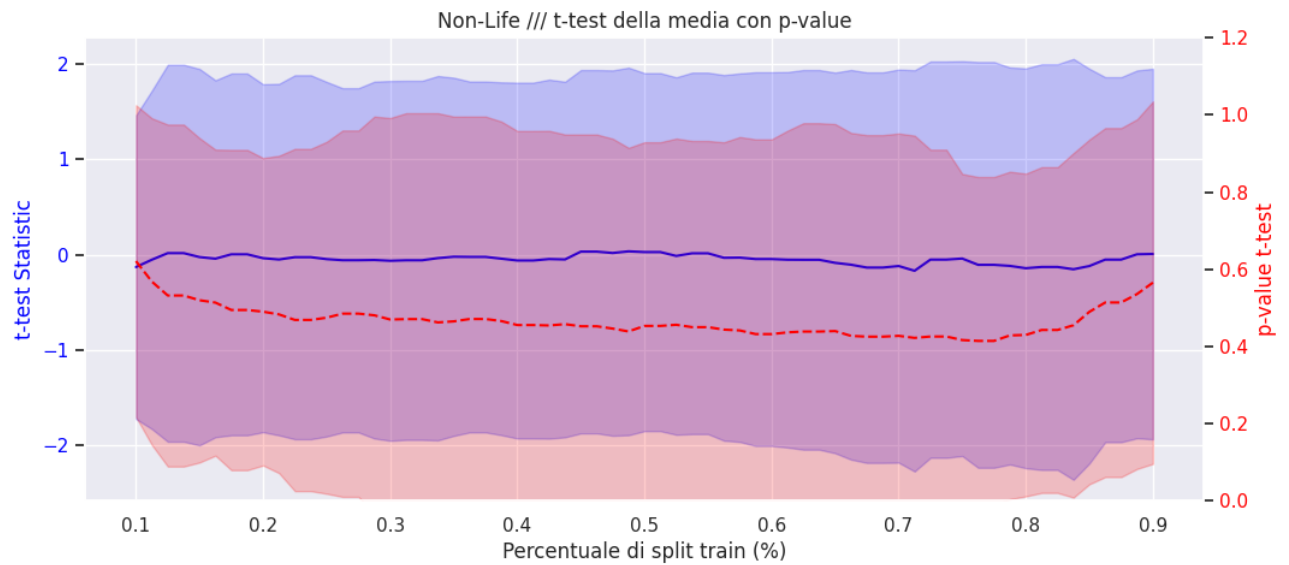


Figura 16: Test di WMW (Campionamento casuale semplice) – SCR Non Life



Inoltre, il test-t, che valuta la significatività nella differenza tra medie, è costante, al contrario delle altre statistiche. Significa che la media, di per sé, non è sufficiente a valutare la bontà del campionamento, e dunque le statistiche KS e WMW sono sicuramente da affiancare:

Figura 17: t-Test (Campionamento casuale semplice) – SCR Non Life



Campionamento casuale stratificato (percentili)

L'utilizzo del campionamento stratificato tramite il ricorso alle classi percentili, in particolare i percentili 20%, 40%, 60% e 80%, porta a risultati migliori. La significatività media, ovvero il valore medio dei p-value, è superiore rispetto al campionamento semplice per tutti e tre i test. L'intervallo di confidenza costruito attorno ai p-value è molto più stretto e sempre superiore allo zero per KS e WMW, portando all'accettazione dell'ipotesi nulla di identica distribuzione.

Figura 18: Test KS (Campionamento casuale stratificato) – SCR Non Life

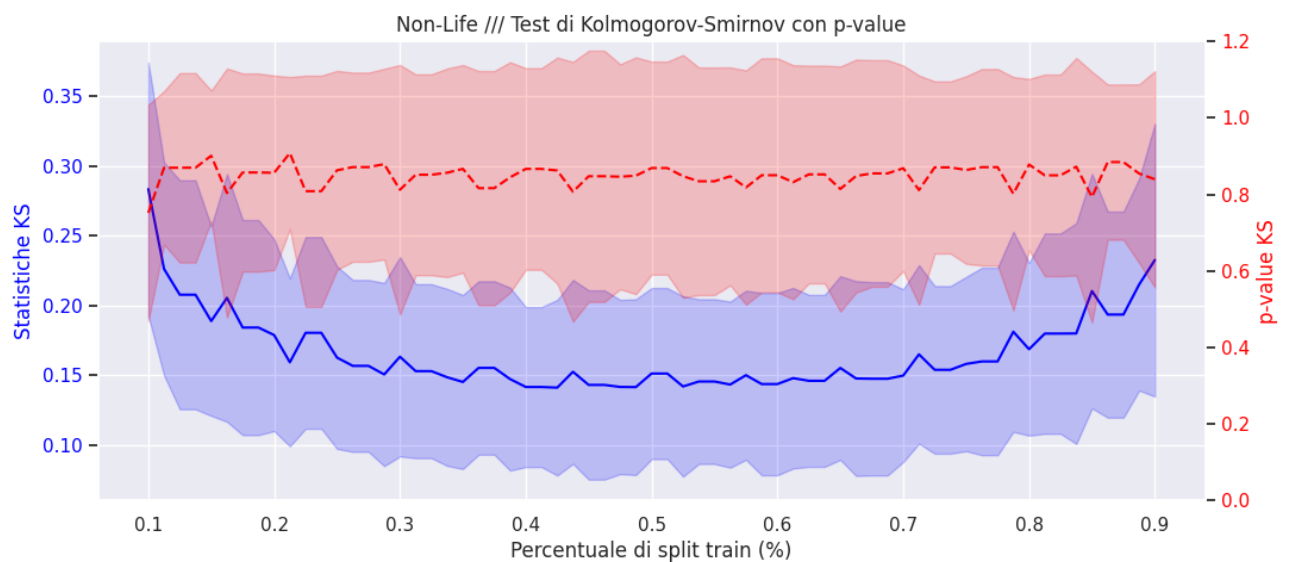
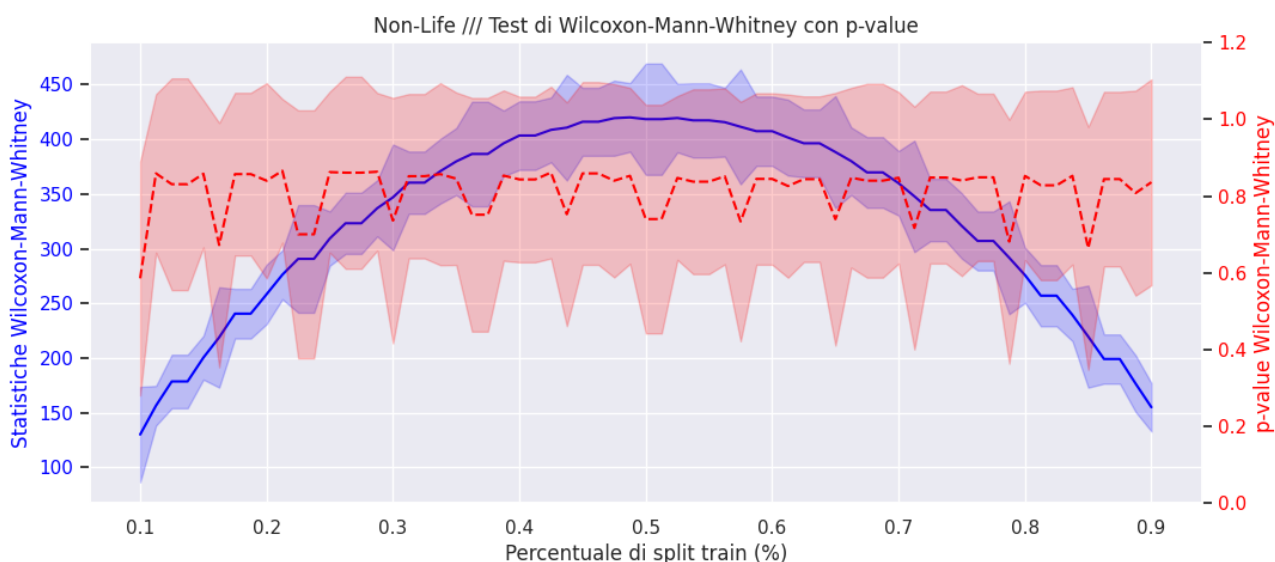
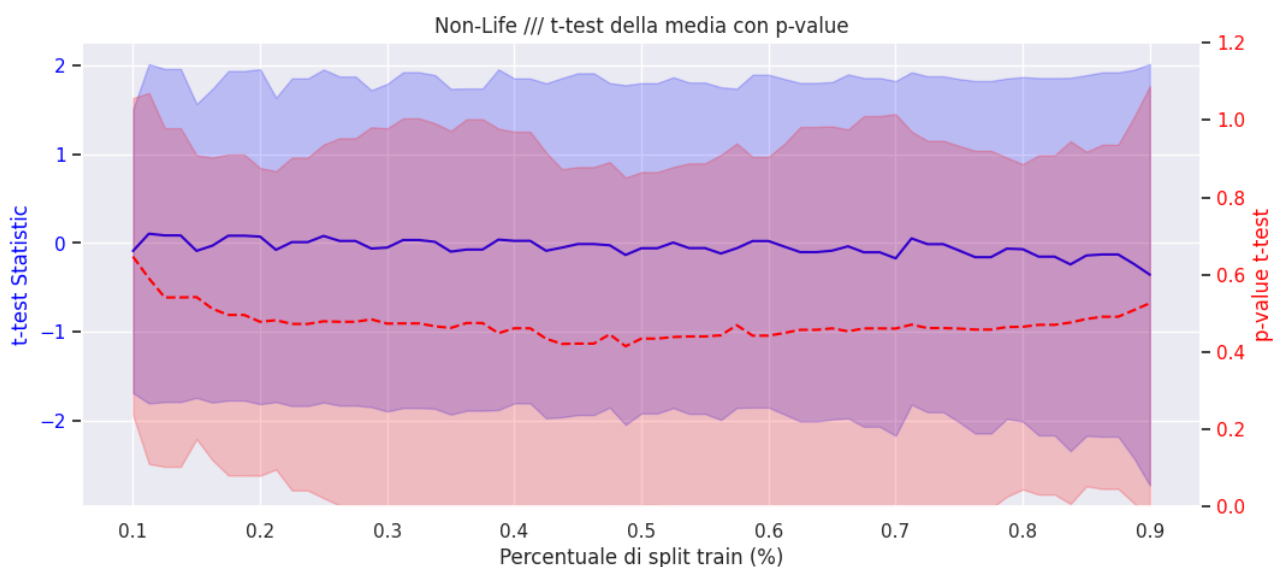


Figura 19: Test WMW (Campionamento casuale stratificato) – SCR Non Life



Il test-t fa eccezione per il motivo esposto sopra, ovvero che la media non è una metrica ideale nella valutazione della bontà di campionamento. Si osserva comunque un leggero miglioramento nell'intervallo superiore dell'intervallo di confidenza del p-value per percentuali di split nel range di interesse della modellazione statistica, ovvero 60% – 80%:

Figura 20: t-Test (Campionamento casuale stratificato) – SCR Non Life

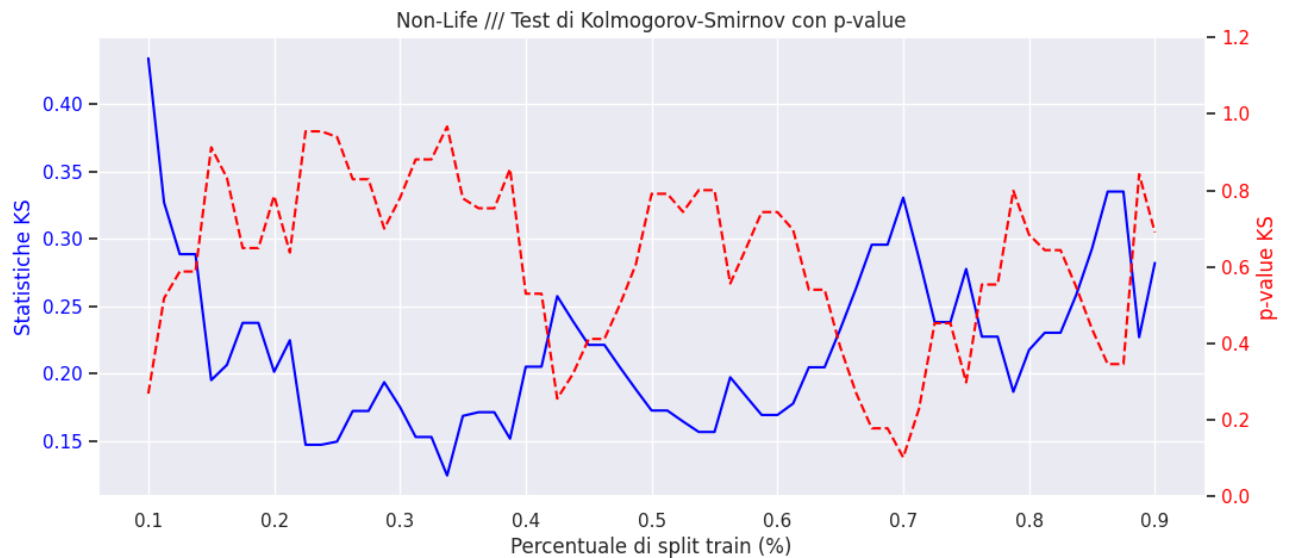


Oltre ai risultati appena illustrati, è interessante osservare come si comporta un seme casuale preso singolarmente, così da evidenziare quanto la resa del campionamento sia suscettibile a piccole variazioni nei parametri e nella tecnica adottata. La scarsa numerosità campionaria ha un impatto molto rilevante sulla bontà del campionamento, tuttavia è un aspetto che solitamente non viene misurato o riportato nelle diagnostiche statistiche.

Nei grafici seguenti sono riportate le traiettorie dei test statistici al variare della percentuale di split, avendo fissato come seme casuale il valore 42.

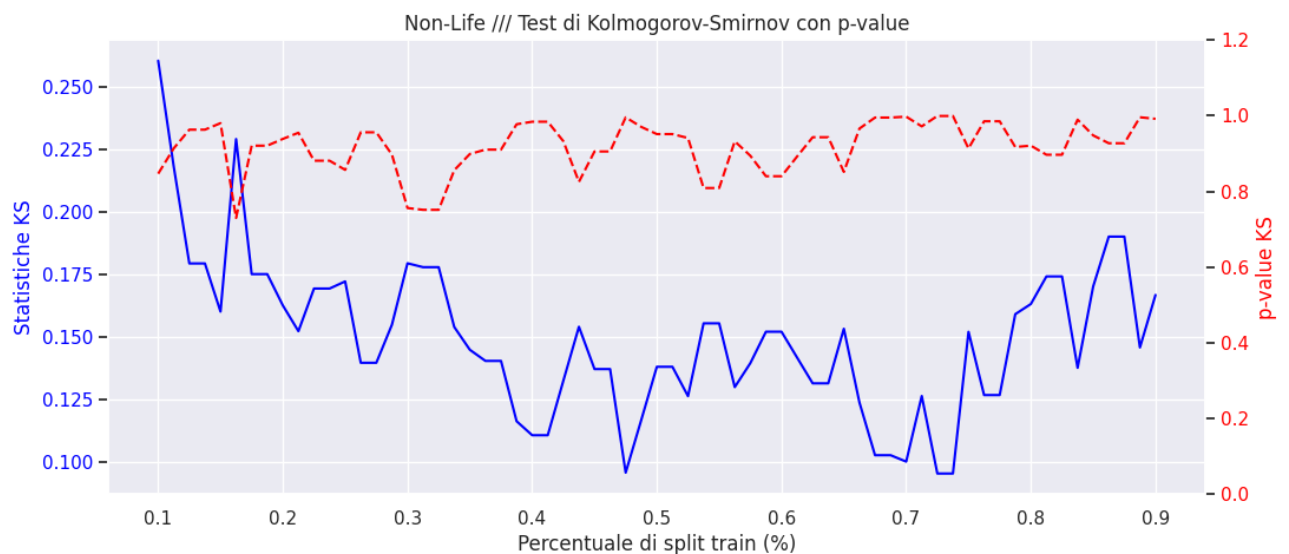
Nel campionamento casuale semplice, la percentuale di split che massimizza il p-value e minimizza la statistica KS è inferiore al 65% e superiore all'85%, mostrando un buco importante attorno al 70%. Questi valori sono di poca applicabilità e buon senso:

Figura 21: Test KS (Campionamento casuale semplice) – SCR Non Life - % split



Nel campionamento stratificato invece è evidente come il p-value e la statistica test siano omogenee e prive di particolari vuoti. Tutto il range 50%-80% è utilizzabile senza particolari controindicazioni, con un range ottimale attorno al 70%:

Figura 22: Test KS (Campionamento casuale stratificato) – SCR Non Life - % split



Queste considerazioni, che sono state effettuate anche per ciascun modulo, sono di fondamentale importanza in quanto durante la fase di stima dei modelli sarà necessario utilizzare la stessa percentuale di split per ciascun modulo. Sarà quindi opportuno individuare una soglia ottimale per tutti i 5 moduli.

Sempre a fini esemplificativi, si riportano i risultati grafici del test KS applicato a ciascun modulo per il campionamento stratificato ($J=100$) con le 4 soglie percentile:

Figura 23: Test KS (Campionamento casuale stratificato) – SCR Non Life

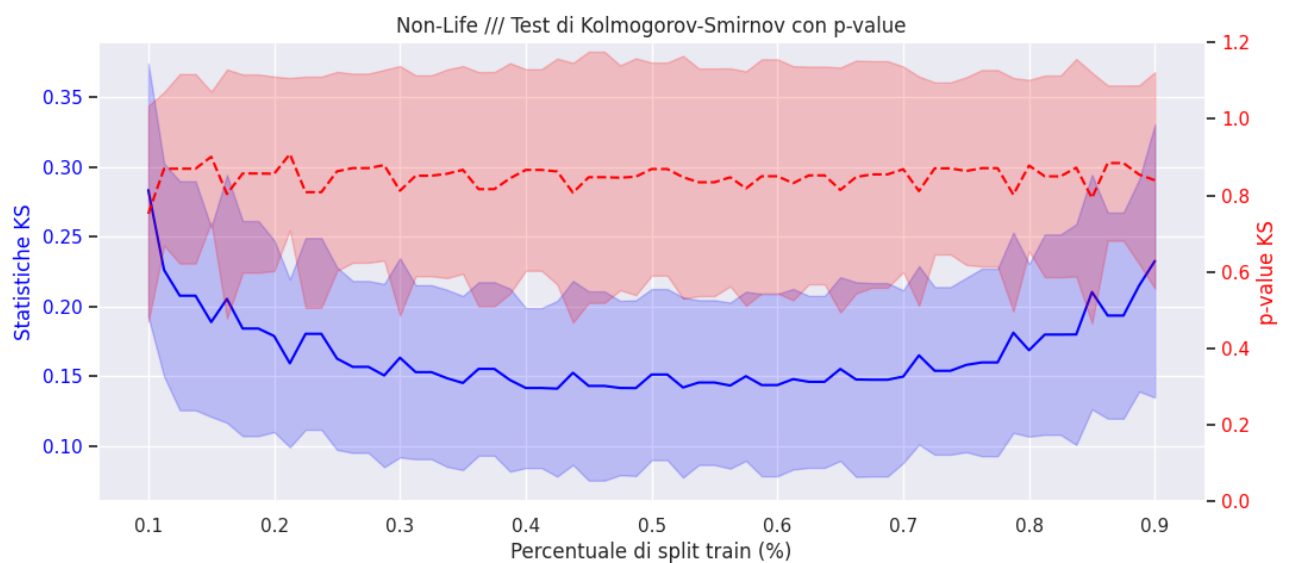


Figura 24: Test KS (Campionamento casuale stratificato) – SCR Life

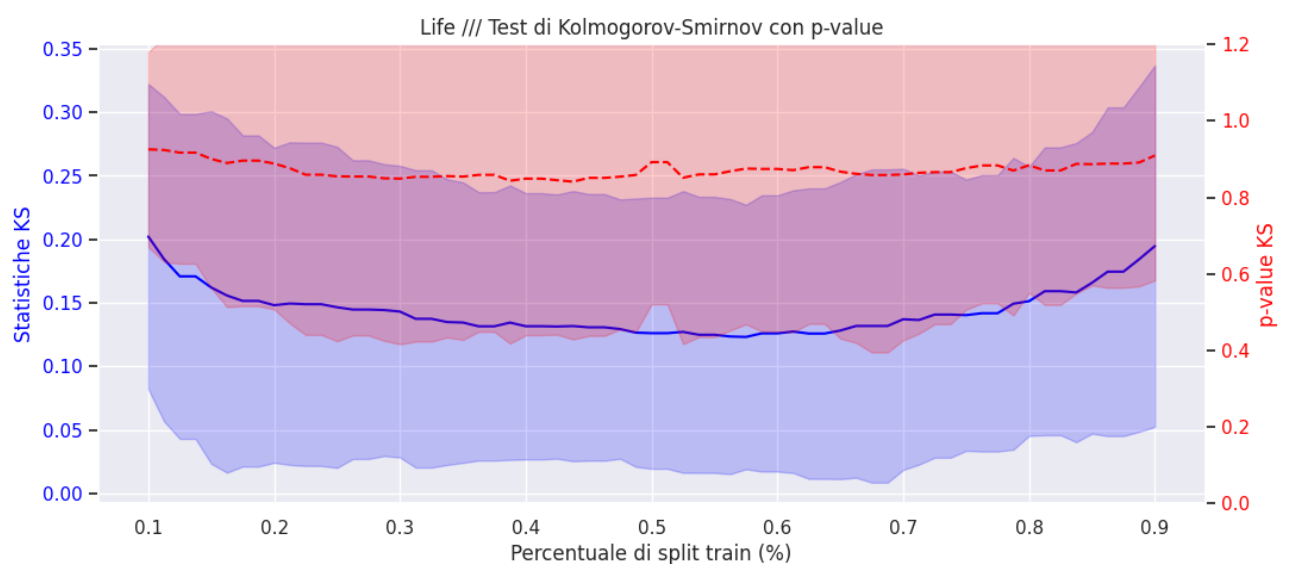


Figura 25: Test KS (Campionamento casuale stratificato) – SCR Health

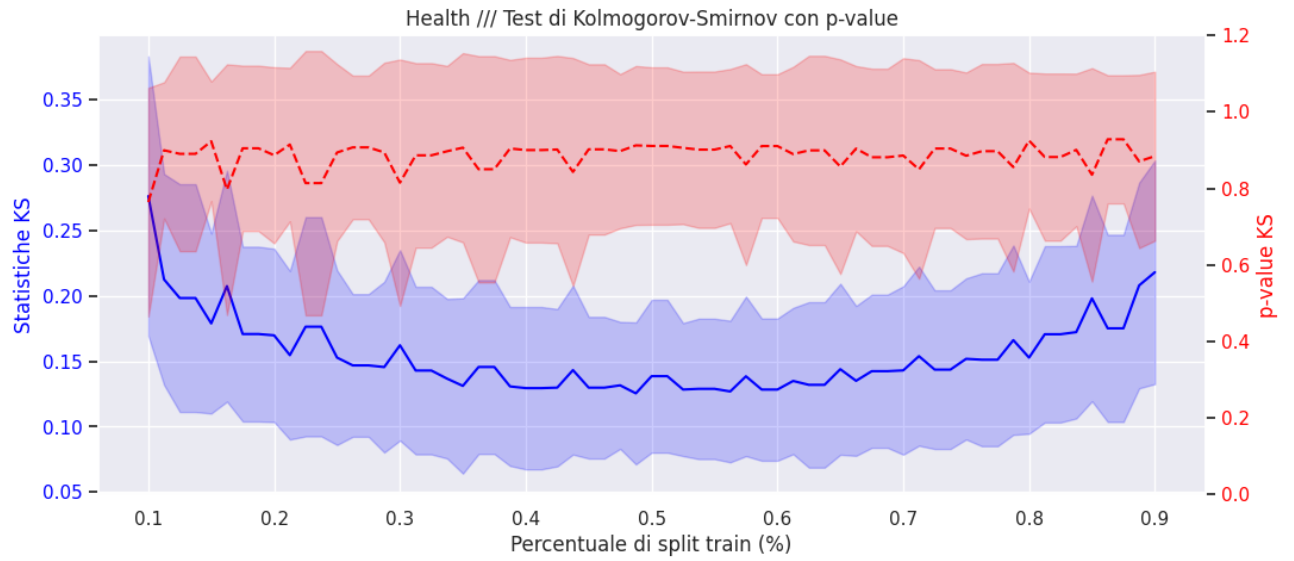


Figura 26: Test KS (Campionamento casuale stratificato) – SCR Default

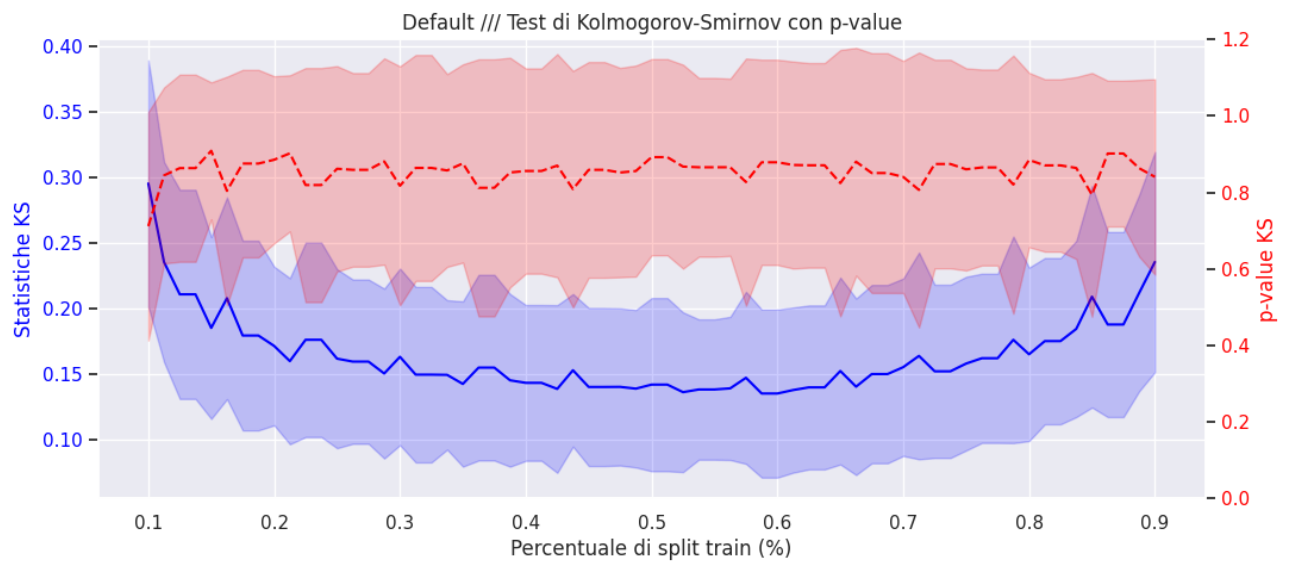
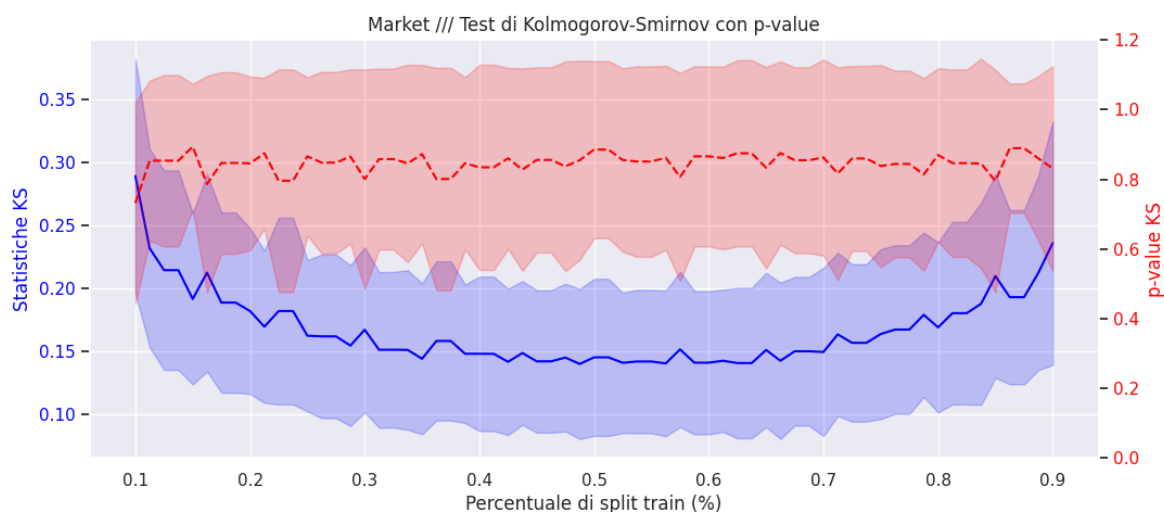


Figura 27: Test KS (Campionamento casuale stratificato) – SCR Market



Se ne conclude che Life è il modulo per il quale il test di Train è meno rappresentativo e sul quale si aspettano maggiori problematiche in termini di rappresentatività. Tutti i moduli conducono a valori delle statistiche test KS mediamente sempre significative, portando a concludere che il campionamento stratificato così definito garantisce una buona somiglianza tra la distribuzione della variabile target originale e quella campionata nel set di Train.

4.3. Approccio Machine Learning

La stima di modelli statistici tradizionali e di algoritmi più moderni e sofisticati, può essere gestita in modo assolutamente equivalente all'interno di un flusso di analisi Machine Learning, comunemente noto come *pipeline*. Il Machine Learning applica una visione ingegneristica all'analisi dei dati tramite la definizione di fasi di elaborazione sequenziali, all'interno delle quali è prevista l'applicazione di alcune tecniche specifiche.

La fase preliminare consiste nella raccolta del dato e in una sua eventuale segmentazione. Nel caso in studio, come già anticipato, si è deciso di unire in un unico dataset tutte le informazioni numeriche relative a diverse compagnie assicurative e di individuare cinque diverse variabili target, ciascuna rappresentante un singolo modulo SCR: Life, Non-Life, Market, Health, Default. Si tratta di un approccio singolare che ha il vantaggio di mantenere intatte le caratteristiche di correlazione tra feature che condizionano congiuntamente i diversi moduli. Così facendo tuttavia permane un'elevata multicollinearità e, durante la stima di ciascun modulo, la numerosità delle feature rimane pari a quella totale. Uno degli obiettivi

è proprio quello di individuare un modello e/o una metodologia che consenta di affrontare questa sfidante caratteristica strutturale del dato.

Nella fase successiva, si procede con la *feature engineering*, che consiste nel pulire, trasformare ed eliminare le feature presenti nel dataset. Nel caso in studio, il numero grezzo di feature è stato ridotto in modo da escludere tutto ciò che non fosse utile alla stima dei modelli, pur mantenendo un ampio margine di tolleranza.

Una volta che il dataset è pronto all'uso, si procede con la sua suddivisione nei set di Train e Test. In presenza di un numero significativo di record, solitamente si procede con la definizione di un terzo set, noto come set di Validation. Nel caso in studio, ciò non è stato possibile in quanto non si dispone di una numerosità sufficiente, ma è bene chiarire quale sia lo scopo di ciascun set.

Nel caso in cui vi siano solo due set, il modello viene addestrato con i dati di Train e sugli stessi dati viene effettuata la cross-validation. Il set di Test viene quindi utilizzato per valutare la bontà del modello su dati mai visti durante l'addestramento. Come spiegato più avanti, la cross-validation suddivide iterativamente il set di Train in un set di Train e Validation annidato. Questo approccio consente quindi di stimare tanti modelli quante sono le iterazioni, andando a massimizzare l'uso dei dati e al contempo mitigare un eventuale bias da selezione.

Quando invece si procede con la suddivisione dei dati in tre set, la stima del modello avviene sempre sul set di Train tramite cross-validation, per poi procedere con la validazione separata sul set di Validation. Il set di Test viene utilizzato per misurare la bontà del modello su dati mai visti prima, sapendo però che si tratta del miglior modello possibile addestrato sui dati disponibili. Il motivo per il quale questa versione risulta migliore, è che il confronto sul Test avviene dopo aver validato il modello su un'ampia gamma di varianti possibili del modello. In altri termini, usando il Test come set decisionale, ci si allontana ancora di più dal dato originale, ponendosi in un'ottica futura, immaginando quindi che il modello venga messo alla prova su dati nuovi.

Siccome la scarsa disponibilità di osservazioni a disposizione ha reso impossibile la creazione dei tre set di dati, si è reso necessario applicare e calibrare con particolare attenzione le tecniche di selezione (campionamento, già descritte nei precedenti paragrafi), stima (cross-validation) e ottimizzazione dei modelli.

Sulla base del risultato dell'analisi sulle tecniche di campionamento, si è deciso di adottare una percentuale di split del set di Train del 71,25%, portando quindi a 41 osservazioni per il set di Train e 17 per il set di Test.

Utilizzo delle metriche di bontà dei modelli

Tutte le metriche descritte al paragrafo 2.2 (MSE, MAE, MAPE, R^2 e R^2_{adj}) sono applicabili a modelli di regressione indipendentemente dal fatto che appartengano alla famiglia dei GLM o meno. Queste metriche misurano la discrepanza tra le predizioni e i valori reali, e la loro definizione non dipende dalle assunzioni specifiche dei GLM (come la distribuzione normale degli errori o la funzione link). Pertanto, possono essere utilizzate per valutare modelli di regressione lineare, alberi di regressione, SVR, reti neurali e qualsiasi altro modello che produca predizioni numeriche.

In letteratura non sono presenti regole o algoritmi che consentano di stabilire la metrica migliore per la valutazione della bontà di un modello; tuttavia, è opportuno considerare alcuni aspetti utili per una corretta interpretazione dei risultati. L'MSE è utile quando si vogliono penalizzare maggiormente errori di grande entità, mentre il MAE è più robusto agli outliers. Il MAPE, d'altro canto, è utile per interpretare gli errori in termini percentuali, o relativi, rendendo agevole il dimensionamento dell'impatto nell'errore di stima, tuttavia presenta limitazioni con valori nulli.

I coefficienti R^2 e R^2_{adj} sono utili per valutare la proporzione di varianza spiegata, fornendo una visione più statistica circa la comprensione del fenomeno da parte del modello. R^2_{adj} è preferibile quando si confrontano modelli con un numero diverso di predittori oppure quando si è in presenza di alta dimensionalità, anche se è bene fare una precisazione a questo proposito. Riprendendo la formula $R^2_{adj} = 1 - [(n - 1) / (n - p - 1)] (SS_{res} / SS_{tot})$, si notano immediatamente due elementi che possono compromettere l'utilità di questa metrica:

- quando $p \approx n$, il denominatore $(n - p - 1)$ si avvicina a zero e $[(n - 1) / (n - p - 1)]$ assume valori molto elevati, con la conseguenza che piccole variazioni in SS_{res} portano all'aumento di R^2_{adj} . Ciò comporta che R^2_{adj} può assumere valori prossimi a 1 anche se il modello non generalizza bene ai dati di validazione;
- quando $p \gg n$, come nel caso in studio, il denominatore $(n - p - 1)$ diventa negativo e il risultato finale è sempre maggiore di 1: maggiore è il valore di SS_{res} , maggiore sarà la distanza da 1. Dunque, seppur questo scenario ponga dei dubbi sulla

validità della metrica R_{adj}^2 , i valori che si ottengono esprimono ugualmente un concetto di bontà di adattamento: a parità del numero di features e di osservazioni, valori di R_{adj}^2 quasi identici a 1 indicano che il modello performa sicuramente meglio di modelli che restituiscono valori maggiori.

Per comprendere meglio i passaggi descritti più avanti, è necessario comprendere in quali fasi vengono impiegate le metriche, poiché la loro scelta è determinante nell'esito dell'intero processo. Solitamente si distingue in metriche *esterne* e metriche *interne*, riferendosi al momento del processo di training in cui viene calcolata la metrica:

1. Funzione di perdita (*loss function*): alcuni algoritmi, come XGBoost, Random Forest e le reti neurali, prevedono l'applicazione della metrica *internamente* al modello, applicata durante l'addestramento.
2. Cross-Validation: la metrica, applicata *esternamente* al modello, viene utilizzata dall'algoritmo CV per valutare la bontà del modello stimato sui $k - 1$ fold rispetto ai valori previsti sul fold rimanente (vedere paragrafo successivo per maggiori dettagli).
3. Ottimizzazione degli iperparametri: per questa fase, spiegata nel dettaglio più avanti, punta a minimizzare una funzione di perdita definita solitamente a partire da una delle metriche sopra descritte, anche se è possibile implementare metriche personalizzate. Il problema di minimizzazione può essere esteso anche a metriche multidimensionali. Si tratta dunque di metriche *esterne* alla stima stessa del modello.
4. Ensemble Learning: quest'ultimo step considera a sua volta delle metriche *esterne* al modello, in modo da individuare il meta-modello, o la combinazione di modelli migliore.

Appare evidente quanto sia importante, nonché determinante, l'utilizzo consapevole delle metriche, nonché la capacità di tenere traccia, lungo l'intero sviluppo del processo di Machine Learning, di quali metriche sono state impiegate nelle fasi di addestramento (training), validazione (validation) e valutazione (evaluation).

La cross-validation¹⁶, o validazione incrociata, rappresenta una tecnica statistica di primaria importanza nel machine learning, deputata alla valutazione delle prestazioni di un modello predittivo su dati non impiegati nella fase di addestramento. Tale metodologia consente di ottenere una stima più robusta e affidabile della capacità di generalizzazione del modello, mitigando il rischio di overfitting ai dati di training. L'obiettivo fondamentale è simulare la performance del modello su dati reali, non ancora osservati, fornendo una misura della sua effettiva capacità predittiva, pur rimanendo all'interno del set di Train.

L'approccio cardine della cross-validation consiste nella suddivisione del dataset disponibile in sottoinsiemi disgiunti, destinati alternativamente all'addestramento e alla validazione del modello. Questo processo iterativo permette di sfruttare al meglio l'informazione contenuta nei dati, impiegando ciascuna osservazione sia in fase di training che di validazione.

Tra le diverse tecniche di cross-validation, la k-fold cross-validation riveste un ruolo di particolare rilievo. Questa tecnica prevede la suddivisione del dataset in k sottoinsiemi, o "fold", di cardinalità comparabile. Il modello viene addestrato k volte, utilizzando in ciascuna iterazione k-1 fold come set di Train e il fold rimanente come set di Validation.

Le metriche di valutazione del modello, calcolate su ciascun set di Validation, vengono successivamente aggregate, tipicamente attraverso il calcolo della media, per ottenere una stima complessiva delle prestazioni. Ad esempio, nella 5-fold cross-validation, il dataset viene suddiviso in cinque parti, e il modello viene addestrato cinque volte, utilizzando ogni volta una parte diversa come validation set e le restanti quattro come training set. Questo approccio garantisce che ogni osservazione venga utilizzata una volta per la validazione.

Una variante di questa tecnica, particolarmente utile in presenza di dataset sbilanciati, è la stratified k-fold cross-validation. In questo caso, la suddivisione in fold viene effettuata in modo da preservare la proporzione delle diverse classi presenti nel dataset originale all'interno di ciascun fold. Tale stratificazione previene la formazione di fold con una rappresentazione non uniforme delle classi, che potrebbe condurre a valutazioni distorte delle prestazioni del modello.

Un caso limite della k-fold cross-validation è rappresentato dalla Leave-One-Out Cross-Validation (LOOCV), in cui k coincide con il numero di osservazioni nel dataset. In ogni iterazione, una singola osservazione viene utilizzata come validation set, mentre le restanti $n - 1$ osservazioni costituiscono il training set. La LOOCV sfrutta al massimo i dati

¹⁶ Si veda "An Introduction to Statistical Learning with Applications in Python".

disponibili per l'addestramento, ma risulta computazionalmente onerosa per dataset di grandi dimensioni e può presentare un'elevata varianza, specialmente in presenza di outlier.

La repeated k-fold cross-validation prevede la ripetizione della k-fold cross-validation per diverse suddivisioni casuali dei dati in fold, riducendo ulteriormente la variabilità della stima delle prestazioni del modello.

Nel caso in studio, sono state testate le tecniche appena descritte in modo da individuare la versione più adatta a mitigare problemi di sbilanciamento del dato. Si è comunque visto che l'impatto più significativo si ottiene intervenendo sulla tecnica di campionamento piuttosto che sulla tecnica di cross-validation (il campionamento viene effettuato durante la cross-validation, ma viene definito in modo indipendente). Per quanto riguarda l'algoritmo di cross-validation, ciò che ha inciso maggiormente è stata la selezione del numero di fold, ovvero del numero k. Le tecniche LOOCV e stratified k-fold hanno portato a risultati leggermente migliori, a scapito però di un aumento esponenziale dei tempi di elaborazione. Si è quindi deciso di optare per la k-fold con 3 fold, ottenendo quindi tre modelli stimati per ogni algoritmo testato, in cui i set annidati di test hanno dimensione $2/3$ ($n=27$) e i set di Validation hanno dimensione $1/3$ ($n=14$).

Come spiegato nel paragrafo successivo, la cross-validation viene ripetuta per ogni combinazione di iperparametri testati. Il numero totale di modelli stimati risulta quindi pari al numero di fold per il numero di iterazioni dell'algoritmo di ottimizzazione degli iperparametri.

4.3.2. Ottimizzazione degli iperparametri

L'ottimizzazione degli iperparametri¹⁷ è un passaggio cruciale nel machine learning che consiste nel determinare la configurazione ottimale per i parametri che governano il processo di apprendimento di un modello, ma che non vengono appresi direttamente dai dati durante la fase di addestramento. In altre parole, gli iperparametri sono impostazioni esterne che influenzano la struttura e il comportamento del modello, e la loro scelta incide significativamente sulle sue prestazioni.

Per distinguere chiaramente tra parametri e iperparametri, possiamo affermare che i parametri sono le variabili interne del modello apprese durante l'addestramento sui dati,

¹⁷ Si veda “Practical Bayesian Optimization of Machine Learning Algorithms” e “Bayesian Optimization and Data Science”

come ad esempio i coefficienti di una regressione lineare o i pesi di una rete neurale, mentre gli iperparametri sono impostazioni definite prima dell'addestramento che influenzano il processo di apprendimento, come ad esempio i coefficienti di regolarizzazione $L1$ e $L2$ descritti in precedenza per Lasso e Ridge, il learning rate in un algoritmo di ottimizzazione o la profondità di un albero decisionale.

L'obiettivo primario dell'ottimizzazione degli iperparametri è individuare la combinazione di valori che massimizza la capacità del modello di generalizzare a dati non visti, prevenendo fenomeni di underfitting o overfitting.

Esistono diverse tecniche di ottimizzazione ben consolidate e affermate. Grid Search rappresenta sicuramente la più semplice e intuitiva in quanto esplora sistematicamente tutte le possibili combinazioni di valori definiti per ciascun iperparametro all'interno di una griglia predefinita. Il modello viene addestrato e valutato per ogni combinazione, e quella che produce le migliori prestazioni su un set di validazione o tramite cross-validation viene selezionata. Questo metodo, sebbene semplice da implementare e in grado di trovare l'ottimo entro la griglia, diventa computazionalmente oneroso quando si vogliono testare molti iperparametri o ampi intervalli di valori.

La Random Search¹⁸, invece, campiona casualmente un numero predefinito di combinazioni di iperparametri da distribuzioni definite per ciascuno di essi. Addestrando e valutando il modello per ogni combinazione campionata, si cerca di individuare una buona soluzione. Questo approccio è meno oneroso computazionalmente della ricerca esaustiva, specialmente in spazi di ricerca ad alta dimensionalità, e può essere sorprendentemente efficace. Tuttavia, la ricerca non è guidata, ovvero non segue un qualche criterio di convergenza: è infatti possibile individuare una buona combinazione di iperparametri già alle prime iterazioni e continuare a testare altre combinazioni fino a quando non si raggiunge il numero prefissato di iterazioni. Per ovviare a questa limitazione e individuare un algoritmo più efficiente, è stata sviluppata l'ottimizzazione Bayesiana, nota come Bayesian Search.

Bayesian Search

L'algoritmo Bayesian Search, emerge come una metodologia di ottimizzazione globale particolarmente adatta a funzioni obiettivo il cui calcolo è computazionalmente oneroso, ovvero che richiedono un notevole dispendio di risorse o tempo per la loro valutazione. A differenza delle altre due tecniche descritti in precedenza, la Bayesian Search sfrutta

¹⁸ Si veda "Random Search for Hyper-Parameter Optimization"

l'apprendimento dalle valutazioni precedenti per orientare la ricerca verso le regioni più promettenti dello spazio degli iperparametri.

Il nucleo di questa metodologia risiede nell'impiego di due componenti fondamentali: un modello surrogato e una funzione di acquisizione. Il modello surrogato, tipicamente un Processo Gaussiano (GP), funge da approssimazione della funzione obiettivo sconosciuta $f(x)$. A differenza di una semplice predizione puntuale, il GP fornisce una distribuzione di probabilità sui possibili valori di $f(x)$ per ogni punto x nello spazio degli iperparametri, quantificando anche l'incertezza associata a tale predizione. Questa stima dell'incertezza è cruciale per il bilanciamento tra esplorazione e sfruttamento.

Il processo di aggiornamento del GP avviene tramite l'inferenza Bayesiana, combinando una conoscenza a priori sulla forma della funzione obiettivo (codificata nella distribuzione a priori del GP) con le informazioni ottenute dalle valutazioni precedenti (la verosimiglianza dei dati osservati). Il risultato di questa combinazione è una distribuzione a posteriori sul GP, che rappresenta la conoscenza aggiornata della funzione obiettivo dopo aver considerato i dati. Questo processo può essere sintetizzato con la formula del teorema di Bayes:

$$p(f|X) \propto p(X|f) p(f)$$

dove $p(f|X)$ è la distribuzione a posteriori del GP dato l'insieme di dati osservati X , $p(X|f)$ è la verosimiglianza dei dati X dato il GP f , e $p(f)$ è la distribuzione a priori sul GP.

La funzione di acquisizione $a(x)$, invece, guida la scelta del successivo punto da valutare. Il suo ruolo è quello di bilanciare l'esplorazione e lo sfruttamento, valutando regioni con alta incertezza e regioni dove il modello surrogato prevede valori ottimi. Esistono diverse funzioni di acquisizione, come l'Expected Improvement (EI), il Probability of Improvement (PI) e l'Upper Confidence Bound (UCB). Queste funzioni trasformano la distribuzione a posteriori del GP in un punteggio per ogni punto x , e il punto con il punteggio più alto viene scelto per la successiva valutazione della funzione obiettivo. La definizione generale della funzione di acquisizione ha forma $a(x) = g(\mu(x), \sigma(x))$, dove $\mu(x)$ e $\sigma(x)$ sono rispettivamente la media e la deviazione standard della distribuzione a posteriori del GP in x , e g è una funzione che specifica la strategia di bilanciamento tra esplorazione e sfruttamento.

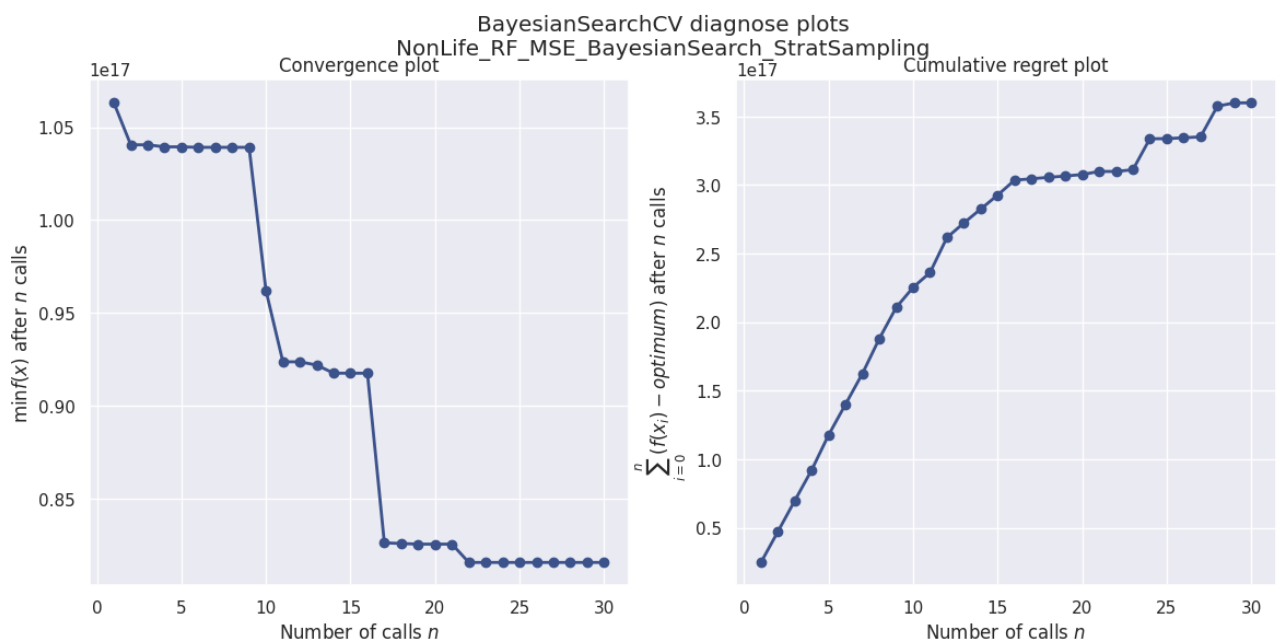
L'algoritmo si ripete iterativamente: si valuta la funzione obiettivo nel punto selezionato, si aggiorna il modello surrogato, si calcola la funzione di acquisizione e si seleziona il nuovo punto. Questo ciclo continua fino a soddisfare un criterio di arresto, come un numero

massimo di iterazioni o un budget computazionale prefissato. La capacità di gestire l'incertezza rende la ricerca Bayesiana particolarmente efficace per l'ottimizzazione di funzioni costose, consentendo di ottenere buone soluzioni con un numero limitato di valutazioni.

Nel caso in studio, si è dapprima deciso di applicare la Random Search in quanto la numerosità campionaria era sufficientemente bassa per riuscire ad eseguire l'ottimizzazione degli iperparametri in tempi brevi. Tuttavia, esplorando diverse alternative modellistiche, si è raggiunto un punto in cui la scelta dello spazio degli iperparametri diventava determinante nella buona riuscita del modello, richiedendo la definizione di uno spazio sempre più ampio, un numero sempre maggiore di iterazioni (da 100 a 1000) e tempi di elaborazione sempre maggiori. La Bayesian Search, invece, ha consentito di ridurre a 10 – 40 iterazioni il processo di ottimizzazione, fornendo anche un'indicazione sulla convergenza dell'algoritmo.

A titolo esemplificativo, di seguito è riportata una parte della diagnostica dell'algoritmo Bayesian Search durante la stima di una Random Forest sul modulo Non-Life, valutato con la metrica MSE (Mean Squared Error):

Figura 28: Bayesian Search RF – SCR Non-Life

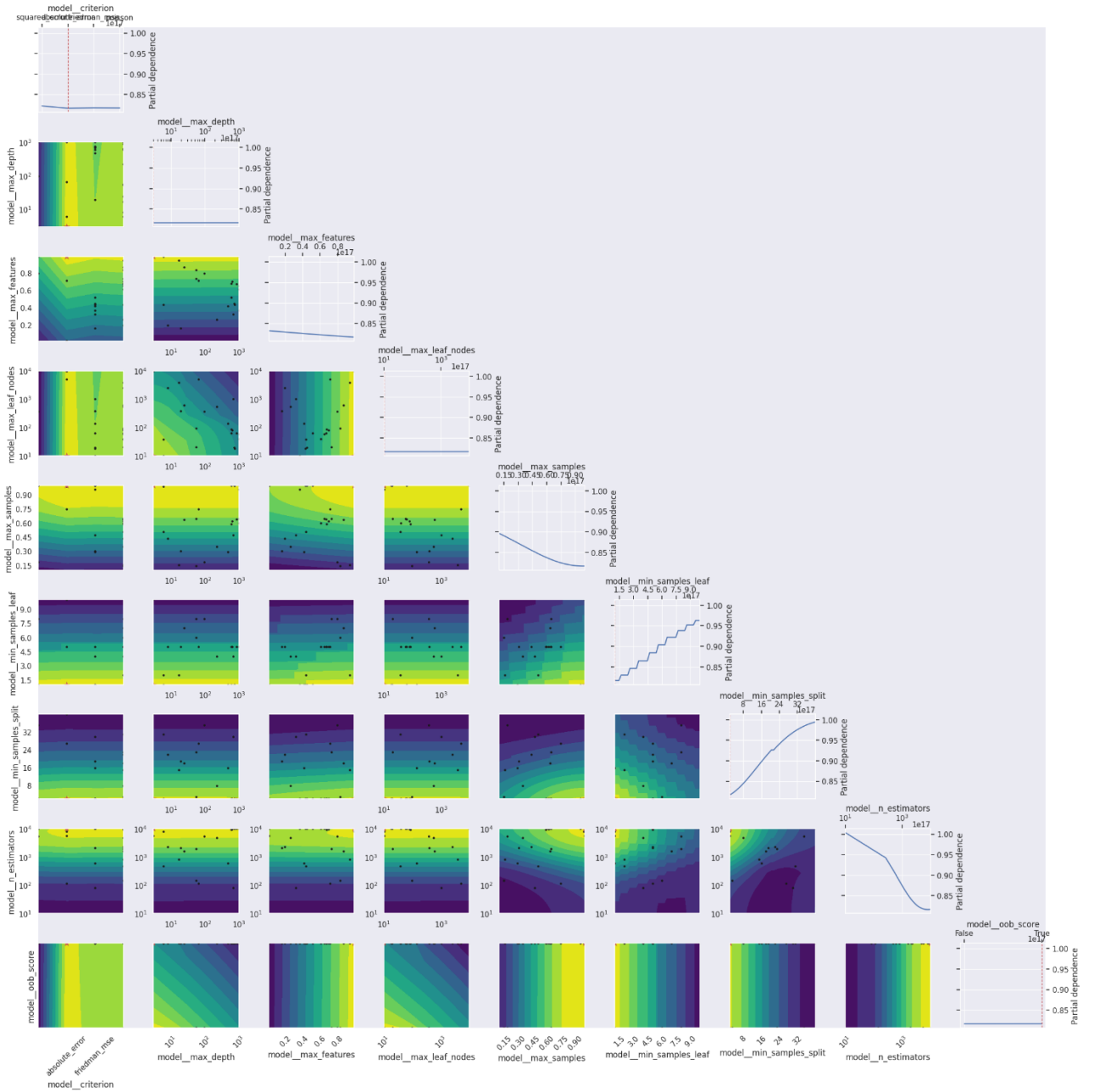


A sinistra si trova il grafico di convergenza, che mostra come il valore della metrica diminuisce dopo ciascuna iterazione e si stabilizza già a partire dalla 22-esima iterazione. Il grafico di destra mostra quanto velocemente l'algoritmo si allontana dal risultato ottimale: nel momento in cui si stabilizza, significa che ha individuato una zona ottimale nello spazio

degli iperparametri. Si vede infatti che a partire dalla 16-esima iterazione l'algoritmo ha agganciato la zona ottimale, e questo si riflette nella tendenza a convergere del grafico di sinistra.

Sempre a titolo esemplificativo, per lo stesso modello si riporta il grafico che mostra le combinazioni di iperparametri testate, rappresentate dai punti neri, e il valore della funzione obiettivo, rappresentato al variare di ciascuna coppia di iperparametri. Il grafico deve essere letto come se fosse una matrice simmetrica. I grafici marginali, denominati Partial Dependence Plot, sono uno strumento molto usato in ambito Machine-Learning: l'asse delle ordinate si riferisce alla funzione obiettivo $\min(f(x))$, mentre l'asse delle ascisse è relativa a un singolo iperparametro. La curva del PDP rappresenta un'approssimazione della curva di minimizzazione reale, ottenuta tramite campionamento e aggregazione.

Figura 29: PDP RF – SCR Non-Life



4.3.3. Model Ranking

Per valutare lo spazio dei possibili esiti di stima, si è deciso di testare le metriche sopra descritte e metterle a confronto subito dopo la fase di ottimizzazione degli iperparametri. Questo processo rappresenta la prima fase di selezione del miglior modello, effettuata rispetto ai modelli stimati per un singolo modulo e un singolo algoritmo. Ad esempio, per il modulo Life sono stati stimati, tra i tanti, quattro modelli GPR, ciascuno con una metrica di ottimizzazione diversa (MAE, MSE, MAPE e R^2).

Come già spiegato, ciascuna metrica è utile a prioritizzare aspetti diversi del concetto di errore di stima. Allo stesso modo, gli algoritmi come XGBoost, per i quali è possibile definire una metrica interna, possono comportarsi in modo diverso variando la metrica esterna usata durante l'ottimizzazione degli iperparametri. Avendo a che fare con un problema ad alta dimensionalità, questi aspetti possono risultare determinanti e meritano di essere vagliati.

Si rende dunque necessario definire un criterio che consenta di determinare quale sia la versione migliore della combinazione modulo-algoritmo-metrica. Per questo motivo è stata sviluppata una semplice tecnica di model ranking, basata sulla costruzione di una classifica che prenda in considerazione tutte le metriche finora analizzate.

Semplificando, si procede per metrica, andando a calcolarne il valore per ciascuna combinazione di modulo-modello, ottenendo una classifica delle combinazioni più performanti. Si ripete il procedimento con la metrica successiva, collezionando una serie di punteggi per ciascuna combinazione di modulo-modello-metrica. Infine, si sommano i punteggi su tutte le metriche di ciascuna combinazione.

Ai fini della fase di ensemble selection spiegata più avanti (paragrafo 4.3.4), per ogni combinazione modulo-algoritmo viene tenuta quella con punteggio più alto. Ad esempio, avendo per il modulo Life sono stati stimati quattro modelli XGBoost. Per ogni algoritmo si sceglie quello che più performante, in modo da escludere le altre versioni dello stesso algoritmo che fanno uso di metriche di ottimizzazione diverse.

Per ciascun modulo, l'intero processo di model ranking porta dunque alla selezione di tanti modelli quanti sono gli algoritmi testati. Questo risultato rappresenta l'input per la fase di ensemble learning.

4.3.4. *Ensemble Learning*

Una volta stimati i migliori modelli per ciascun algoritmo e per ciascun modulo tramite la tecnica del model ranking, è necessario adottare una metodologia che consenta di individuare una sola combinazione vincente per i singoli moduli. Questa tematica si inserisce all'interno dell'Ensemble Learning, un paradigma del Machine Learning che si fonda sull'aggregazione di molteplici modelli predittivi individuali, detti *base learners*, con l'obiettivo di ottenere prestazioni superiori rispetto a quelle raggiungibili da un singolo modello isolato. Quanto esposto, tuttavia, si discosta da tecniche note e si propone come un metodo alternativo.

Il concetto di ensemble learning si basa sull'assunto secondo cui diversi modelli, addestrati su diverse porzioni di dati o con diversi algoritmi, come nel caso qui descritto, possano catturare aspetti differenti della complessità intrinseca nei dati. Combinando le loro predizioni, si ottiene una sorta di consenso che tende a ridurre l'errore sistematico e a migliorare la generalizzazione a dati non visti prima. Invece di affidarsi a un unico modello, potenzialmente soggetto a errori specifici, l'approccio d'insieme sfrutta la diversità per ottenere una predizione più stabile e affidabile.

Le strategie più comunemente impiegate sono:

- **Bagging (Bootstrap Aggregating):** i base learners vengono addestrati su diversi sottoinsiemi del dataset di training, ottenuti tramite campionamento con reinserimento (bootstrap). Le predizioni dei singoli modelli vengono poi aggregate, ad esempio tramite media per problemi di regressione o voto a maggioranza per problemi di classificazione. Il bagging mira a ridurre la varianza del modello, migliorando la stabilità e prevenendo l'overfitting. Un esempio noto di bagging è la Random Forest.
- **Boosting:** addestra i base learners in modo sequenziale, e ogni nuovo modello cerca di correggere gli errori commessi dai modelli precedenti. I modelli vengono pesati in base alla loro performance, dando maggiore importanza a quelli che hanno commesso meno errori. Il boosting mira a ridurre il bias e la varianza del modello, migliorando l'accuratezza predittiva. Esempi noti di boosting sono AdaBoost, Gradient Boosting e XGBoost.
- **Stacking:** prevede l'addestramento di diversi base learners e l'utilizzo di un meta-modello, o meta-learner, per combinare le loro predizioni. Il meta-modello viene addestrato su un dataset composto dalle predizioni dei base learners, con l'obiettivo di apprendere la combinazione ottimale delle diverse predizioni. Lo stacking mira a sfruttare i punti di forza di diversi modelli per ottenere una predizione ancora più accurata.

Le prime due tecniche, Bagging e Boosting, solitamente sono implementate come step intermedi dei modelli in cui hanno trovato applicazione e, molto raramente, si vedono implementazioni *esterne* ai modelli. Lo Stacking invece rappresenta una tecnica più semplice e agnostica, e ha fornito l'ispirazione per il metodo sviluppato in questo lavoro.

Ai fini della selezione della miglior combinazione di modelli, si è proceduto a definire le liste di modelli candidati:

1. ensemble di modelli GLM con soluzione OLS;
2. ensemble di modelli GLM con algoritmo SVD;
3. ensemble dei migliori modelli di ciascun modulo (inclusi i GLM con algoritmo SVD);
4. ensemble definito come la lista dei primi migliori modelli di ciascun modulo, classificati in base alla tecnica di model ranking descritta nel precedente paragrafo. Questo ensemble è stato utilizzato come input per la tecnica di *ensemble selection* descritta nel prossimo paragrafo.

Si rimanda al paragrafo 4.4 per maggior informazioni sulle liste di modelli candidati.

Sviluppo di un algoritmo di Ensemble Selection

Come già ampiamente illustrato, il problema qui trattato presenta elementi di criticità che limitano enormemente le tecniche comunemente utilizzate. Da un lato, la presenza di cinque variabili target e l'obiettivo di stimare una loro trasformata, pone il problema della separabilità, o indipendenza, dei modelli: tra i moduli sussiste una correlazione reciproca, tant'è che il calcolo del SCR prevede l'imposizione di una matrice di correlazione che attribuisce un peso, più precisamente un coefficiente di correlazione, a questo legame.

La correlazione esistente tra i moduli non consente peraltro di selezionare semplicemente il modello migliore per ogni modulo. L'impatto che i moduli hanno congiuntamente sul SCR stimato è condizionato dalle loro interazioni e dall'entità del peso attribuito dalla formula stessa del SCR, tutti elementi osservabili solamente a posteriori e dopo aver stimato i moduli stessi. Per questo motivo, un modello con prestazioni peggiori rispetto ad un altro, per uno specifico modulo, potrebbe dare risultati migliori una volta valutato assieme agli altri moduli.

I modelli dei diversi moduli sono addestrati su feature comuni, con la conseguenza che i valori numerici che guidano i modelli di un modulo sono gli stessi che guidano i modelli degli altri moduli. Inoltre, va considerato che ogni modulo ha caratteristiche distributive diverse dagli altri moduli: non è per nulla scontato che uno stesso algoritmo riesca a modellare correttamente tutti i moduli, ma è probabile che per ogni modulo sia necessario individuare un algoritmo specifico.

L'algoritmo sviluppato per tentare di conciliare queste criticità, esegue una ricerca esaustiva nello spazio delle combinazioni di modelli, calcola le metriche per ogni combinazione, le normalizza e seleziona la combinazione che minimizza una serie di funzioni obiettivo (i

percentili da 1 a 100) definite sulle metriche normalizzate. Questo approccio combina elementi di *ensemble selection* e *model averaging* per ottimizzare la stima di variabili target multiple.

Per descrivere formalmente l'algoritmo, è necessario introdurre gli elementi principali:

- $\mathbf{M} = \{M_1, M_2, \dots, M_5\}$ è l'insieme dei $J = 5$ moduli (Life, Non-Life, Market, Default, Health).
- $\mathbf{M}_j = \{m_{j1}, m_{j2}, \dots, m_{jP}\}$ è l'insieme dei modelli candidati per il modulo j -esimo, dove P è il numero di modelli candidati per ciascun modulo ($P = 8$). I modelli candidati sono stati preventivamente selezionati tramite model ranking delle stesse metriche usate durante dall'algoritmo di ensemble selection.
- $\mathbf{c}_k = (m_{1k}, m_{2k}, \dots, m_{8k})$ è la k -esima combinazione di modelli, dove $k = 1 \dots P$.
- $\mathbf{X}_{Test}$ è il set di Test, costituito dalle 17 osservazioni del dataset originale escluse dal processo di addestramento dei modelli.
- $\hat{y}_{jck}(\mathbf{x})$ è la predizione del modello j -esimo nella combinazione $\mathbf{c}_k$ per l'istanza $\mathbf{x} \in \mathbf{X}_{Test}$.
- $\hat{y}_{ck}(\mathbf{x}) = SCR(\hat{y}_{1ck}(\mathbf{x}), \hat{y}_{2ck}(\mathbf{x}), \dots, \hat{y}_{5ck}(\mathbf{x}))$ è la funzione di aggregazione delle predizioni per l'istanza $\mathbf{x}$ nella combinazione $\mathbf{c}_k$. $SCR(.)$ applica la formula standard di calcolo dell'SCR, producendo una singola stima aggregata $\hat{y}_{ck}$.
- $y(\mathbf{x})$ rappresenta il valore teorico del SCR calcolato sui valori reali dei moduli Health, Life, Non-Life, Market e Default osservati per l'osservazione $\mathbf{x}$.
- $L_{\mathbf{c}_k}^{(u)}(y(\mathbf{x}), \hat{y}_{ck}(\mathbf{x}))$ è la funzione di perdita e corrisponde a una delle seguenti metriche:
 - $L_{\mathbf{c}_k}^{(1)} = R^2$
 - $L_{\mathbf{c}_k}^{(2)} = MSE$
 - $L_{\mathbf{c}_k}^{(3)} = MAE$
 - $L_{\mathbf{c}_k}^{(4)} = MAPE$
- $L_{norm, \mathbf{c}_k}^{(u)}$ rappresenta la metrica $L_{\mathbf{c}_k}^{(u)}$ normalizzata rispetto a tutte le combinazioni $\mathbf{c}_k$.
- $O_{\mathbf{c}_k} = O(L_{norm, \mathbf{c}_k}^{(1)}, \dots, L_{norm, \mathbf{c}_k}^{(4)})$ rappresenta la funzione obiettivo per la combinazione $\mathbf{c}_k$ sul dataset di Test. Si è deciso di utilizzare la funzione percentile, da cui:

$$O_{\mathbf{c}_k}(\cdot, q) = percentile_{\mathbf{c}_k}(\cdot, q)$$

dove $q = 1, 2, \dots, 100$ è il percentile.

- N_{c_k} è il numero totale di combinazioni.

L'algoritmo prevede l'esecuzione dei seguenti passaggi:

1. Si fissa un valore di profondità massima P , in questo caso $P = 8$, per la selezione dei primi modelli migliori di ciascuno dei $J = 5$ moduli. Il numero totale di combinazioni da testare risulta:

$$N_{JP} = P^J = 8^5 = 32.768$$

2. Si generano le $N_{JP} = 32.768$ combinazioni di modelli $c_P \in C$, dove C è l'insieme di tutte le possibili disposizioni con ripetizione ottenibili con i 5 moduli e il numero di modelli M_j stimati per ciascun modulo (si ricorda che si è ristretto il problema di ensemble selection ai primi 8 modelli migliori di ciascun modulo).

3. Per ogni combinazione $c_k \in C$, si procede come segue:

- a. Per ogni osservazione $x \in X_{Test}$:

- i. Si calcolano le predizioni individuali: $\hat{y}_{jck}(x)$ per $j = 1, \dots, 5$.
- ii. Si combinano le predizioni:

$$\hat{y}_{jck}(x) = F(\hat{y}_{1ck}(x), \hat{y}_{2ck}(x), \dots, \hat{y}_{5ck}(x))$$

- b. Si calcolano le metriche di valutazione per l'intera combinazione c_k :

$$L_{c_k}^{(u)}(y(x), \hat{y}_{c_k}(x)), \text{ con } u = \{1, 2, 3, 4\}.$$

Avendo unità di misura differenti, le metriche utilizzate sono eterogenee. Risulta pertanto è necessario normalizzarle, così da poterle aggregare. L'algoritmo procede come segue:

4. Normalizzazione del vettore di metriche $L^{(u)}$:

- a. Si calcolano $\min_u = \min(L^{(u)})$ e $\max_u = \max(L^{(u)})$, rispettivamente il valore minimo e massimo calcolati su tutte le combinazioni per la metrica u -esima.

- b. La metrica normalizzata $L_{norm, c_k}^{(u)}$ per la combinazione c_k assume la forma:

- $L_{norm, c_k}^{(1)} = (1 - L_{c_k}^{(1)}) / (1 - \min_u)$ per R^2 ($u = 1$).

- $L_{norm, c_k}^{(u)} = \frac{(L_{c_k}^{(u)} - \min_u)}{(\max_u - \min_u)}$ per le altre metriche ($u > 1$).

5. Per ogni combinazione $c_k \in C$, si calcola la funzione obiettivo utilizzando i valori normalizzati delle metriche:

$$O_{q, c_k} = O_q(L_{norm, c_k})$$

dove $\mathcal{L} = \{L_{norm, c_k}^{(1)}, \dots, L_{norm, c_k}^{(4)}\}$. La funzione $O_q(.)$ calcola il percentile q della distribuzione delle metriche $\mathcal{L}_{norm, c_k}$.

6. Si cerca la combinazione $\mathbf{c}_k^{(opt)}$ che minimizza la funzione obiettivo:

$$\mathbf{c}_k^{(opt)} = \underset{\mathbf{c}_k \in \mathcal{C}}{\operatorname{argmin}} O_q(\mathcal{L}_{norm, c_k})$$

Si tratta di trovare la combinazione di modelli che produce un set di metriche $\mathcal{L}_{norm, c_k}$ il cui percentile q è il più piccolo osservato tra tutte le combinazioni.

7. L'intero algoritmo viene ripetuto per tutti i percentili compresi nel range $q = 1, \dots, 100$, portando alla generazione di 100 combinazioni ottimali.
8. Vengono generati dei grafici di calibrazione che riportano l'andamento delle metriche al variare del percentile q , così come la variazione di una metrica in funzione di un'altra. Così facendo, si decide arbitrariamente il valore di q che conduce al miglior compromesso tra le metriche. Sulla base di questa valutazione, si individuano i modelli di ciascun modulo associato a quel particolare percentile, determinando quindi la combinazione ideale.

Impatto sul trade-off Bias/Varianza

In ambito Machine Learning, uno degli obiettivi principali è la costruzione di modelli che generalizzino bene a dati non visti, ovvero che siano in grado di fare predizioni accurate su dati non visti durante l'addestramento. L'errore di generalizzazione di un modello può essere scomposto in due componenti principali: il bias e la varianza.

Il bias, noto anche come errore sistematico, rappresenta l'errore introdotto dalle assunzioni semplificatrici fatte dal modello per apprendere la variabile target, modellata da una funzione $f(.)$. Un modello con alto bias introduce assunzioni troppo forti sulla forma della funzione, non riuscendo a catturare la complessità reale dei dati. Questo porta a un fenomeno chiamato underfitting, in cui il modello non si adatta bene neanche ai dati di training, come ad esempio l'utilizzo di un modello lineare in cui si cerca di approssimare una relazione non lineare.

La varianza rappresenta invece la sensibilità del modello alle fluttuazioni presenti nel set di Train. Un modello con valori elevati della varianza si adatta eccessivamente ai dati, incluso il rumore presente in essi. Questo porta a un fenomeno chiamato overfitting, in cui il modello performa molto bene sul set di Train ma performa male sul set di Test, come ad esempio un albero decisionale molto profondo, che memorizza ogni dettaglio dei dati di addestramento.

La ricerca del trade-off bias/varianza consiste nel trovare un equilibrio tra queste due componenti dell'errore. Idealmente, si vorrebbe un modello con basso bias, in grado di

catturare bene la complessità dei dati, e bassa varianza, dunque che sia anche robusto alle fluttuazioni. Tuttavia, spesso ridurre il bias porta ad aumentare la varianza, e viceversa.

La metodologia proposta, grazie all'introduzione del model ranking, agisce direttamente sulla riduzione del bias. Selezionando i modelli con le migliori performance, testati sui set di Validation interni alla cross-validation, si escludono i modelli con un bias elevato e che non riescono a catturare adeguatamente la complessità dei dati. Questa scrematura preliminare dei modelli candidati porta a un insieme di base learners con un bias generalmente inferiore rispetto a un insieme di modelli selezionati casualmente.

L'algoritmo di ensemble selection, a sua volta, contribuisce principalmente alla riduzione della varianza. La selezione di una specifica combinazione di modelli, anziché l'utilizzo di tutti i modelli con pesi (come nel model averaging), restringe lo spazio delle possibili predizioni e rende l'ensemble meno sensibile alle fluttuazioni del training set. Inoltre, la diversità introdotta dall'utilizzo di differenti algoritmi per ciascun modulo e la normalizzazione delle metriche contribuiscono ulteriormente a ridurre la varianza.

L'utilizzo dei percentili come funzione obiettivo nell'ensemble selection introduce una nuova dimensione al trade-off. La scelta del percentile influenza il tipo di compromesso che si ottiene tra le diverse metriche di valutazione: minimizzare un percentile basso, ad esempio il 25-esimo percentile ($q = 25$), enfatizza la ricerca di combinazioni che performano bene nella maggior parte delle metriche, privilegiando una combinazione più equilibrata e potenzialmente riducendo la varianza complessiva, ma potrebbe portare a un leggero aumento del bias in alcune metriche specifiche. Al contrario, minimizzare un percentile alto, ad esempio $q = 75$, può portare a selezionare combinazioni che eccellono in alcune metriche, potenzialmente a scapito di altre, in favore di una riduzione del bias in quelle metriche specifiche e un aumento della varianza complessiva. In altre parole, la scelta del percentile permette di calibrare il trade-off tra le diverse metriche e, di conseguenza, di influenzare indirettamente il trade-off bias/varianza a livello di ensemble.

Impatto sull'alta dimensionalità

La metodologia proposta, combinando le tecniche di model ranking ed ensemble selection, mitiga indirettamente il problema dell'alta dimensionalità attraverso diversi meccanismi sinergici. Innanzitutto, la suddivisione del problema in sotto-problemi, uno per ciascun

modulo, riduce la dimensionalità di ogni singolo problema di regressione da un punto di vista della correlazione tra i moduli e le features. Questa decomposizione permette ai modelli di concentrarsi su sottoinsiemi di feature potenzialmente più rilevanti per ciascuna variabile target, attenuando l'impatto delle feature irrilevanti o ridondanti tipiche dei contesti ad alta dimensionalità.

Il model ranking agisce come un filtro preliminare, selezionando i modelli che mostrano le migliori performance nella fase immediatamente successiva all'ottimizzazione degli iperparametri e della cross-validation. Questo processo di selezione è cruciale perché i modelli che superano questa fase tendono a essere quelli che gestiscono meglio la presenza di un numero elevato di feature, dando maggiore peso a quelle informative. In altre parole, il model ranking effettua una forma di selezione implicita delle feature, privilegiando i modelli intrinsecamente più robusti all'alta dimensionalità.

L'ensemble selection, a sua volta, apporta un contributo specifico alla gestione dell'alta dimensionalità. Pur non essendo una tecnica di riduzione di dimensionalità esplicita, l'algoritmo seleziona una specifica combinazione di modelli, selezionando implicitamente anche un sottoinsieme di feature, ovvero quelle a cui i modelli selezionati danno maggiore importanza. Questo processo di selezione, guidato dalla minimizzazione della funzione obiettivo sulle metriche normalizzate, contribuisce a ridurre ulteriormente l'impatto delle feature irrilevanti. Inoltre, la riduzione del numero di modelli candidati operata dal model ranking ha un impatto diretto sulla complessità computazionale dell'ensemble selection. Riducendo il numero di combinazioni possibili, si rende l'ottimizzazione più efficiente anche in contesti con un elevato numero di feature e modelli candidati. Scegliendo una combinazione specifica tra un insieme ristretto di modelli pre-selezionati, si riduce la varianza e si mitiga ulteriormente gli effetti negativi dell'alta dimensionalità, rafforzando l'azione di filtraggio delle feature già operata dal model ranking.

In conclusione, pur non affrontando direttamente l'alta dimensionalità con tecniche esplicite di riduzione di dimensionalità, come PCA o selezione di feature basata su filtri o wrapper, la metodologia proposta mitiga efficacemente il problema attraverso l'analisi indipendente dei singoli moduli, la selezione implicita di feature operata dal model ranking e la riduzione del numero di combinazioni da valutare, offrendo un compromesso efficace tra accuratezza e complessità computazionale.

La calibrazione, in generale, si riferisce al processo di ottimizzazione dei parametri di un modello al fine di migliorare la sua accuratezza predittiva. In contesti in cui si desidera ottimizzare un modello rispetto a più criteri di performance (metriche), si parla di calibrazione multi-obiettivo. Tale processo mira a trovare un insieme di soluzioni di compromesso, rappresentate da combinazioni di modelli, che offrono un bilanciamento accettabile tra le diverse metriche considerate. Questo approccio è particolarmente rilevante nel contesto dell'ensemble selection, dove si desidera combinare modelli che possono eccellere rispetto alle metriche specifiche del problema. La calibrazione multi-obiettivo cerca di superare la limitazione di ottimizzare rispetto a una singola metrica, che potrebbe portare a trascurare altri aspetti importanti della performance del modello.

Nell'algoritmo di ensemble selection precedentemente illustrato, la performance di ciascuna combinazione di modelli viene valutata attraverso un insieme di metriche normalizzate $L_{norm,c_k} = (L_{norm,c_k}^{(1)}, \dots, L_{norm,c_k}^{(4)})$, aggregate per mezzo della funzione obiettivo percentile:

$$O_{q,c_k} = O_q(L_{norm,c_k}^{(1)}, \dots, L_{norm,c_k}^{(4)})$$

dove q è il percentile delle metriche L_{norm,c_k} . Facendo variare q nel range $[1, 100]$, si ottengono 100 sequenze di metriche. Ogni sequenza sarà costituita dalle quattro metriche $L_{norm,c_k}^{(u)}$ associate al rispettivo percentile q delle quali produce una serie di combinazioni di valori che rappresentano la combinazione q -esima che minimizza il percentile q , dando la possibilità di valutare l'andamento delle metriche al variare di q .

L'idea alla base di questo approccio risiede nella maggiore robustezza ai valori anomali (outlier) offerta dalla minimizzazione di un percentile rispetto alla minimizzazione di un valore aggregato come la somma o la media. Di conseguenza, minimizzando un percentile, si ricercano combinazioni di modelli che offrono una buona performance nella “maggior parte” delle metriche, anche a costo di performance leggermente inferiori in alcune metriche specifiche. Tale approccio si rivela particolarmente utile quando si sospetta che alcune metriche possano essere più soggette a rumore o a errori di stima, e si preferisce quindi concentrarsi su una performance più consistente tra le diverse metriche.

L'utilizzo dei percentili come funzione obiettivo offre diversi vantaggi, che arricchiscono l'analisi e la comprensione del comportamento dell'ensemble selection. La robustezza agli outlier implica che la selezione della combinazione ottimale sarà meno influenzata da singole metriche con performance particolarmente degradate, ma si baserà su una

valutazione più complessiva e bilanciata. Variando il valore del percentile p , si ha la possibilità di esplorare il trade-off tra le diverse metriche.

4.4. Risultati

Avendo a disposizione un ampio set di strumenti, costituito da modelli, metriche (interne ed esterne) e tecniche di campionamento, è stato stimato un numero complessivo di oltre 250 modelli per le stime dei modelli basati sulla tecnica del campionamento stratificato. Grazie all'applicazione del model ranking, sono stati mantenuti circa 13 modelli per modulo, riducendo il totale a 65 modelli. Il passaggio alla fase di ensemble selection è stato applicato ai primi $P = 8$ modelli di ciascuno dei 5 moduli, per un totale di 40 modelli distinti. La scelta di limitarsi ai primi otto modelli è dipesa da questioni di efficienza computazionale e dal fatto che una quota significativa dei modelli meno performanti presenta un rank (punteggio) molto basso, rendendone inutile la presenza durante il processo di ensemble selection.

Di seguito sono riportate le liste di modelli candidati utilizzate per valutare la bontà dei metodi finora descritti:

1. **OLS**: ensemble di modelli GLM con soluzione OLS;
2. **SVD**: ensemble di modelli GLM con algoritmo SVD;
3. **Model-Ranking**: ensemble dei migliori modelli di ciascun modulo (inclusi i GLM con algoritmo SVD), intesi come l'insieme dei modelli classificati al primo posto con il metodo del model ranking applicato a ciascun modulo:
 - Non-Life: GLM Lineare con algoritmo SVD
 - Life: KNN (MAE)
 - Market: Bayesian Ridge (MAPE)
 - Health: SVM Lineare (MAE)
 - Default: SVM Lineare (R^2)
4. **Ensemble-Selection**: i modelli utilizzati sono i primi otto di ciascun modulo classificati con il metodo del model ranking. L'algoritmo di ensemble selection è stato quindi applicato su tutte le $8^5 = 32.768$ combinazioni generate partendo dalla seguente tabella (i modelli sono ordinati in ordine decrescente di punteggio per ciascun modulo):

Figura 30: Primi otto modelli di ciascun modulo

Non-Life	Life	Market	Health	Default
GLM+SVD	KNN (MAE)	Bayesian Ridge (MAPE)	SVM Lineare (MAE)	SVM Lineare (R^2)
Bayesian Ridge (MAE)	SVM Lineare (MSE)	Ridge (MSE)	*XGBoost (R^2)	Ridge (R^2)
ElasticNet (MSE)	GPR (MSE)	NN (MSE)	KNN (R^2)	Bayesian Ridge (MSE)
Ridge (MSE)	*XGBoost (MSE)	GLM+SVD	Random Forest (R^2)	GLM+SVD
SVM Lineare (R^2)	Random Forest (R^2)	SVR Lineare (R^2)	Ridge (R^2)	GPR (MAE)
KNN (MAPE)	GLM+SVD	KNN (MSE)	Bayesian Ridge (R^2)	KNN (MSE)
SVM-Nu (MSE)	Bayesian Ridge (R^2)	GPR (MSE)	GPR (MSE)	**XGBoost (MAE)
NN (MAE)	SVR-Nu (MAE)	Random Forest (MSE)	GLM+SVD	Random Forest (R^2)

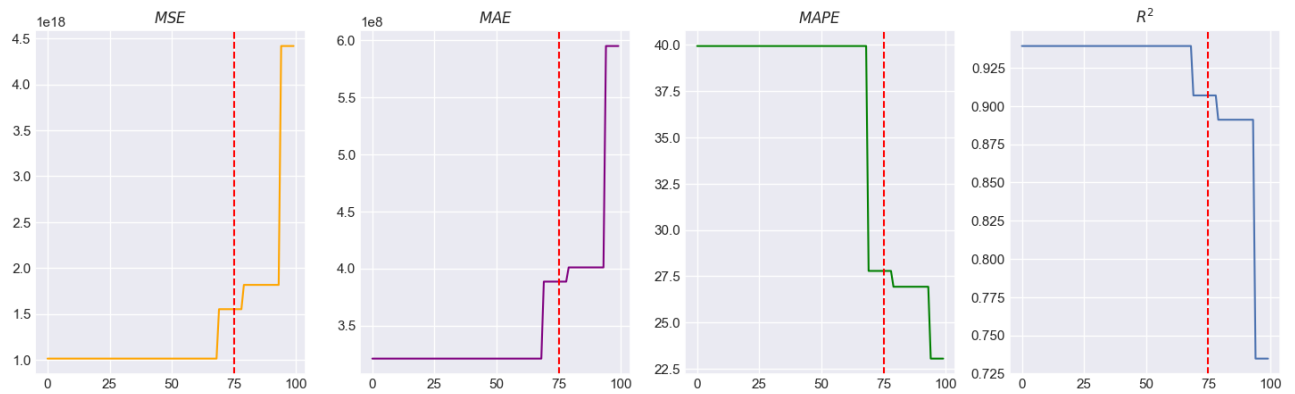
*XGBoost: versione dell'algoritmo in cui la funzione obiettivo è la funzione quantilica, con funzione di perdita pinball

**XGBoost: versione dell'algoritmo in cui la funzione obiettivo è la regressione lineare OLS, con funzione di perdita RMSE (radice quadrata di MSE)

L'algoritmo di ensemble selection è stato applicato calcolando la distribuzione percentile delle metriche normalizzate MSE, MAE, MAPE, R^2 rispetto ai valori stimati dell'SCR, consentendo di visualizzare l'andamento congiunto delle metriche e di effettuare la calibrazione del modello finale.

Si riportano i grafici che mostrano l'andamento delle metriche facenti parte della combinazione ottimale di ciascun percentile q , calcolate sul set di Test: facendo variare q , sono state individuate 100 sequenze di metriche, ognuna costituita da quattro valori (MSE, MAE, MAPE, R^2), associate al percentile q minimo osservato per quello specifico valore di q :

Figura 31: Metriche per ciascun percentile



La linea rossa tratteggiata rappresenta il 75-esimo percentile ($q = 75$), in corrispondenza del quale si verificano una serie di condizioni desiderabili:

- il valore di MSE e MAE subisce un primo incremento;
- MAPE subisce un drastico decremento;
- R^2 decresce a un valore accettabile.

I risultati del processo di calibrazione sono riportati nella seguente tabella:

Figura 32: Risultati

Percentile q	MSE	MAE	MAPE	R^2	n_q	Ensemble
[1; 69]	1,01E+18	321091118	39,9	0,9392	69	Life: GPR (MSE) NonLife: GLM+SVD Market: GLM+SVD Default: GPR (MAE) Health: GPR (MSE)
[70; 77]	1,01E+18	388462113	27,8	0,9069	8	Life: GLM+SVD Non-Life: Elastic-Net (MSE) Market: Bayesian Ridge (MAPE) Default: GPR (MAE) Health: GPR (MSE)
[78; 79]	1,55E+18	388462113	27,8	0,9069	2	Life: Bayesian Ridge (R^2) Non-Life: Elastic-Net (MSE) Market: Bayesian Ridge (MAPE) Default: GPR (MAE) Health: GPR (MSE)
[80; 94]	1,82E+18	400893230	26,9	0,8910	15	Life: GPR (MSE) Non-Life: NN (MAE)

						Market: Bayesian Ridge (MAPE) Default: GPR (MAE) Health: GPR (MSE MSE)
[95; 100]	4,42E+18	594657526	23,0	0,7348	6	Life: GLM+SVD Non-Life: Elastic-Net (MSE) Market: SVR Lineare (R^2) Default: GPR_MAE Health: GPR_MSE

La colonna *Ensemble* riporta la combinazione dei modelli che minimizza i percentili q appartenenti al range percentile corrispettivo.

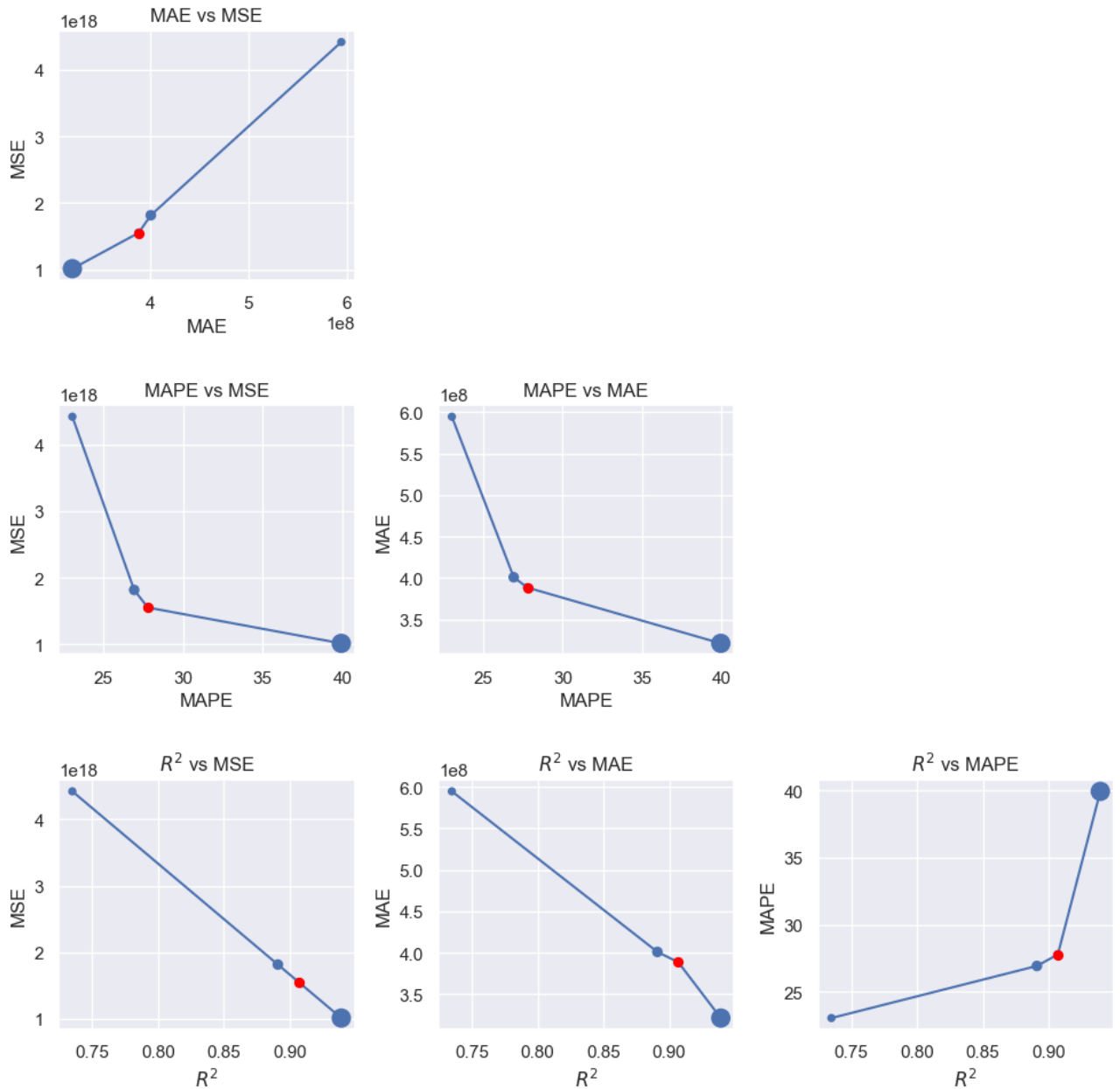
Accettando range percentili superiori rispetto a [70; 77], si migliorerebbe solo leggermente la metrica MAPE, a discapito delle altre tre metriche. Accettando la soglia percentile più alta, ovvero [95; 100], MAPE migliorerebbe ulteriormente, mentre MSE, MAE e R^2 raggiungerebbero valori inaccettabili. Soglie percentili inferiori a [70; 77], invece, porterebbero a un valore di MAPE troppo elevato, pur favorendo le altre metriche, in particolare il coefficiente R^2 .

Il 75-esimo percentile, così come tutto il range [70; 77], appare dunque il più equilibrato.

Guardando alla numerosità n_q , ovvero l'ampiezza di ciascun range percentile, appare evidente come la maggior parte delle combinazioni ottimali ($n_q = 69$) sia a favore dello scenario più scontato, in cui MSE, MAE e R^2 assumono valori ideali. Non si tratta tuttavia dello scenario ottimale in senso assoluto, rappresentato da $n_q = 8$. Ciò dimostra, in modo del tutto empirico e circoscritto al caso in studio, che il concetto di *voting* (elezione, o votazione) nell'ambito dell'Ensemble Learning non sempre è la scelta ottimale.

Considerazioni analoghe a quelle appena esposte, possono essere effettuate osservando l'andamento di ciascuna metrica rispetto a un'altra. L'attenzione è rivolta ai gomiti, ovvero quei punti del grafico in cui l'andamento di una metrica subisce un'impennata, positiva o negativa, rispetto a un'altra metrica. In rosso è rappresentato il gruppo di combinazioni associate al 75-esimo, mentre il diametro dei punti è proporzionale alla numerosità n_q di ciascun punto.

Figura 33: Confronto tra metriche



Appare evidente come il punto rosso, ovvero il range percentile [70; 77], si trovi in corrispondenza di un gomito, a ridosso di un aumento della metrica MAPE rispetto a R^2 tale da non giustificare valori di R^2 superiori. Analoga considerazione può essere fatta per MSE e MAE rispetto a MAPE.

Inoltre, il coefficiente R^2 risulta avere una relazione lineare negativa con MSE e MAE, mentre tra MSE e MAE sussiste una relazione lineare positiva. Per tutte le curve a gomito, l'andamento è di natura esponenziale, positiva per MAPE rispetto a R^2 , negativa per MSE e MAE rispetto a MAPE.

Procedendo nell'analisi dei risultati, si propone il confronto tra il metodo di ensemble selection e le altre metodologie (OLS, GLM+SVD e Model Ranking), con l'intento di valutarne le performance:

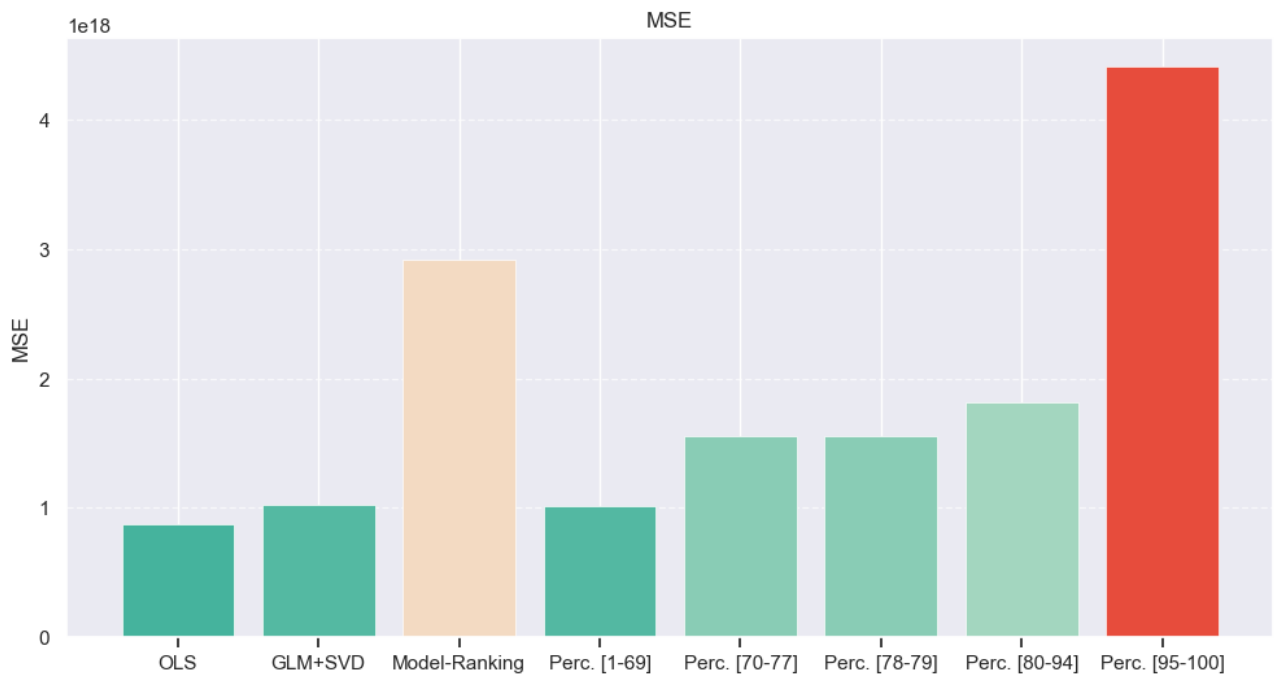
Figura 34: Confronto tra metodo Ensemble e metodologie OLS, GLM+SVD, Model Ranking

Metodologia	MSE	MAE	MAPE	R²
OLS	8,67E+17	335850490	54,6	0,9479
GLM+SVD	1,02E+18	330864113	43,9	0,9385
Model Ranking	2,91E+18	504524611	44,0	0,8254
Ensemble Selection [1; 69]	1,01E+18	321091118	39,9	0,9392
Ensemble Selection [70; 77]	1,01E+18	388462113	27,8	0,9069
Ensemble Selection [78; 79]	1,55E+18	388462113	27,8	0,9069
Ensemble Selection [80; 94]	1,82E+18	400893230	26,9	0,8910
Ensemble Selection [95; 100]	4,42E+18	594657526	23,0	0,7348

Gli stessi risultati sono stati graficati per agevolarne la rappresentazione.

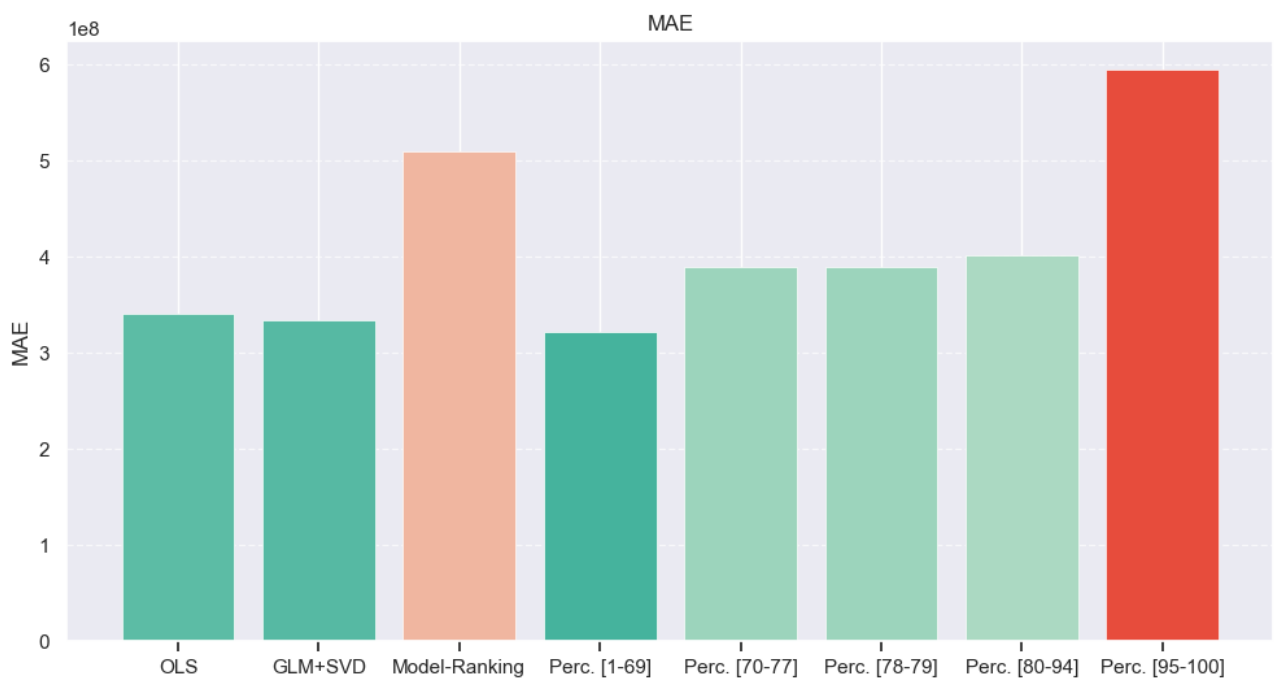
Nel caso dell'MSE, OLS è la metodologia che performa meglio, seguito da GLM+SVD e dai primi quattro range percentili. I percentili più estremi dell'ensemble selection performano decisamente male, seguito dal model ranking:

Figura 35: MSE per ciascun metodo



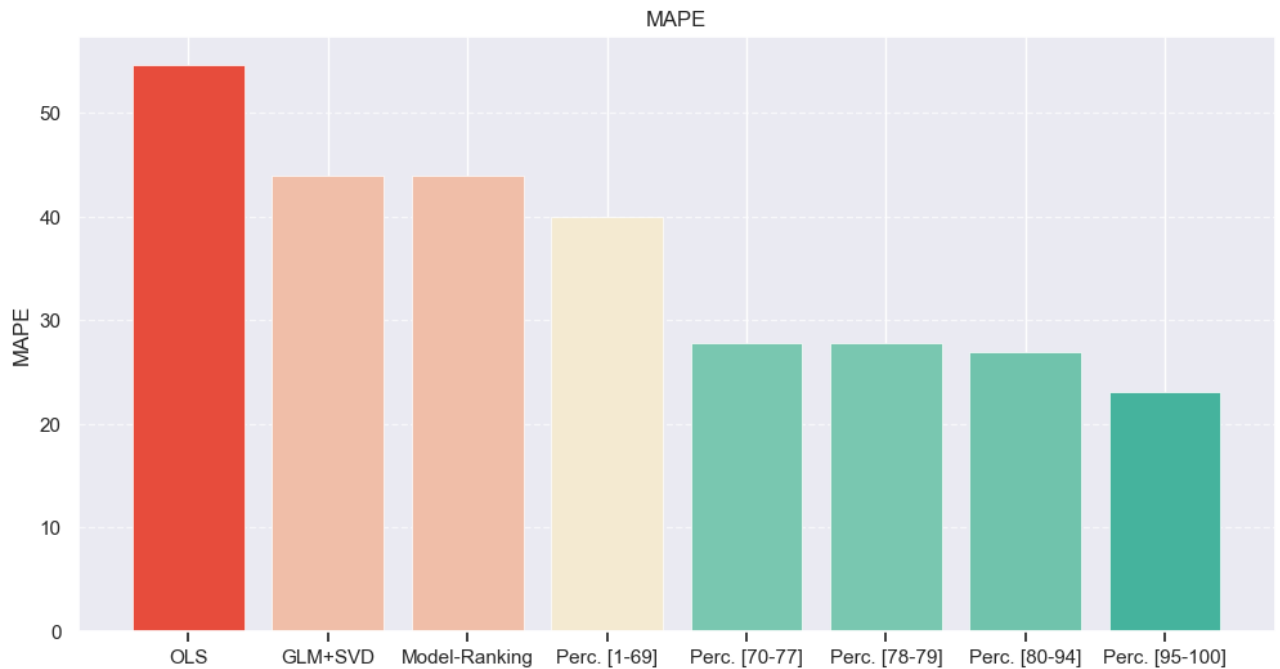
Per il MAE si osservano risultati simili a quelli di MSE per le metodologie peggiori, mentre la metodologia più performante risulta essere l'ensemble selection per il range percentile [1; 69]. I modelli lineari OLS e GLM+SVD performano leggermente peggio, seguito dal Model Ranking e dal range [95; 100]:

Figura 36: MAE per ciascun metodo



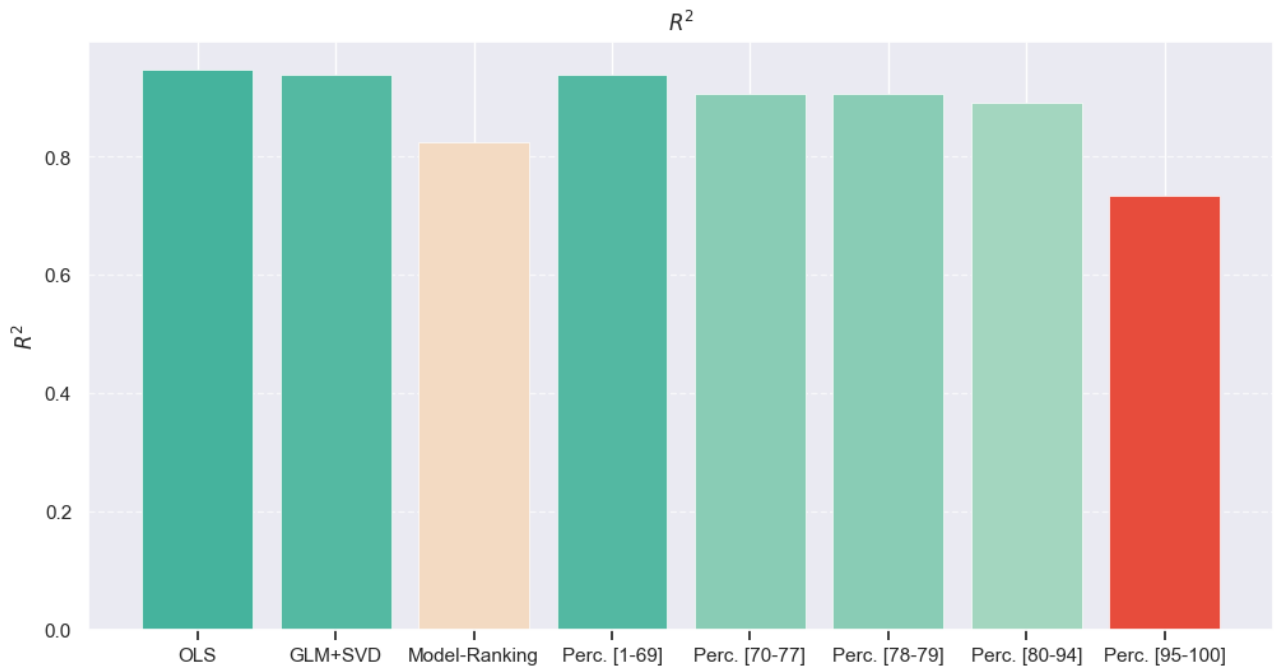
Con il MAPE si osservano comportamenti opposti rispetto a MSE e MAE. OLS performa peggio rispetto alle altre metodologie, seguito da GLM+SVD, Model-Ranking e model selection sul range percentile [1; 69]. Gli altri range percentili performano meglio:

Figura 37: MAPE per ciascun metodo



I modelli puramente lineari, rappresentati da OLS e GLM+SVD e dall'ensemble selection per il range percentile [1; 69], il quale ha un GLM+SVD nel modulo Life, tendono a massimizzare la metrica R^2 . I range percentili intermedi performano comunque bene, mentre il Model Ranking e il range percentile [95; 100] si posizionano rispettivamente al penultimo e ultimo posto della classifica.

Figura 38: R^2 per ciascun metodo



Complessivamente emerge come la metodologia di ensemble selection fornisca un bilanciamento tra metriche per il range percentile [70; 77]. In particolare, riesce a contenere il MAPE, sacrificando solo in parte le altre metriche. Il Model-Ranking, inteso come l'ensemble costituito dal miglior modello di ciascun modulo, non è sufficiente a migliorare le prestazioni di OLS e GLM+SVD, molto probabilmente in virtù delle considerazioni esposte nel paragrafo precedente relative alla mancanza di una visione congiunta delle cinque variabili target. È interessante osservare come GLM+SVD riesca a mitigare leggermente il MAPE rispetto a OLS, molto probabilmente per le dinamiche di regolarizzazione indiretta introdotte dall'algoritmo SVD.

5. CONCLUSIONI

Negli ultimi anni, il settore assicurativo ha vissuto una trasformazione significativa nella gestione del rischio, fortemente influenzata dall'introduzione della normativa Solvency II. Tale regolamentazione ha fornito un quadro strutturato e standardizzato per il calcolo del requisito patrimoniale, introducendo indicatori chiave come il Solvency Ratio. Parallelamente, si è assistito ad un rapido avanzamento delle tecniche di machine learning consentendo di integrare approcci data-driven nei processi di controllo e monitoraggio del rischio.

Il presente lavoro si è posto l'obiettivo di sviluppare uno strumento di screening preliminare capace di supportare principalmente i revisori e le autorità di vigilanza nell'identificazione delle imprese il cui Solvency Ratio si discosta in modo significativo dalle attese del modello calibrato. Questo obiettivo è stato perseguito mediante un approccio metodologico progressivo, articolato in due fasi principali: una iniziale basata su indicatori aggregati e la definizione di intervalli predittivi, e una successiva in cui sono stati impiegati dati più dettagliati e tecniche avanzate di machine learning con l'obiettivo di individuare il modello maggiormente performante.

Nella fase iniziale, come illustrato nel paragrafo 3, la costruzione di un numero limitato di indicatori rispetto alla base informativa a disposizione, ha permesso di stimare il Solvency Ratio per ciascuna compagnia e definire degli intervalli predittivi, con l'obiettivo di quantificare l'incertezza associata alle stime prodotte dai modelli utilizzati. Tale approccio consente di costruire uno strumento di screening efficace principalmente per il controllo interno ed esterno della compagnia. Pertanto al valore metodologico, lo studio offre anche dei risvolti operativi. Ad esempio, per le compagnie assicurative, lo strumento sviluppato potrebbe supportarle nell'ORSA e nella pianificazione strategica, mentre per i revisori e l'autorità di vigilanza, gli intervalli predittivi possono rappresentare un mezzo oggettivo per selezionare le imprese da sottoporre a controlli più approfonditi, ottimizzando le risorse e rendendo più efficace il processo di supervisione.

Nella seconda fase, la combinazione di modelli differenti, addestrati su sottoinsiemi dei dati (per ciascun modulo di rischio) o con algoritmi diversi, ha consentito di cogliere aspetti complementari della complessità informativa del dataset, migliorando la capacità del modello di identificare deviazioni significative del SCR rispetto ai valori attesi.

In particolare, come illustrato nel paragrafo 4, l'applicazione del ranking dei modelli e della selezione degli ensemble ha mostrato come sia possibile ridurre il numero complessivo di modelli mantenendo le performance predittive ottimali, ottimizzando così anche l'efficienza computazionale.

Inoltre, come evidenziato nel caso presentato, ossia in un processo di verifica, tali tecniche possono essere impiegate in alternativa alle valutazioni che solitamente vengono affidate all'expert judgment, fornendo un contributo significativo al processo di innovazione che le società stanno vivendo in un contesto data-driven.

BIBLIOGRAFIA

- **Archetti, Candeliere (2019)** Bayesian Optimization and Data Science. *Springer*
- **Bergstra e Bengio (2012)**. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research* 13 (2012)
- **Bonfatti, Tiezzi, Miglior, Carnier (2017)**. Comparison of Bayesian regression models and partial least squares regression for the development of infrared prediction equations. *Journal of Dairy Science. Volume 100, Issue 9*
- **Breiman (2001)**. Random Forests. *Machine Learning. Volume 45, pages 5-32. Springer*
- **Castellani, Fiore, Marino, Passalacqua, Perla, Scognamiglio e Zanetti (2018)**. An Investigation of Machine Learning Approaches in the Solvency II Valuation. *Available at SSRN: <https://ssrn.com/abstract=3303296>*
- **Chen, Guestrin (2016)**. XGBoost: A Scalable Tree Boosting System. *Available at: [arXiv:1603.02754](https://arxiv.org/abs/1603.02754)*
- **Commissione Europea (2014)**. Regolamento delegato (UE) 2015/35 della Commissione del 10 ottobre 2014. *Gazzetta ufficiale dell'Unione Europea*
- **Commissione Europea (2015)**. Regolamento delegato (UE) 2015/13 della commissione del 22 novembre 2015. *Gazzetta ufficiale dell'Unione Europea*
- **Commissione Europea (2015)**. Regolamento delegato (UE) 2016/25 della commissione del 26 luglio 2016. *Gazzetta ufficiale dell'Unione Europea*
- **Cover, Hart (1967)**. Nearest neighbor pattern classification. *Transactions on Information Theory (Volume: 13)*
- **Flaig e Junike (2021)**. Scenario Generation for Market Risk Models Using Generative Neural Networks. *Risks, 10(11), 199*
- **Géron, A. (2023)**. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. *O'Reilly Media (ISBN 978-1-098-12597-4)*
- **Hastie, Tibshirani, Friedman (2009)**. The Elements of Statistical Learning - Data Mining, Inference, and Prediction, Second Edition. *Springer*
- **Hejazi e Jackson (2016)**. Efficient Valuation of SCR via a Neural Network Approach. *Available at: [arXiv:1610.01946](https://arxiv.org/abs/1610.01946)*
- **Hoerl, Kennard (1970)**. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics Vol. 12, No. 1*
- **James, Witten, Hastie, Tibshirani, Taylor (2023)**. An Introduction to Statistical Learning with Applications in Python. *Springer*

- **Ke, Meng, Finley, Wang, Chen, Ma, Ye, Liu (2017).** LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems* 30
- **Krah, A.-S., Nikolić, I. e Korn, R. (2020).** Least-Squares Monte Carlo for Proxy Modeling in Life Insurance: Neural Networks. *Risks*, 8(4), 116
- **Latpate, Kshirsagar, Gupta , Chandra (2021).** Advanced Sampling Methods. *Springer*
- **Loisel, Piette e Tsai (2019).** Applying economic measures to lapse risk management with machine learning approaches. *Avaiable at: [arXiv.1906.05087](https://arxiv.org/abs/1906.05087)*
- **Raiffa, Schlaifer (1961).** Applied Statistical Decision Theory, *Harvard University, Boston, Chapters 11 and 13*
- **Rumelhart, Hinton, Williams (1986).** Learning internal representations by error propagation. *Avaiable at: <https://doi.org/10.7551/mitpress/4943.003.0128>*
- **Schölkopf, Smola, Williamson, Bartlett (2000).** New support vector algorithms. *Neural Computation*, 12
- **Sijie, Sia, Alfred e Moung (2024).** Ensemble Machine Learning Method for Health Insurance Premium Prediction. *Available at: [10.1109/IICAIET62352.2024.10730721](https://doi.org/10.1109/IICAIET62352.2024.10730721)*
- **Snoek, Larochelle, Adams (2012).** Practical Bayesian Optimization of Machine Learning Algorithms. *Avaiable at: [arXiv:1206.2944](https://arxiv.org/abs/1206.2944)*
- **Steinwart, Christmann (2008).** Support Vector Machines. *Springer*
- **Tibshirani (1996).** Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1), 267–288
- **Zollanvari, A. (2023).** Machine Learning with Python. Theory and Implementation. *Springer (ISBN 978-3-031-33341-5)*
- **Zou, Hastie (2005).** Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology, Volume 67, Issue 2*, 301–320