UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI ECONOMIA "GIORGIO FUÀ"

Corso di Laurea Magistrale o Specialistica in Data Science per l'Economia e le Imprese

CLASSIFICATORI SAMPLING BASED PER EVENTI RARI: UN'APPLICAZIONE AGLI EARLY WARNING SYSTEMS PER CRISI FINAZIARIE

SAMPLING BASED CLASSIFIERS FOR RARE EVENTS: AN APPLICATION TO EARLY WARNING SYSTEMS FOR FINANCIAL CRISIS

Relatore:
Prof.ssa Claudia Pigini
Correlatore:
Prof. Domenico Potena

Tesi di Laurea di:
Riccardo Rocchi

Anno Accademico 2021 – 2022

A mamma, papà e Beatrice.
Alle difficoltà, alle lacrime, agli imprevisti.
Questo piccolo traguardo è per voi.
Vi amo.

Indice

Capitolo 1

Introduzione

La crisi finanziaria del 2007-2008 ha colpito le economie avanzate innescando una recessione economica mondiale con gravi perdite nei settori reale e finanziario. Questa crisi si è manifestata come una crisi bancaria sistemica [1] e ha attirato l'attenzione delle istituzioni nazionali e sovranazionali sui legami tra le fluttuazioni della moneta e del credito e l'insorgere di una crisi, con l'obiettivo di attenuare la propagazione di simili eventi.

A tal fine, diversi progetti di ricerca hanno sviluppato *Early Warning Systems* (EWS), sistemi di allerta precoce, che permettono di quantificare, studiando congiuntamente indicatori economici e finanziari, la probabilità di una crisi. Raccogliendo vari fattori considerati discriminanti del fenomeno, diversi ricercatori si sono occupati di modellare questi sistemi per prevedere l'avvento di crisi e studiarne l'effetto. La gran parte di questi lavori si

[1] Una crisi bancaria (sistemica) si verifica quando molte banche si trovano contemporaneamente in seri problemi di solvibilità o liquidità, o perché sono colpite tutte dallo stesso shock esterno o perché il fallimento di una banca o di un gruppo di banche si diffonde ad altre banche nel sistema. [Worldbank, 2016]

è focalizzata sull' analisi dell'impatto di tali fattori così da indivduarne le vulnerabilità finanziarie al fine di progettare politiche macroprudenziali.

Nonostante la gran parte dei ricercatori per descrivere i sistemi di allerta precoce abbia utilizzato principalmente modelli econometrici quali modelli *logit* e *probit*, negli ultimi anni sono stati pubblicati lavori di ricerca che si basano sull'utilizzo di tecniche di *Machine Learning* per la previsione di queste crisi. Sono stati quindi sviluppati studi che vanno oltre la comprensione di questi fenomeni e si concentrano principalmente sulla loro classificazione. Infatti la peculiarità delle crisi è che si tratta di eventi rari, cioè a bassa probabilità di avvenimento ma ad alto impatto economico. Ciò rende necessario lo sviluppo di modelli di classificazione accurati in grado di cogliere la previsione di questi eventi rari. Infatti i modelli *logit* e *probit* tendono a soffirre del *paradosso dell'accuratezza*, cioè ad avere buone performance di previsione della classe maggiortaria mentre spesso si predilige un modello in grado di cogliere gli eventi di classe minoritaria. Nel contesto di riferimento, lo sviluppo di questi modelli deve basarsi principalmente sul cogliere questi eventi rari e ridurre al minimo l'errore di classificazione della classe minoritaria, cioè ridurre il numero di falsi negativi.

Pertanto le tecniche di *Machine Learning* hanno avuto un significativo riscontro in termini di risultati grazie alla loro capacità, a differenza della regressione logistica, di cogliere relazioni non lineari nei dati. Come si evin-

ce da diversi articoli quali Casabianca et al. [2019] e Holopainen and Sarlin [2017], sembrerebbe che modelli di *Machine Learning* tendano a performare meglio dei modelli *logit/probit* in termini di metriche di valutazione delle performance quali *F-measure*, *Precision* e *Recall*. Altri studi come quello di Beutel et al. [2019] mostrano invece come i modelli *logit* tendono a generalizzare meglio dei modelli di *Machine Learning* in previsioni *out of sample*. Tuttavia non è possibile fare un confronto diretto tra i differenti approcci in quanto molto dipende dall'impostazione del problema, dalle variabili utilizzate, dagli intervalli temporali considerati (trimesti, quadrimestri, anni) e da come il modello viene sviluppato in fase di addestramento.

Nella letteratura relativa agli EWS si sta ancora studiando come migliorare la capacità previsiva di modelli tradizionali e di *Machine Learning* ma fino ad ora non sono mai state prese in considerazione tecniche di campionamento al fine di bilanciare gli eventi di crisi e non. Pertanto, il presente lavoro di tesi si pone l'obiettivo di proporre una nuova soluzione per la previsione di questi fenomeni tramite tecniche di *Machine Learning* applicate a metodologie di *over* e *undersampling* rispettivamente tramite *SMOTE* e *Clustering*. L'idea è quella di presentare un nuovo metodo nella letteratura sugli EWS cercando di superare quello che in gergo *Data Science* viene definito problema di *Class Imbalance* al fine di migliorare la performance previsiva.

Il testo è organizzato come segue: nel secondo capitolo viene fatta una

revisione della letteratura sugli EWS; nel terzo viene descritto il dataset e la metodologia utilizzata; nel quarto vengono riportati i risultati; nel quinto le conclusioni.

Capitolo 2

Letteratura sugli Early Warning Systems

L'avvento di crisi finaziarie/bancarie negli anni ha posto l'obiettivo di sviluppare dei sistemi di monitoraggio al fine di riuscire a prevenire eventuali impatti catastrofici. In merito, la letteratura ha proposto diversi modelli in grado di prevedere queste crisi ottenendo risultati differenti a seconda della soluzione proposta. L'obiettivo di questi modelli è appunto quello di migliorare la capacità predittiva nel captare uno scenario di crisi per poi poter effettuare un confronto diretto sulla base di alcune metriche, le quali permettono di valutare la capacità di previsione e generalizzazione dei modelli stessi.

Per quanto riguarda lo sviluppo di un EWS, esistono diversi approcci:

l'approccio a segnali, il *Binary Classification Tree* (BCT) ed il modello *logit/probit*. Alla fine degli anni Kaminsky et al. [1998] hanno proposto questo approccio a segnali, metodo non parametrico che studia il comportamento ex post di variabili macroeconomiche e verifica se gli indicatori seguono uno specifico pattern nei periodi che precedono le crisi rispetto a periodi normali. L'approccio a segnali stabilisce quindi un livello di soglia per ogni predittore di crisi e confronta il valore di ogni predittore con il suo livello di soglia: se il valore di un predittore supera tale livello, segnala l'inizio di una crisi entro 12-24 mesi. Tuttavia, con l'approccio a segnali ogni indicatore viene utilizzato in modo isolato e il modello non consente di aggregare i singoli segnali. La soluzione più semplice consiste nel contare il numero di indicatori anticipatori che segnalano una situazione di sofferenza. Il rischio però è che questa statistica non sia la scelta migliore perché l'economia può essere vulnerabile, ma molti degli indicatori potrebbero non segnalare congiuntamente che qualcosa non va (problema dei segnali contradditori).

Il BCT è uno strumento ad albero decisionale che grazie ad un criterio di *split* stabilisce una serie di regole a partire dalle variabili [Duttagupta and Cashin, 2008]. Questo è utile per analizzare le crisi bancarie, in quanto riconosce che una combinazione di vulnerabilità può essere più determinante nell'innescare le crisi piuttosto che il deterioramento di un unico fattore. Il modello riconosce inoltre che gli indicatori economici possono avere un

impatto non lineare sulla probabilità di crisi, in quanto qualsiasi aumento o diminuzione di un indicatore chiave non deve necessariamente scaturire in una crisi finanziaria, a meno che il valore dell'indicatore non superi una soglia identificata.

L'approccio più utilizzato nella letteratura degli EWS è quello parametrico dei modelli *logit* e *probit*, tool ampiamente utilizzati in microeconometria per stimare la probabilità di un evento. Quindi la probabilità che una crisi avvenga è stimata come funzione di alcuni predittori e dalla stima dei coefficienti e dei segni del modello è possibile recuperare le probabilità stimate delle crisi. Demirgüç-Kunt and Detragiache [1998] hanno adottato questi metodi su un grande campione di paesi sviluppati ed in via di sviluppo tra il 1980 ed il 1994 scoprendo che le determinanti principali di una crisi bancaria sono l'alta inflazione e la bassa crescita del PIL. L'utilizzo di modelli con outcome discreto sono stati quindi altamente impiegati in letteratura e nonostante non siano modelli macroeconomici strutturali, permettono un' interpretazione economica del rapporto che c'è tra la variabile dipendente ed i regressori attraverso i coefficienti ed i segni stimati. Demirgüç-Kunt and Detragiache [1998] dimostrano che i modelli logit sono sia utili per quanto riguarda l'individuazione dei fattori scatenanti le crisi bancarie sia in termini predittivi. Al fine di valutare il forecast dei modelli utilizzano in particolare l' AIC e la *classification accuracy in-sample* giungendo ad un risultato pari

al 70% delle crisi correttamente predette.

Antunes et al. [2018] hanno utilizzato modelli *probit* ed hanno scoperto che l'inserimento dei *lag* delle variabili consente di prevedere correttamente un numero di crisi maggiore rispetto al semplice modello *probit*. Altri studiosi come Caggiano et al. [2016] hanno utilizzato invece modelli *multinomial logit*. In questo caso quindi la variabile dipendente non è composta solo dagli eventi crisi e non crisi ma anche dall'anno successivo alla crisi. Caggiano et al. [2016] giungono alla conclusione che nei campioni in cui la durata media della crisi è relativamente lunga, il modello *multinomial logit*, che distingue esplicitamente tra il primo anno di crisi e gli anni successivi alla crisi, migliora rispetto ai modelli *logit* binomiali più comunemente utilizzati. Altri modelli econometrici sono stati utilizzati in questo campo. Schularick and Taylor [2012] hanno sviluppato due tipologie di modelli: un modello *pooled logit* e un modello *logit* ad effetti fissi. Gli autori evidenziano come il modello ad effetti fissi permetta di ottenere prestazioni di *forecast* migliori rispetto al *pooled logit* nonostante questo continui ad essere quello più utilizzato per alcune considerazioni econometriche.

Oltre a questi modelli econometrici, anche il *Machine Learning* si è insidiato prepotentemente nella letteratura sugli EWS. Tuttavia a differenza dei modelli econometrici, le tecniche di *Machine Learning* perdono la capacità interpretativa tipica dei modelli tradizionali. D'altra parte la loro capaci-

tà di catturare relazioni non lineari permette di migliorare le previsioni *out of sample*. La gran parte della letteratura sugli EWS basata sul *Machine Learning* utilizza 2 approcci per addestrare i modelli:

1. *K-fold cross validation*: il dataset viene suddiviso in *k-fold e k-1 fold* vengono utilizzate per l'addestramento mentre la restante per testare il modello. Ciò permette di capire qual è l'errore di generalizzazione del nostro modello;

2. *Recursive/Time series cross validation*: in questa procedura, vi è una serie di *fold* di test, ciascuno dei quali è costituito da una singola osservazione. Il dataset di addestramento corrispondente è costituito solo da osservazioni precedenti a quella che costituisce la *fold* di test e, pertanto, nessuna osservazione al tempo t+1 può essere utilizzata per costruire la previsione. Hyndman and Athanasopoulos [2018]

Casabianca et al. [2019] hanno mostrato le differenze predittive tra modelli *logit* e tecniche di *Machine Learning*. Studiando 55 crisi bancarie per 33 economie avanzate e 87 crisi per 67 economie emergenti, hanno effettuato l'analisi dividento gli eventi in 2 gruppi: paesi emergenti e paesi avanzati. Dai risultati si evince come in termini di previsione *out of sample* i *Random Forest* performano molto meglio dei modelli *logit* in quanto appunto riescono a delineare bordi non lineari. Ulteriori lavori come quello di Holopainen and

Sarlin [2017] mostrano come queste tecniche riescono a performare meglio e minimizzare appunto l'errore di classificazione della classe minoritaria. Beutel et al. [2019] dichiarano invece che dall'analisi di economie avanzate dal 1970-2016, i modelli *logit* siano i più stabili in termini di previsioni *out of sample* anche all'utilizzo di variabili e scenari differenti se addestrati tramite *recursive cross validation*. Infatti dalla loro analisi si evince che i metodi di *Machine Learning* tendano a funzionare bene *in-sample* e ad *overfittare out of sample*. Bluwstein et al. [2021] mostrano come quasi tutti gli algoritmi di *Machine Learning* testati performano meglio rispetto ai tradizionali modelli *pooled logit* sia nelle previsioni *out of sample cross validation* che *recursive forecasting* ed in particolare gli *Extremly Randomised Trees* e *Random Forest* risultano i migliori classificatori in termini di performance. Inoltre gli autori hanno provato a superare il problema della scarsa capacità interpretativa di questi modelli non parametrici proponendo un' applicazione al *Machine Learning* delle teorie proposte in Shapley [1953]. Tölö [2020] ha sviluppato modelli in grado di prevedere crisi finanziarie sistemiche con un anticipo da uno a cinque anni utilizzando reti neurali ricorrenti. A partire dalla letteratura precedente dalla quale è risultato che semplice architetture di reti neurali superano il tradizionale modello di regressione logistica nella previsione di crisi finanziarie, gli autori hanno dimostrato che tali previsioni possono essere notevolmente migliorate utilizzando le RNN-LSTM (*Long Short Term*

Memory) e RNN-GRU (*Gated RecurrentUnit*).

Tuttavia una dei più grandi problemi nella letteratura degli EWS è che il fenomeno delle crisi finanziarie è caratterizzato da un alto sbilanciamento delle classi, il quale può portare a problemi in fase di realizzazione del modello. La maggior parte dei paper sugli EWS utilizza come riferimento le curve ROC (*Receiver Operating Characteristic*), curve che mettono a confronto il tasso di falsi positivi con il tasso di veri positivi. Valutare modelli di previsione su crisi finanziarie tramite l'AUC (*Area Under the Curve)* e quindi con l'*Accuracy*, porta a risultati con forte *bias*. Infatti l'*Accuracy* è una metrica che dice quanti dati sono stati correttamente etichettati sul totale dei dati, ma avendo un dataset composto quasi completamente da una classe, ciò può portare ad una valutazione errata della qualità del modello.

Essendo le crisi finanziarie degli eventi rari ad altissimo impatto, potrebbe essere opportuno quindi osservare anche le *Precision-Recall Curve*, grafici che mostrano il *trade-off* tra *Precision* e *Recall* per differenti *threshold*, dove:

- *Precision* è il numero di previsioni di classi positive che appartengono effettivamente alla classe positiva.

- *Recall* è la percentuale delle previsioni positive corrette sul totale delle istanze positive.

Ciò ci permette di valutare modelli sulla base di differenti threshold in modo

tale da tenere in considerazione sia il *tasso di veri positivi* sia i *positivi predetti positivi*. Da queste due metriche è possibile poi calcolare la *F-Measure*, cioè la media armonica di *Precision* e *Recall* che varia tra 0 (peggiore) e 1 (migliore). In particolare, in caso di eventi rari e quindi fortissimo sbilanciamento delle classi è opportuno avere modelli che abbiano un guadagno in *Recall* piuttosto che in *Precision*, poiché l'effetto di prevedere una non crisi a fronte di una crisi può essere devastante per il sistema economico.

Il *Machine Learning* ed in generale le reti neurali possono dare un apporto fondamentale allo sviluppo di EWS. In particolare grazie alla capacità computazionale di cui oggi si dispone sarà possibile migliorare sempre più questi modelli in modo tale da riuscire ad agire in anticipo ed impostare politiche macroeconomiche di salvaguardia in caso di avvento di crisi finanziarie sistemiche.

Capitolo 3

Dati e metodologia

3.1 Il dataset

Il dataset utilizzato è stato costruito a partire dal contributo della letteratura sugli EWS, in particolare da Demirgüç-Kunt and Detragiache [1998], Davis and Karim [2008] e Caggiano et al. [2016]. I dati utilizzati sono ripresi dal lavoro di Pigini [2021] e sono disponibili pubblicamente come *International Financial Statistics* (IFS), pubblicate dal Fondo Monetario Internazionale, o come *World Development Indicators*(WDI) emessi dalla *World Bank*. La variabile dipendente descrive le crisi bancarie sistemiche e la sua definizione è tratta da Laeven and Valencia [2018], i quali affermano che questo evento si verifica se, in un determinato anno, si verificano segnali di sofferenza finanziaria nel sistema bancario (corse agli sportelli, perdite e liquidazioni

bancarie) e se ci sono stati interventi politici a seguito di perdite significative nel sistema bancario.

Il panel di dati considerato è di 129 paesi per una serie temporale che va dal 1984 al 2017 per un totale di 195 crisi bancarie sistemiche registrate come mostrato in tabella 3.1

Tabella 3.1: Crisi finanziarie per paese

Paese	Anno di crisi	Paese	Anno di crisi
ARG	1995, 2001, 2002, 2003	KEN	1985, 1992, 1993, 1994
BDI	1994, 1995, 1996, 1997, 1998	KGZ	1998, 1999
BGD	1987	KOR	1997, 1998
BGR	1996, 1997	KWT	1984, 1985
BOL	1994	LBN	1992, 1993
BRA	1998	MAR	1984
CAF	1995, 1996, 1997	MDA	2014, 2015, 2016
CHE	2008, 2009	MEX	1984, 1985, 1994, 1995, 1996
CHN	1998	MNG	2008, 2009
CMR	1995, 1996, 1997	MRT	1984
COG	1992, 1993, 1994	MYS	1997, 1998, 1999
COL	1998, 1999, 2000	NGA	1992, 1993, 1994, 1995, 2009, 2010, 2011, 2012
CRI	1987, 1988, 1989, 1990, 1991, 1994	NIC	1990, 1991, 1992, 1993, 2000, 2001
CZE	1997, 1998, 1999, 2000	NPL	1988
DOM	2003, 2004	PAN	1988, 1989
DZA	1990, 1991, 1992, 1993, 1994	PHL	1984, 1985, 1986, 1997, 1998, 1999, 2000, 2001
GBR	2007, 2008, 2009, 2010, 2011	PRY	1995
GNB	2014, 2015, 2016	SLE	1990, 1991, 1992, 1993, 1994
GUY	1993	SLV	1989, 1990
HRV	1998, 1999	SWE	1994, 1995
HTI	1996, 1997, 1998	SWZ	1995, 1996, 1997, 1998, 1999
HUN	2008, 2009, 2010, 2011, 2012	TCD	1992, 1993, 1994, 1995, 1996
IDN	1997, 1998, 1999, 2000, 2001	THA	1997, 1998, 1999, 2000
IND	1993	TUR	2000, 2001
ISL	2008, 2009, 2010, 2011, 2012	UKR	1998, 1999, 2008, 2009, 2010, 2014, 2015, 2016
JAM	1996, 1997, 1998	URY	1984, 1985, 2002, 2003, 2004, 2005
JOR	1989, 1990, 1991	USA	1988, 2007, 2008, 2009, 2010, 2011
JPN	1997, 1998, 1999, 2000, 2001	VEN	1994, 1995, 1996, 1997, 1998

Il dataset considerato è composto da 8 variabili:

- *Current crisis*: presenza o meno della crisi nell'anno di riferimento. *Current crisis* è una variabile binaria caratterizzata da un forte sbilanciamento verso la classe *non crisi*;

- *Real GDP growth*: il tasso di crescita economica reale, o tasso di crescita del PIL reale, misura la crescita economica, espressa dal prodotto interno lordo (PIL), da un periodo all'altro, depurato dall'inflazione o dalla deflazione;

- *Log GDP per capita*: è dato dalla divisione diretta del PIL totale per la popolazione;

- *GDP deflator growth*: il deflatore del PIL, detto anche deflatore implicito dei prezzi, è una misura dell'inflazione. È il rapporto tra il valore dei beni e servizi che un'economia produce in un determinato anno a prezzi correnti e quello dei prezzi prevalenti nell'anno di riferimento [Thehindu, 2018]

- *Real interest rate*: il tasso di interesse reale è il tasso di interesse sui prestiti che un investitore, risparmiatore o prestatore riceve (o si aspetta di ricevere) dopo aver tenuto conto dell'inflazione;

- *M2/Reserves*: rapporto tra M2 (misura dell'offerta di moneta che include contanti, depositi bancari e denaro vicino facilmente convertibile)

e riserve valutarie della Banca Centrale. Permette di cogliere la capacità del Paese di resistere a un'improvvisa interruzione e inversione degli afflussi di capitale. Come scritto da Caggiano et al. [2016], più alto è il valore di questa variabile, maggiore è la vulnerabilità ai deflussi di capitale e quindi la probabilità di incorrere in una crisi bancaria;

- *Credit to GDP growth*: tasso di crescita del rapporto tra credito privato interno reale e PIL. Il credito privato si riferisce alle risorse finanziarie fornite al settore privato da società finanziarie, ad esempio attraverso prestiti, acquisti di titoli non azionari, crediti commerciali e altri crediti, che stabiliscono un diritto di rimborso.

- *Growth of net foreign assets to GDP*: rapporto tra attività nette estere e PIL. Le attività estere nette (NFA) determinano se un paese è creditore o debitore misurando la differenza tra le sue attività e passività esterne.

Di seguito si riportano in Tabella 3.2 le statistiche descrittive del dataset:

Tabella 3.2: Statistiche descrittive

Statistiche	Mean	St. Dev.	Min	Median	Max
Current crisis	0.068	0.251	0	0	1
Real gdp growth	4.004	4.117	-13.823	4.153	18.180
Log GDP per capita	7.726	1.459	4.713	7.675	11.390
GDP deflator growth:	19.515	200.995	-31.566	5.930	6,261.240
Real interest rate	2.335	10.771	-46.111	2.124	66.152
M2/Reserves	14.766	32.919	0.029	3.966	144.997
Credit to GDP growth	15.577	27.807	-237.334	12.483	518.081
Growth of net foreign assets to GDP	0.018	0.228	-2.440	0.015	2.297

3.1.1 Analisi esplorativa

In questa sezione vengono illustrati i *boxplot* per ogni variabile del dataset in base alla presenza o meno di crisi. Essendo le distribuzioni di alcuni predittori molto asimmetriche, al fine di una migliore visualizzazione sono state tagliate (graficamente) le osservazioni molto distanti dal primo e terzo quartile.

In tabella 3.3, vengono riportate le mediane di ogni variabile per gruppo.

Tabella 3.3: Mediana delle variabili per gruppo.

Variabili	**Crisi**	**Non crisi**
Real gdp growth	2.0996	4.254
Log GDP per capita	7.513	7.687
GDP deflator growth:	9.778	5.784
Real interest rate	2.2290	2.108
M2/Reserves	4.3958	3.94402
Credit to GDP growth	14.348	12.404
Growth of net foreign assets to GDP	0.000000	0.01577

Nelle figure 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 e 3.7 vengono invece riportati i *boxplot* di tutte le variabili:

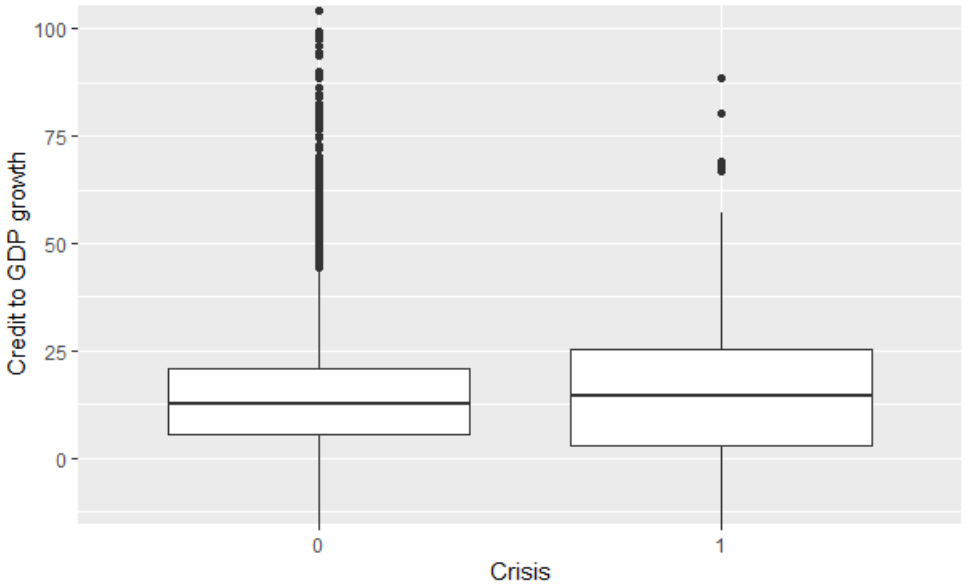

Figura 3.1: Crisi vs crescita del rapporto tra credito privato e PIL

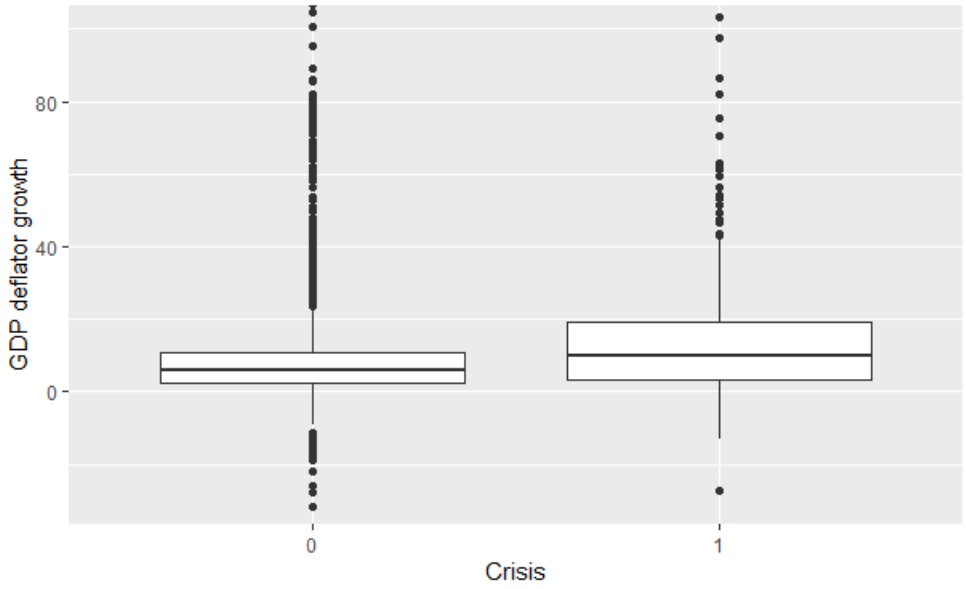

Figura 3.2: Crisi vs crescita del deflatore del PIL

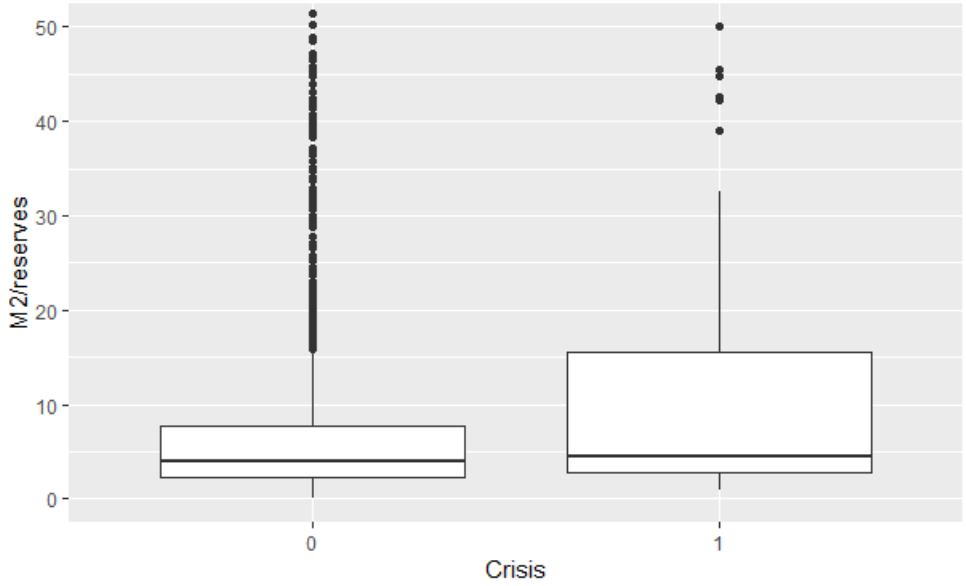

Figura 3.3: Crisi vs rapporto M2 su riserve valutarie

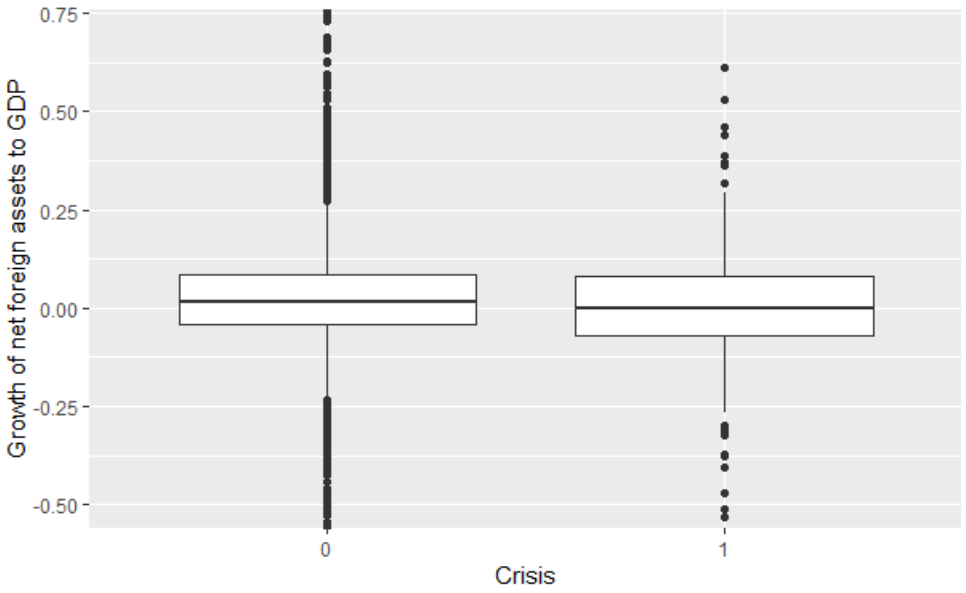

Figura 3.4: Crisi vs crescita del rapporto tra asset esteri e PIL

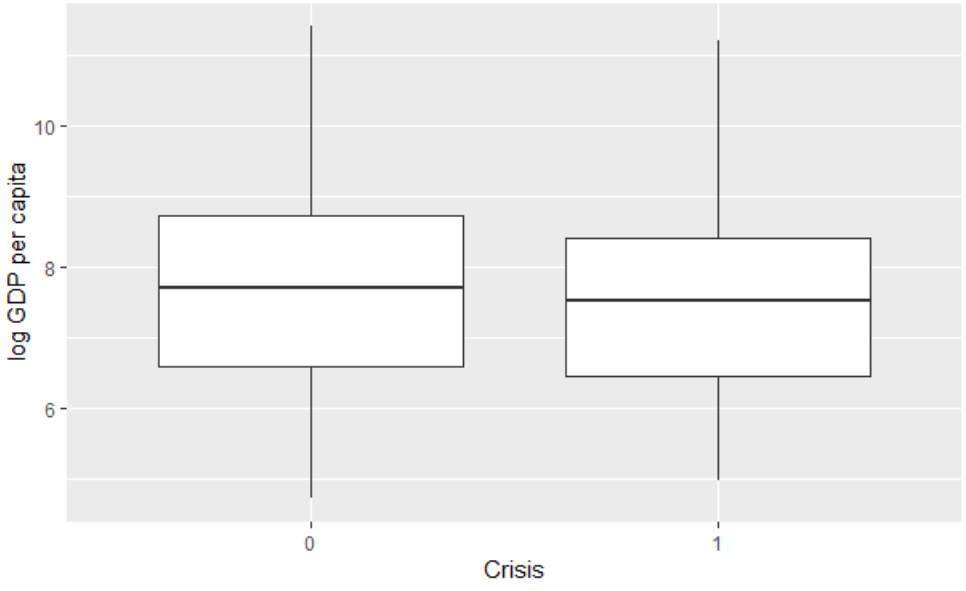

Figura 3.5: Crisi vs logaritmo del PIL pro capite

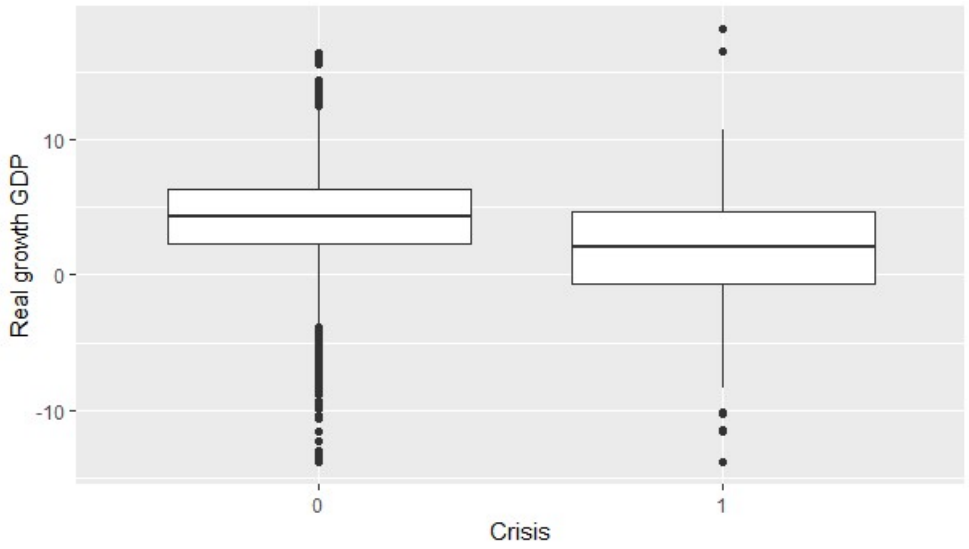

Figura 3.6: Crisi vs crecita reale del PIL

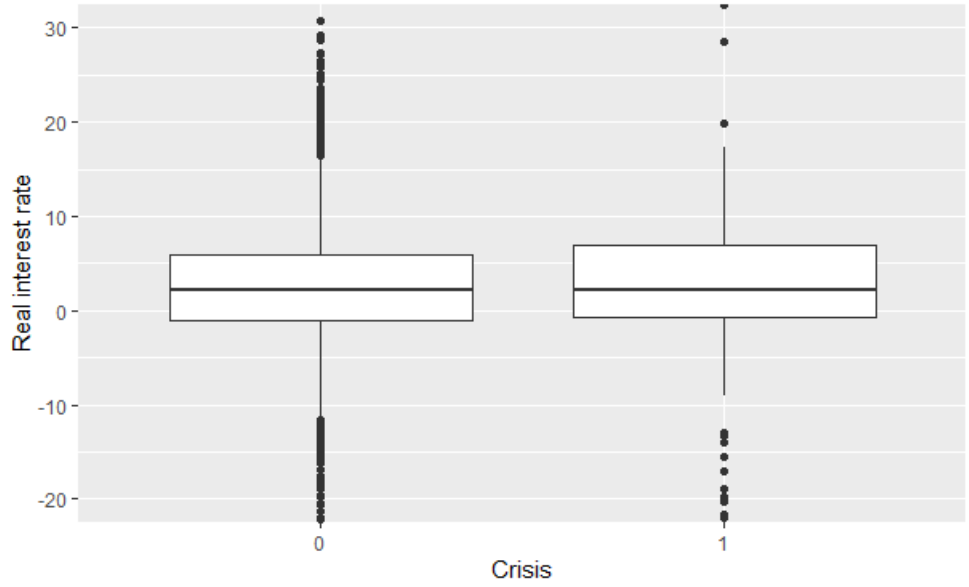

Figura 3.7: Crisi vs tasso di interesse reale

In particolare, come mostrato in figura 3.1 e figura 3.6, possiamo notare come *GDP deflator growth* e *Real growth GDP* assumano comportamenti

differenti a seconda della presenza o meno di crisi. Negli anni precedenti alla crisi, il deflatore del PIL è maggiore rispetto a periodi di non crisi con un valore mediano di 9.778 contro 5.784; al contrario, la crescita reale del PIL risulta minore negli anni precedenti alla crisi con un valore mediano di 2.0996 rispetto ai 4.254 in tempi normali.

3.1.2 Preprocessing del dataset

Una volta ripuliti i dati da quelle poche osservazioni con *missing values* e scalati i dati tramite trasformazione *min-max*, è stata effettuata una trasformazione nella struttura del dataset tramite codice Python.

L'idea, come mostrato in figura 3.9 è che a partire da un dataset formato wide, ogni paese in ogni anno viene trattato come un' osservazione indipendente. Ciò significa che per ogni osservazione (ad esempio USA 2001) vengono riportati i valori delle variabili all'anno t-1, la crisi al tempo t-1 e la crisi al tempo t. Questo è stato fatto al fine di addestrare un modello ad un lead temporale in avanti in modo tale da effettuare previsioni ad un anno.

Di seguito, figura 3.8, viene riportato il codice Python con la quale è stato trasformato il *panel* dataset in *cross section* dataset:

```python
import pandas as pd
df = pd.read_csv('ews_wide.csv')
df.drop(['imfcode', 'countryname'], axis=1, inplace=True)
df.set_index('isocode', inplace=True)
new_df = pd.DataFrame(columns=['isocode', 'A', 'B', 'C', 'D', 'E', 'F',
                               'G', 'H', 'I', 'year'])
num_columns = 8
start_year = 1984

for row in df.iterrows():
    year = start_year
    for i in range(0, len(row[1])-num_columns, num_columns):
        to_add = [row[0]] + row[1].iloc[i:i+num_columns+1].values.tolist() + [year]
        new_df.loc[len(new_df)] = to_add
        year += 1

new_df.to_csv('prova.csv', index=False)
```

Figura 3.8: Codice Python della trasformazione da *wide* dataset a struttura *cross section.*

In questo progetto di tesi è stato scelto di effettuare previsioni ad un anno ma è possibile poter allargare l'orizzonte temporale andando ad inserire ulteriori *features* a *lag* precedenti. Tuttavia aggiungere ulteriori predittori utilizzando lo stesso numero di dati può causare il fenomeno del *curse of dimensionality*, secondo il quale all'aumentare della dimensionalità, il volume dello spazio cresce così rapidamente che i dati disponibili diventano scarsi. Al fine di ottenere un risultato affidabile, la quantità di dati necessari deve crescere esponenzialmente con la dimensionalità [Tan et al., 2016].

L'idea di questa trasformazione è quella di trattare il nostro *panel* dataset come un *cross section* dataset e valutare la capacità di generalizzazione dei modelli tramite *K-fold cross validation.*

Country	x1 t	x2 t	y t	x1 t+1	x2 t+1	y t+1	...	x1 t+n	x2 t+n	y t+n
USA										
IT										
. . .										
GER										

COUNTRY	x1 t	x2 t	y t	y t+1
USA t				
USA t+1				
. . .				
USA t + n				

Figura 3.9: Sopra viene mostrato il dataset in formato *wide*; sotto la struttura trasformata.

3.2 Metodologia

3.2.1 Tecniche di campionamento

Al fine di affrontare il problema della *class imbalance* si è deciso di adottare tecniche di *undersampling* ed *oversampling* in modo tale da addestrare un classificatore su una porzione di dati bilanciata. Le tecniche utilizzate sono due:

1. *K-Means*: algoritmo di apprendimento non supervisionato che raggruppa i dati cercando di separare le osservazioni in n gruppi di uguale varianza, minimizzando un certo criterio. In questo caso è stato scelto come criterio l'indice di **Davies-Bouldin** [Rhys, 2020], metrica che calcola la varianza *intracluster* e la distanza tra i centroidi di ogni *cluster*. Per ogni cluster viene poi identificato il *cluster* più vicino e la somma delle loro varianze *intracluster* viene divisa per la differenza tra i loro centroidi. Questo valore viene calcolato per ogni *cluster* e l'indice *Davies-Bouldin* è la media di questi valori. Più il valore di questo indice è basso, migliore è la *clusterizzazione*. Il *K-means* inoltre ha bisogno di utilizzare un metodo di inizializzazione dei centroidi. Nel nostro caso è stato scelto il *K-means++*, il quale, rispetto alla selezione casuale dei centroidi, richiede meno iterazioni e ha più elevate possibilità di trovare

l'ottimo globale. Quindi, l'algoritmo funziona come segue [Tan et al., 2016]:

(a) Seleziona K punti come centroidi iniziali

(b) Ripete:

i. Forma K *clusters* assegnando tutti i punti al centroide più vicino.

ii. Ricalcola il centroide di ogni *cluster*.

(c) Fino a quando i centroidi non cambiano.

2. *SMOTE* (*Synthetic Minority Oversampling Technique*): questo algoritmo [Brownlee, 2020] lavora selezionando istanze vicine nello spazio delle *features*, "disegnando" una linea tra i punti nello spazio delle features ed estraendo un nuovo punto lungo questa linea. Inizialmente viene scelto un esempio casuale dalla classe minoritaria, vengono trovati K vicini più vicini a quell'istanza, si sceglie un vicino a caso e si crea un esempio sintetico in un punto casuale tra le due istanze nello spazio delle *features*.

Il *K-Means* è stato utilizzato per fare *undersampling* in modo tale da non buttare via troppa informazione rilevante ma farlo su base *clustering*. Come

mostrato in tabella 3.4, sulla base dell'indice *Davies-Bouldin* e della misura di distanza utilizzata, 2 sono i gruppi ottimali per il *clustering*.

Per quanto riguarda invece lo *SMOTE*, per il numero di K è stata scelta una griglia di valori che va da 2 a 10. Tuttavia, oltre al numero di vicini più vicini da scegliere per la creazione dell'istanza artificiale, nello *SMOTE* bisogna anche scegliere il numero di istanze artificiali da creare. Per questo motivo, si è deciso di creare 9 dataset differenti (escluso il dataset originale) in modo tale da poter sviluppare modelli per ognuno di essi e fare un confronto finale sulle performance. Tramite il software RapidMiner, sono stati creati infatti 9 dataset differenti in cui vengono applicati contemporaneamente *K-Means* (per *undersampling*) e *SMOTE* (*oversampling*) in diverse proporzioni ma con l'obiettivo di avere sempre dataset composti dal 50% di osservazioni della classe minoritaria e un 50% della classe maggioritaria.

3.2.2 Struttura dei dataset

La tabella 3.5 mostra la struttura dei dataset di training originati. Tra questi non figura il dataset di training originario, composto invece da 2599 osservazioni di cui 2438 di classe 0 (non crisi) e 161 di classe 1 (crisi). Il test set è invece composto da 289 osservazioni di cui 271 di classe 0 e 18 di classe 1 e mantiene per cui lo sbilanciamento dei dati originari che è di circa di 19:1.

Tabella 3.4: Numeri ottimali di gruppi sulla base dell'indice *Davies-Bouldin*

Iterazione	**k**	**Misura di distanza**	**Davies-Bouldin**
1	2	MixedMeasures	0.653252534
10	2	NumericalMeasures	0.653252534
19	2	BregmanDivergences	0.653252534
20	3	BregmanDivergences	0.897613188
2	3	MixedMeasures	0.897618474
11	3	NumericalMeasures	0.897618474
27	10	BregmanDivergences	1.027796536
24	7	BregmanDivergences	1.034062015
18	10	NumericalMeasures	1.03924125
23	6	BregmanDivergences	1.050615365
9	10	MixedMeasures	1.059642146
3	4	MixedMeasures	1.060921997
12	4	NumericalMeasures	1.06144986
5	6	MixedMeasures	1.063133132
26	9	BregmanDivergences	1.064730824
17	9	NumericalMeasures	1.075527903
16	8	NumericalMeasures	1.086349501
14	6	NumericalMeasures	1.088320205
4	5	MixedMeasures	1.090193158
25	8	BregmanDivergences	1.097671867
13	5	NumericalMeasures	1.098782265
22	5	BregmanDivergences	1.110900787
6	7	MixedMeasures	1.133909738
7	8	MixedMeasures	1.142520262
8	9	MixedMeasures	1.153310517
21	4	BregmanDivergences	1.15497724
15	7	NumericalMeasures	1.160617052

Tabella 3.5: Struttura dei dataset utilizzati per gli esperimenti

Dataset	**Classe 0**	**Classe 1**	**Tot. osservazioni**
1	360	360	720
2	460	460	920
3	560	560	1120
4	660	660	1320
5	760	760	1520
6	860	860	1720
7	1160	1160	2320
8	210	210	420
9	260	260	520

3.2.3 Classificatori

In questa sezione vengono illustrate le tecniche statistiche e di *Machine Learning* (e le griglie di valori scelte per gli iperparametri) utilizzate al fine di valutare la capacità di generalizzazione della previsione di crisi finanziarie.

Decision tree

Gli alberi decisionali sono modelli non parametrici che possono essere utilizzati sia per la classificazione che per la regressione. Sono composti da:

- Nodo radice: nodo che dà inizio al grafo. Valuta la variabile che meglio *splitta* i dati;
- Nodi intermedi: nodi in cui vengono valutate le variabili
- Nodi foglia: nodi finali in cui vengono effettuate le previsioni

Gli alberi decisionali [Datascience, 2020b] vengono costruiti dividendo ricorsivamente i dati di training utilizzando le *features* che minimizzano un certo indice di impurità del nodo [1]. Ciò avviene valutando alcune metriche, come l'indice di Gini o l'entropia per gli alberi decisionali categorici, o l'errore quadratico medio per gli alberi di regressione. Quindi avremo nella parte alta dell'albero le variabili che meglio discriminano la nostra variabile dipendente perchè sono quelle che "garantiscono" maggiore impurità nei nodi. A seconda poi se le variabili sono discrete o continue la regola di split è differente: se la variabile è discreta vengono valutati tutti i possibili valori, ottenendo N metriche calcolate per ciascuna variabile; se la variabile è continua viene utilizzato come possibile *threshold* la media di due valori consecutivi (ordinati dal più basso al più alto).

Inoltre più l'albero è profondo e maggiore è il rischio che non riesca a generalizzare bene e quindi di *ovefittare*. Per questo vi è la possibilità di *potare* (*pruning*) l'albero e far si che vengano buttati via sottonodi dell'albero stesso.

La tabella 3.6 mostra la griglia degli iperaprametri scelta per i *Decision Trees*.

[1] l'impurità del nodo è una misura dell'omogeneità delle labels nel nodo.

Tabella 3.6: Iperparametri Decision Tree

Criterio di split	Information gain (IG) - Gain ratio (GR) - Gini index (GI) - Accuracy (ACC).
Profondità massima	3-4-5-6-8-9-11-14-17-20.
Pruning e pre-pruning	TRUE.

Random forest

Il *Random Forest* è un *ensemble learning algorithm* per regressione e classificazione che combina i risultati di molti algoritmi di *Machine Learning* per cercare di ottenere migliori performance. Da notare che il nome *forest* è dato dal fatto che combina tanti alberi decisionali, mentre *random* perchè un insieme casuali di dati è usato per il training di ogni *Decision Tree*.

Per ogni albero, ad ogni iterazione, solo $m<n$ *features* sono usate per scegliere il miglior criterio di split. La predizione è poi basata sulla combinazione dei risultati dei singoli modelli ottenuti dai *Decision Trees* (tipicamente tramite *majority voting*).

Come riportato da Tan et al. [2016], il Random Forest funziona come segue:

1. Per b=1 a B:

   (a) crea un campione *boostrap* $\boldsymbol{Z}$ di dimensione N dal set di dati di traning;

(b) genera una "foresta di alberi casuali" T_b dai dati *bootstrappati* ripetendo ricorsivamente i seguenti step, fino a che la dimensione minima del nodo n_{min} viene raggiunta:

i. seleziona m variabili random dal totale delle p variabili;

ii. sceglie la variabile/punto di split migliore tra le m;

iii. Divide il nodo in due nodi figli.

2. Produce l'insieme di alberi $\{T_b\}_1^B$.

Nel caso poi di un problema di classificazione, per fare una previsione su un nuovo punto x:

- sia $\hat{C}_b(\mathrm{x})$ la previsione della classe del b-esimo albero del random forest. Allora $\hat{C}_{rf}^B(\mathrm{x})$= *majority voting* $\hat{C}_b(\mathrm{x})_1^B$

Gli iperparametri di questo algoritmo sono gli stessi dei *Decision Trees*, con l'aggiunta però del numero di alberi casuali da generare come mostrato in tabella 3.7.

Regressione logistica

Essendo i modelli *logit e probit* quelli più utilizzati nella letteratura sugli EWS, si è deciso di adottare la regressione logistica per poter fare un confronto con le tecniche di *Machine Learning*.

Tabella 3.7: Iperparametri Random Forest

Criterio di split	Information gain (IG) - Gain ratio (GR) - Gini index (GI) - Accuracy (ACC).
Profondità massima	3-4-5-6-8-9-11-14-17-20.
Pruning e pre-pruning	TRUE.
Numero di alberi	10-33-55-78-100

La regressione logistica è un modello parametrico che stima la probabilità che si verifichi un evento, sulla base di un dato insieme di variabili indipendenti. Poiché l'output è una probabilità, la variabile dipendente varia tra 0 e 1. Nella regressione logistica, viene applicata una trasformazione *logit* alle probabilità, trasformazione nota anche come *log odds*, (logaritmo naturale delle probabilità), ovvero la probabilità di successo sulla probabilità di fallimento.

Essendo un modello parametrico, vi sono dei parametri da dover stimare e, nel caso della regressione logistica, questi vengono stimati tramite *Maximum Likelihood* (ML).

Support Vector Machines (SVM)

I *Support Vector Machines* [KDnuggets, 2016] sono metodi di apprendimento supervisionato che possono essere utilizzati sia per la classificazione che per la regressione.

Gli SVM si basano sull'idea di trovare un iperpiano che divide al meglio un insieme di dati in due o più classi (nel caso di classificazione) e per fare ciò fanno utilizzo dei *support vectors*. Questi sono i punti più vicini all'iperpiano, che, se rimossi, altererebbero la posizione dell'iperpiano separatore e, per questo motivo, possono essere considerati elementi fondamentali di un set di dati. Intuitivamente si può pensare a un iperpiano come ad una linea che separa linearmente e classifica un set di dati e, più i punti sono lontani dall'iperpiano, più siamo sicuri che siano stati classificati correttamente. Pertanto, vogliamo che i nostri punti di dati si trovino il più lontano possibile dall'iperpiano, pur rimanendo sul suo lato corretto. Quindi, una volta che si aggiungeranno nuovi dati, il lato dell'iperpiano su cui si trovano deciderà la classe che gli assegneremo. La distanza tra l'iperpiano e le osservazioni più vicine all'iperpiano è nota come margine. L'obiettivo è quindi scegliere un iperpiano con il massimo margine possibile tra l'iperpiano e qualsiasi punto del set di dati di training, per avere maggiori possibilità di generalizzazione nel classificare i nuovi dati.

Uno dei grandi vantaggi degli SVM è la possibilità di utilizzare differenti *kernel* per definire la funzione decisionale. Nel seguente progetto di tesi sono stati utilizzati 3 differenti tipologie di SVM:

- **linear SVM**: questa tipologia di SVM ha come iperparametro solo

il *cost parameter* C, parametro di regolarizzazione che rappresenta il costo di *misclassificazione* delle istanze di training. Se C è piccolo, si avrà una bassa penalità per i punti misclassificati e quindi viene scelto un bordo decisionale con un ampio margine. Se C è grande, SVM cerca di ridurre il numero di istanze *misclassificate* a causa di una penalità elevata che si traduce in un bordo decisionale con un margine inferiore;

- **radial basis function (RBF) SVM**: questo *kernel* non lineare oltre al parametro di regolarizzazione C, ha bisogno di definire anche il parametro *gamma*, parametro che controlla la distanza dell'influenza di un singolo punto di training. Valori bassi di *gamma* indicano un maggior numero di punti raggruppati, quindi un ampio "spazio" di somiglianza. Per valori elevati di *gamma*, i punti devono essere molto vicini tra loro per essere considerati della stessa classe [Datascience, 2020a].

- **polynomial SVM**: *kernel* che rappresenta la somiglianza delle istanze di training in uno spazio delle *features* sui polinomi delle variabili originarie. In questo caso, oltre al parametro C va definito anche il grado del polinomio da utilizzare.

Di seguito, la tabella 3.8 mostra i differenti iperparametri ed i loro range di valori per ogni tipologia di *kernel*.

Tabella 3.8: Differenti *kernel* degli SVM ed iperparametri

Linear SVM	**Cost parameter (C)**: 0-10-100-1000-10000.
Radial basis function SVM	**Cost parameter (C)**: 0-10-100-1000-10000; **Gamma**: 0.1-0.3-0.5-0.7-1
Polynomial SVM	**Cost parameter (C)**: 0-10-100-1000-10000; **Degree of the polynomial** : 2-3-4-5.

3.2.4 Workflow del processo

L'addestramento dei modelli, come già detto, è avvenuto tramite l'utilizzo di tecniche di *undesampling* ed *oversampling* con l'obiettivo di bilanciare le classi e creare differenti dataset così da poter condurre esperimenti vari. In figura 3.10 viene mostrato l'intero flusso di lavoro di training e testing dei modelli. Gli step seguiti sono i seguenti:

1. Il dataset viene diviso in base alla tipologia della classe;

2. sulla classe maggioritaria viene applicato il *K-Means* che divide i dati in due gruppi;

3. sulla base delle istanze artificiali da creare ed il vicino più vicino scelto per lo *SMOTE* nella classe minoritaria, la classe maggioritaria viene ridotta su base *clustering* in modo tale da avere un set di dati bilanciato per entrambe le classi. Viene poi fatto un *join* su base ID delle 2 classi per ottenere il dataset bilanciato;

4. in *cross validation* vengono addestrati i vari classificatori (*Decision Tree, Random Forest, SVM* e regressione logistica) con le diverse combinazioni di iperparametri;

5. vengono poi riportate ogni volta le metriche e le matrici di confusione. In questa maniera si può valutare le performance dei classificatori per ogni esperimento.

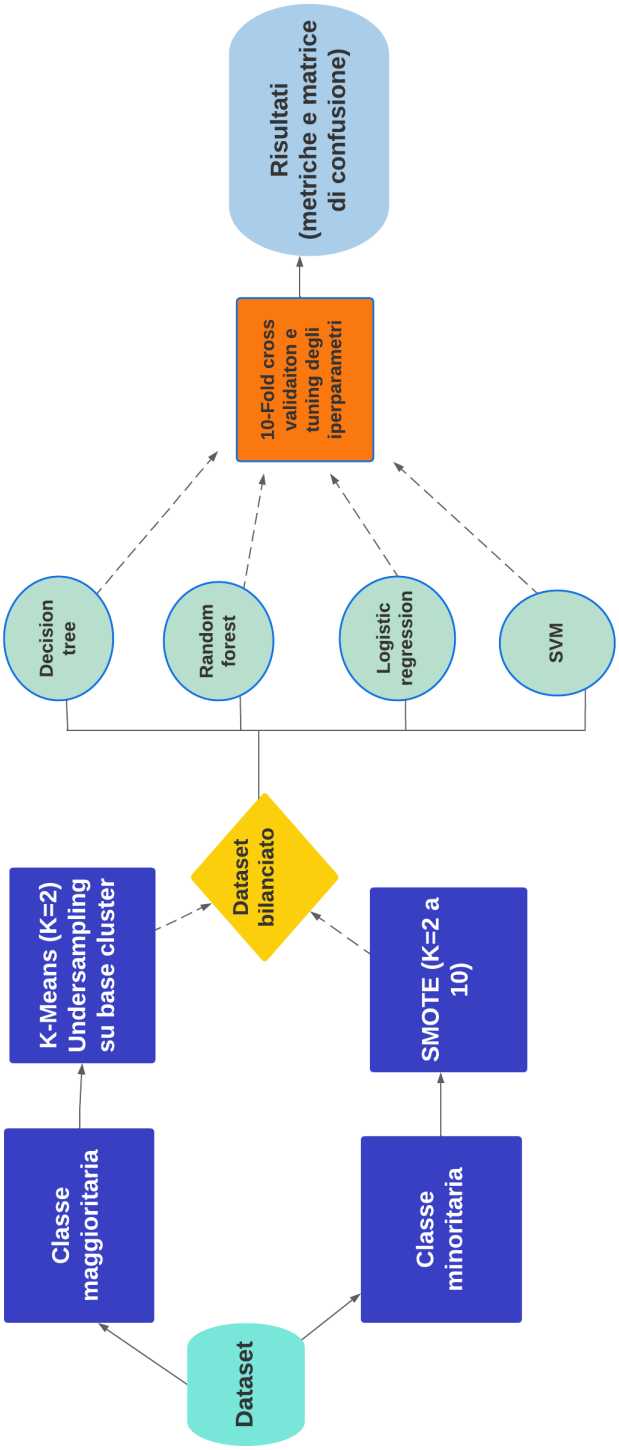

Figura 3.10: Workflow del processo

Capitolo 4

Risultati

In questo capitolo vengono illustrate le metriche di performance e le matrici di confusione per i classificatori in tutti gli esperimenti effettuati. Il primo esperimento (esperimento 0) confronta la capacità di generalizzazione dei classificatori sul dataset iniziale sbilanciato; gli altri 9 invece (3.5 mostrano le performance per ogni dataset creato a partire dalle configurazioni dell'*undersampling* e dello *SMOTE*.

L'obiettivo principale degli algoritmi di ricampionamento è quello di migliorare la *Recall* senza danneggiare la *Precision*. Tuttavia, gli obiettivi della *Recall* e della *Precision* possono essere spesso in conflitto, poiché quando si aumentano i veri positivi per la classe minoritaria, il numero di falsi positivi potrebbe aumentare andando a ridurre la *Precision* [Ding, 2011].

L'idea quindi è quella di vedere se effettivamente queste tecniche di bi-

lanciamento apportano o meno un miglioramento nelle performance in termini di *Recall* (in quanto l'obiettivo è minimizzare i falsi negativi) ma senza sottovalutare la *Precision* ed in generale la *F-Measure*

4.1 Legenda delle tabelle dei risultati

Per ogni esperimento vengono riportate le performance dei classificatori, la matrice di confusione, il numero di iterazione ed il set degli iperparamentri scelti come visto in tabelle 3.6, 3.7 e 3.8. Le notazioni sono le seguenti:

- Esperimento: indica il numero del dataset utilizzato come visto in tabella 3.5;
- Iterazione: indica il numero di iterazione al quale si è raggiunto quel set di performance a quel determinato livello di iperparametri;
- *True Positive* (TP): numero di crisi correttamente predette come crisi;
- *True Negative* (TN): numero di non crisi correttamente predette come non crisi;
- *False Negative* (FN): numero di crisi erratamente predette come non crisi;

- *False Positive* (FP): numero di non crisi erratamente predette come crisi;
- K: numero di *k-nearest-neighbour* considerati;
- *Precision*:

$$\frac{TP}{TP + FP}$$

- *Recall*:

$$\frac{TP}{TP + FN}$$

- *F-Measure*:

$$\frac{2 * Precision * Recall}{Precision + Recall}$$

- *Accuracy*:

$$\frac{TP + TN}{TP + TN + FN + FP}$$

4.2 Regressione logistica

Come mostrato in tabella 4.1 vediamo come la *Recall* tende a migliorare una volta effettuato l'*undersampling* e l'*oversampling*. In particolare negli esperimenti 8 e 9 si ha un buon guadagno in termini di *Recall*, ma dall'altra

parte una forte perdita in *Precision*. Per cui l'esperimento 7 (figura 4.1) sembra essere quello che riesce a dare un buon compromesso, in quanto perde circa di 6,4 punti percentuali in *Precision* ma ha un guadagno in *Recall* di 1,12 punti percentuali, per una *F-Measure* complessiva del 65,66% contro il 68.82% dell'esperimento 0. Questo sta a significare che il modello addestrato sul dataset bilanciato riesce ad identificare correttamente il 72,63% delle crisi contro il 71,51% del modello addestrato sul dataset sbilanciato.

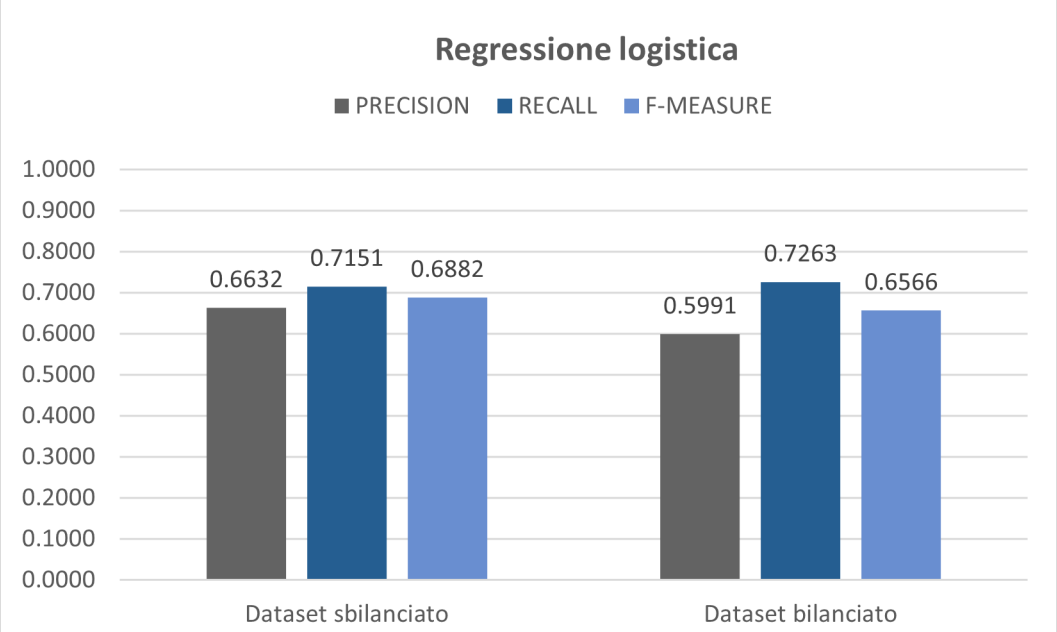

Figura 4.1: Metriche di performance della regressione logistica su dataset bilanciato (esperimento 7) e sbilanciato.

4.3 Decision tree

In tabella 4.2 si può vedere come i risultati dei vari esperimenti siano tutti uguali tra loro tranne che per l'esperimento 0 e l'esperimento 7. Il *Decision Tree* non sembrerebbe risentire di molto dello sbilanciamento e questo potrebbe essere dovuto al fatto che gran parte dei dati della classe minoritaria si trovi in un'area nello spazio delle *features*. Inoltre sembrerebbe che all'aumentare di molto delle istanze artificiali come nel caso 7, il classificatore tenda a perforomare leggermente peggio, con un decremento in termini di *Recall* pari a 0.6 punti percentuali.

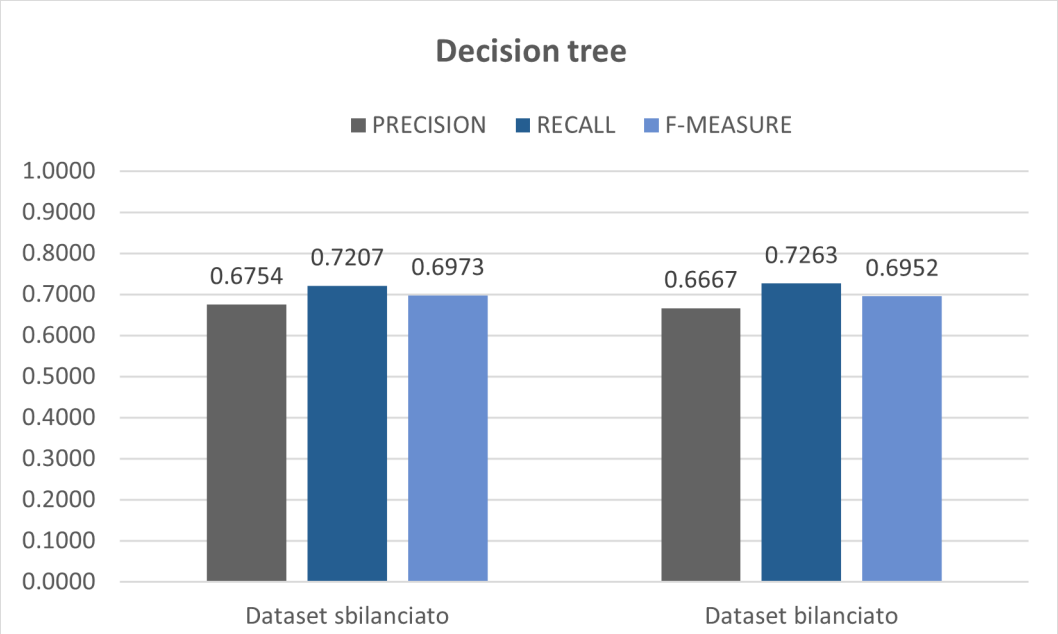

Figura 4.2: Metriche di performance del *Decision Tree* su dataset bilanciato (esperimento 6) e sbilanciato.

Quindi (figura 4.2), il classificatore addestrato su un dataset bilanciato riesce a predirre il 72,6% delle crisi contro il 72% del dataset sbilanciato a fronte di una perdita in *Precision* di circa 0.9 punti percentuali (66,67% contro 67,54%), per una *F-Measure* complessiva del 69,52% contro il 69,73% dell'esperimento 0.

4.4 Random forest

Come i *Decision Trees*, anche i *Random Forest* riescono ad avere ottime performance in termini di *Precision* e *Recall*. Infatti, come mostrato in tabella 4.3, negli esperimenti 6 e 8, i modelli sviluppati riescano a predirre fino al 73,18% delle crisi contro il 72,06% del modello sviluppato su dati sbilanciati (1,12 punti percentuali in più). Per cui i *Random Forest* (come confermato anche nella letteratura sugli EWS) si dimostrano ottimi classificatori che ci permettono di ottenere buoni risultati sulla previsione della classe minoritaria. I modelli (figura 4.3) raggiungono anche un 66,84% di *Precision* ed una *F-Measure* complessiva del 69,87%, leggermente superiore a quella dell'esperimento 0 pari al 69,54%.

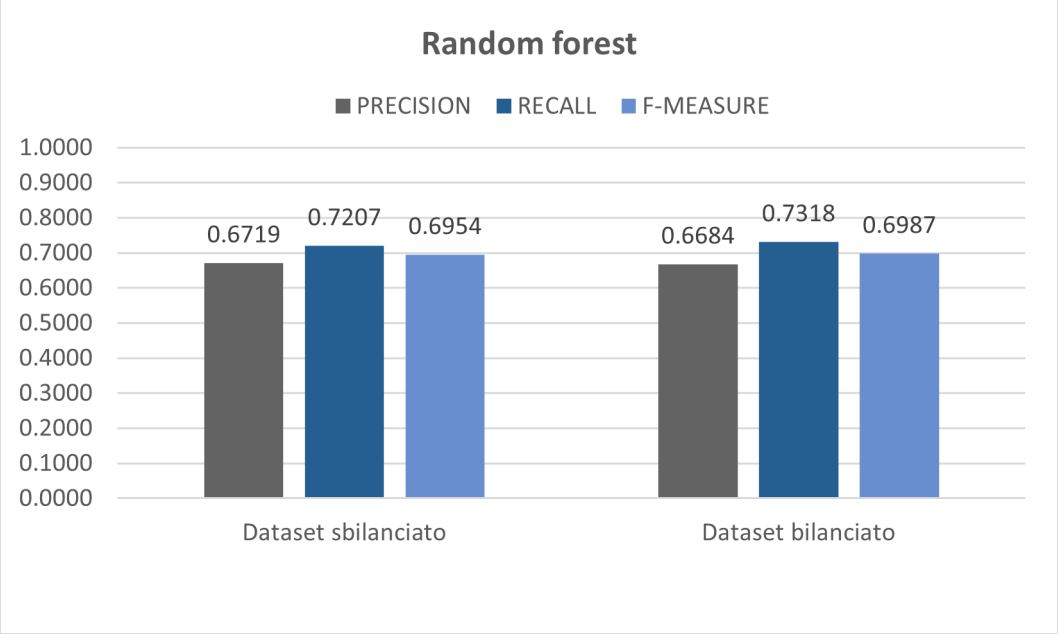

Figura 4.3: Metriche di performance del *Random Forest* su dataset bilanciato (esperimento 6-8) e sbilanciato.

4.5 SVM linear

Nel *SVM linear* (tabella 4.4) si ha un forte miglioramento in termini sia di *Precision* che di *Recall* rispetto all'esperimento 0. In particolare si può notare come negli esperimenti 1-2-3-4-5-7, il modello ha performance pari a 6,6 punti percentuali in più in termini di *Precision* e 7 punti percentuali in più in termini di *Recall*. Inoltre (figura 4.4), nell'esperimento 9, il modello arriva fino ad una *Recall* pari al 73,74% con una *Precision* del 65,67% ed una *F-measure* complessiva del 69,47% contro il 62,73% dell'esperimento 0,

Ciò significa che il modello riesce a prevedere correttamente il 73,74% delle

crisi, riducendo il numero di falsi negativi in media a 4.7. Questo risultato migliora anche il 73,18% delle crisi predette dal *Random Forest* e i suoi 4.8 falsi negativi previsti in media.

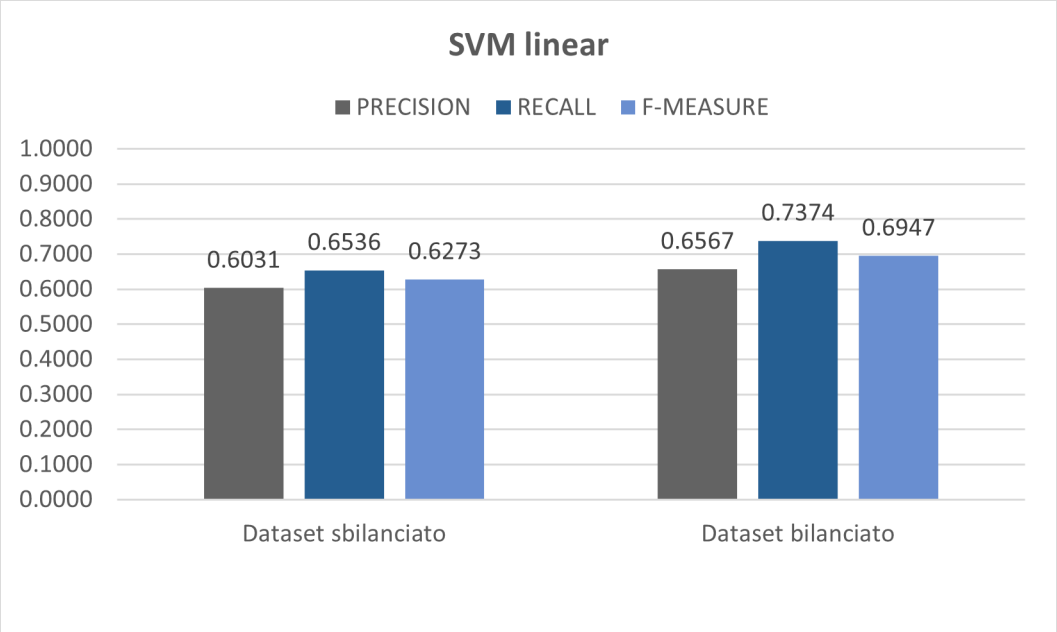

Figura 4.4: Metriche di performance del *SVM linear* su dataset bilanciato (esperimento 9) e sbilanciato.

4.6 SVM radial

Per quanto riguarda il *kernel radial* (tabella 4.5), in particoalre nell'esperimento 1 si può notare il forte miglioramento in termini di *Recall* rispetto all'esperimento 0 (73,74% contro 55,86%). Tuttavia si ha una forte diminuzione in termini di *Precision*, da 65,35 a 48,7%. Nonostante l'obiettivo sia

quello di minimizzare i falsi negativi, è necessario comunque avere un modello in grado di riuscire a sbagliare meno falsi positivi (cioè le non crisi predette crisi).

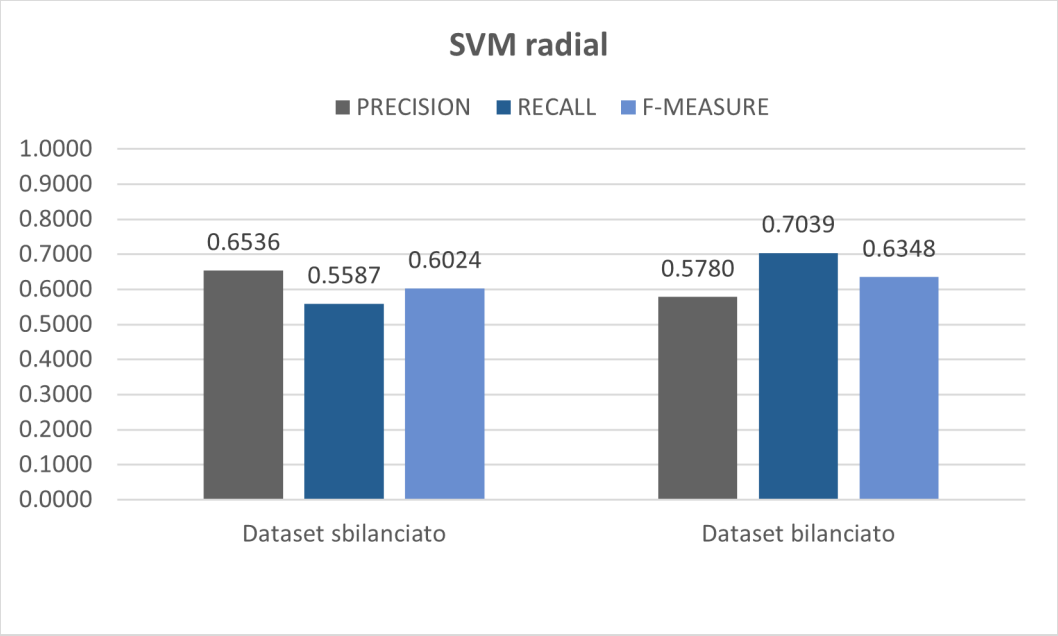

Figura 4.5: Metriche di performance del *SVM radial* su dataset bilanciato (esperimento 7) e sbilanciato.

Per cui l'esperimento 7 sembra riuscire a trovare un buon equilibrio tra falsi positivi e falsi negativi con una *Precision* del 57,8%, *Recall* del 70,39%, per una *F-Measure* complessiva pari al 63,48% contro il 60,24% del dataset sbilanciato (figura 4.5).

4.7 SVM polynomial

Nel seguente *Kernel* (tabella 4.6), gli esperimenti 1,2,4,9 presentano una *Recall* del 72,62% contro il 71,5% dell'esperimento 0 (1,12 punti percentuali in più). Tuttavia anche qui è necessario trovare un giusto *trade-off* tra *Precision* e *Recall*, in quanto in questi esperimenti la *Precision* tende a scendere di molto. L'esperimento 5 (figura 4.6) infatti sembra avere comunque buone performance in termini sia di *Precision* (60,56%) che *Recall* (72,067%), per una *F-Measure* complessiva del 65,82%.

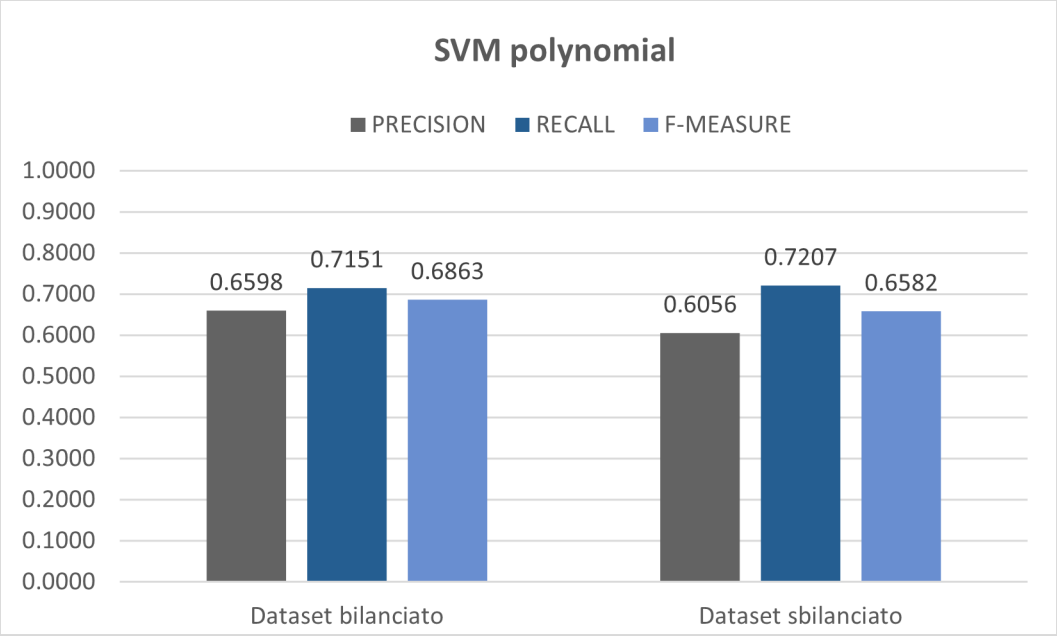

Figura 4.6: Metriche di performance del *SVM polynomial* su dataset bilanciato (esperimento 5) e sbilanciato.

Tabella 4.1: Configurazioni e metriche di performance della regressione logistica

Esperimenti	**Iterazione**	**K**	**FP**	**FN**	**TP**	**TN**	**Precision**	**Recall**	**F-Measure**	**Accuracy**
esp 0	1	//	6.5	5.1	12.8	264.4	0.6632	0.7151	0.6882	0.9598
esp 1	2	3	10.6	4.8	13.1	260.3	0.5527	0.7318	0.6298	0.9467
esp 2	1	2	9.8	4.8	13.1	261.1	0.5721	0.7318	0.6422	0.9494
esp 3	5	6	9.2	4.9	13	261.7	0.5856	0.7263	0.6484	0.9512
esp 4	5	6	9.1	4.9	13	261.8	0.5882	0.7263	0.6500	0.9515
esp 5	2	3	9.4	4.8	13.1	261.5	0.5822	0.7318	0.6485	0.9508
esp 6	2	3	9.5	4.9	13	261.4	0.5778	0.7263	0.6436	0.9501
esp 7	3	4	8.7	4.9	13	262.2	0.5991	0.7263	0.6566	0.9529
esp 8	4	5	13.3	4.7	13.2	257.6	0.4981	0.7374	0.5946	0.9377
esp 9	1	2	12.2	4.7	13.2	258.7	0.5197	0.7374	0.6097	0.9415

Tabella 4.2: Configurazioni e metriche di performance del *Decision Tree*

Esperimenti	**Iterazione**	**K**	**Criterio.**	**Max depth.**	**FP**	**FN**	**TP**	**TN**	**Precision.**	**Recall**	**F-Measure**	**Accuracy**
esp 0	1	//	GR	3	6.2	5	12.9	264.7	0.6754	0.7207	0.6973	0.9612
esp1	10	2	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 2	10	2	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 3	11	3	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 4	242	8	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 5	202	7	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 6	202	7	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 7	2	2	IG	3	6.5	5	12.9	264.4	0.6649	0.7207	0.6917	0.9602
esp 8	2	2	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 9	2	2	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605

Tabella 4.3: Configurazioni e metriche di performance del *Random Forest*

Esperim.	**Iterazione**	**K**	**N. Alberi**	**Criterio**	**Max depth**	**FP**	**FN**	**TP**	**TN**	**Precision**	**Recall**	**F-Measure**	**Accuracy**
esp 0	62	//	33	GR	6	6.3	5	12.9	264.6	0.6719	0.7207	0.6954	0.9609
esp 1	1013	77	55	GINI I.	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 2	1011	7	10	GINI I.	3	6.7	4.8	13.1	264.2	0.6616	0.7318	0.6950	0.9602
esp 3	1407	9	33	IG	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605
esp 4	57	2	33	ACC	5	6.6	4.8	13.1	264.3	0.6650	0.7318	0.6968	0.9605
esp 5	57	2	33	ACC	5	6.7	4.8	13.1	264.2	0.6616	0.7318	0.6950	0.9602
esp 6	57	2	33	ACC	5	6.5	4.8	13.1	264.4	0.6684	0.7318	0.6987	0.9609
esp 7	77	2	33	ACC	6	6.6	4.8	13.1	264.3	0.6650	0.7318	0.6968	0.9605
esp 8	12	2	33	GINI I.	3	6.5	4.8	13.1	264.4	0.6684	0.7318	0.6987	0.9609
esp 9	2	2	33	GR	3	6.5	4.9	13	264.4	0.6667	0.7263	0.6952	0.9605

Tabella 4.4: Configurazioni e metriche di performance del *SVM linear*

Esperimenti	**Iterazione**	**C**	**K**	**FP**	**FN**	**TP**	**TN**	**Precision**	**Recall**	**Accuracy**	**F-Measure**
esp 0	1	0	//	7.7	6.2	11.7	263.2	0.6031	0.6536	0.9519	0.6273
esp 1	1	0	2	6.5	4.9	13	264.4	0.6667	0.7263	0.9605	0.6952
esp 2	32	1	7	6.5	4.9	13	264.4	0.6667	0.7263	0.9605	0.6952
esp 3	4	100	2	6.5	4.9	13	264.4	0.6667	0.7263	0.9605	0.6952
esp 4	44	1	9	6.5	4.9	13	264.4	0.6667	0.7263	0.9605	0.6952
esp 5	7	0	3	6.5	4.9	13	264.4	0.6667	0.7263	0.9605	0.6952
esp 6	8	1	3	7.4	4.7	13.2	263.5	0.6408	0.7374	0.9581	0.6857
esp 7	1	0	2	6.5	4.9	13	264.4	0.6667	0.7263	0.9605	0.6952
esp 8	49	0	10	7	4.8	13.1	263.9	0.6517	0.7318	0.9591	0.6895
esp 9	32	1	7	6.9	4.7	13.2	264	0.6567	0.7374	0.9598	0.6947

Tabella 4.5: Configurazioni e metriche di performance del *SVM radial*

Esperimenti	**Iterazione**	**Gamma**	**C**	**K**	**FP**	**FN**	**TP**	**TN**	**Precision**	**Recall**	**Accuracy**	**F-Measure**
esp 0	1	0.1	0	//	5.3	7.9	10	265.6	0.6536	0.5587	0.9543	0.6024
esp 1	25	0.1	0	6	13.9	4.7	13.2	257	0.4871	0.7374	0.9356	0.5867
esp 2	121	0.1	0	6	12.4	5	12.9	258.5	0.5099	0.7207	0.9397	0.5972
esp 3	121	0.1	0	6	11.8	5.1	12.8	259.1	0.5203	0.7151	0.9415	0.6024
esp 4	211	0.1	0	9	12.4	5.1	12.8	258.5	0.5079	0.7151	0.9394	0.5940
esp 5	121	0.1	0	6	11.2	5.2	12.7	259.7	0.5314	0.7095	0.9432	0.6077
esp 6	121	0.1	0	6	10.6	5.2	12.7	260.3	0.5451	0.7095	0.9453	0.6165
esp 7	121	0.1	0	6	9.2	5.3	12.6	261.7	0.5780	0.7039	0.9498	0.6348
esp 8	161	0.1	10	7	14.4	5.9	12	256.5	0.4545	0.6704	0.9297	0.5418
esp 9	131	0.1	10	6	14.2	5.2	12.7	256.7	0.4721	0.7095	0.9328	0.5670

Tabella 4.6: Configurazioni e metriche di performance del *SVM polynomial*

Esperimenti	**Iterazione**	**C**	**K**	**Degree**	**FP**	**FN**	**TP**	**TN**	**Precision**	**Recall**	**F-Measure**	**Accuracy**
esp 0	2	1	//	2	6.6	5.1	12.8	264.3	0.6598	0.7151	0.6863	0.9595
esp 1	39	10	8	2	10.5	4.9	13	260.4	0.5532	0.7263	0.6280	0.9467
esp 2	39	10	8	2	10.5	4.9	13	260.4	0.5532	0.7263	0.6280	0.9467
esp 3	51	10	10	2	9.2	5	12.9	261.7	0.5837	0.7207	0.6450	0.9508
esp 4	32	1	7	2	12.9	4.9	13	258	0.5019	0.7263	0.5936	0.9384
esp 5	50	1	10	2	8.4	5	12.9	262.5	0.6056	0.7207	0.6582	0.9536
esp 6	20	1	5	2	8.4	5.3	12.6	262.5	0.6000	0.7039	0.6478	0.9526
esp 7	26	1	6	2	7.6	5.2	12.7	263.3	0.6256	0.7095	0.6649	0.9557
esp 8	45	10	9	2	14.1	5.2	12.7	256.8	0.4739	0.7095	0.5682	0.9332
esp 9	45	10	9	2	13.1	4.9	13	257.8	0.4981	0.7263	0.5909	0.9377

4.8 Variabilità dei classificatori

Fino ad ora sono stati confrontati i classificatori sulla base di metriche di performance quali *Precision, Recall e F-Measure*. I risultati appena visti riguardano però i risultati medi in *cross validation* e, al fine di effettuare un'ulteriore valutazione del modello, è necessario studiare il comportamento dei classificatori tra una *fold* e l'altra cosi da analizzare la varaibilità dei classificatori attraverso la deviazione standard in termini di falsi negativi, falsi positivi, veri negativi e veri positivi. Ciò ci permette di capire la stabilità del modello, cioè la capacità di dare risposte più o meno simili al variare dei dati *out of sample*.

Riprendendo i risultati del paragrafo precedente, sono stati presi i classificatori considerati migliori in termini delle tre metriche utilizzate (figure 4.1, 4.2, 4.3, 4.4, 4.5, 4.6), con un occhio in particolare alla *Recall*.

Come mostrato dalla tabella 4.7, il *Random forest* e *SVM linear* sono quelli che presentano minor variabilità nella previsione delle osservazioni tra una *fold* e l'altra. Nonostante SVM linear, come visto sopra, riesca a minimizzare i falsi negativi a 4.7 contro i 4.8 del Random Forest, in termini di variabilità il Random Forest ha una deviazione standard minore. Ciò significa che mediamente il *Random Forest* è leggermente più stabile del *SVM linear* ma anche di tutti gli altri classificatori. Anche *SVM radial* e *poly-*

nomial mostrano una bassa deviazione standard in termini di falsi negativi, ma risultano essere meno stabili in termini di falsi positivi. Il *Decision Tree*, come il *Random Forest*, sembra essere un classificatore stabile in generale, ma presenta comunque una deviazione standard sui falsi negativi più elevata rispetto agli altri classificatori. Lo stesso discorso vale per la regressione logistica che, oltre ad avere una deviazione standard più elevata sui falsi negativi, è molto variabile per quanto concerne i falsi positivi.

Tabella 4.7: Deviazione standard della matrice di confusione di ogni classificatore

Dev. Std. TP	**Dev. Std. TN**	**Dev. Std. FN**	**Dev. Std FP**	**Modello**
1.88561	2.93636	1.96920	3.02030	Regressione log.
1.88561	2.06559	1.96920	2.06827	Decision tree
1.79195	2.06559	1.81352	2.06827	Random forest
1.8737	1.88561	1.94650	1.85292	SVM linear
1.71269	4.13790	1.8287	4.18462	SVM radial
1.79195	3.37474	1.88561	3.43834	SVM polynomial

Capitolo 5

Conclusioni

L'obiettivo di questo progetto di tesi è stato quello di verificare se tecniche di campionamento quali *SMOTE* e *undersampling* su base *K-means* funzionassero su questi dati.

Dai risultati possiamo notare come queste tecniche abbiano portato a risultati soddisfacenti, in quanto hanno permesso di andare, in quasi tutti i casi, a migliorare le metriche di performance. Anche la regressione logistica ha beneficiato di queste tecniche di campionamento, riuscendo a migliorare la *Recall* e diminuire quindi il numero di falsi negativi. Essendo una tecnica ampiamente utilizzata in questi studi per la possibilità di fare inferenza, poterla utilizzare per fare anche previsioni resta comunque un'ottima scelta nonostante i modelli non parametrici sembrerebbero dimostrarsi più precisi nella previsione di entrambe le classi. Ciò potrebbe essere un suggerimento

per future ricerche in ambito EWS, in quanto fino ad ora non è stato mai considerato l'utilizzo di tecniche di *over e undersampling*. Infatti l'idea di queste tecniche è quello di bilanciare la probabilità delle due classi, quindi attribuire un peso pari a 0.5 sia per la classe minoritaria sia per la classe maggioritaria.

Il seguente studio inoltre potrebbe essere ulteriormente migliorato provando ad applicare tecniche ancor più sofisticate quali reti neurali ricorrenti *Long Short Term Memory*, reti in grado di apprendere dipendenze a lungo termine, o tecniche di *Cost-Sensitive Learning*, tecniche che lavorano a livello di algoritmo che si basano sull'assegnazione di un costo elevato per la *misclassificazione* della classe minoritaria e che mirano a minimizzare il costo complessivo (un esempio è il *Metacost di Domingos [1999]*.

Ulteriori alternative di tecniche di *oversampling* sono citate nella tesi di dottorato di Lauro et al. [2014] dove vengono illustrate soluzioni diverse dello *SMOTE* come ad esempio lo *SMOTE-borderline*, dove a differenza del tradizionale algoritmo che genera istanze artificiali in modo casuale tra due dati, questo va a generare dati artificiali solo lungo il bordo decisionale tra le due classi.

Inoltre, come già definito sopra, vi è anche la possibilità di continuare sullo stesso percorso seguito nel presente lavoro di tesi provando ad aggiungere ulteriori ritardi temporali nel dataset di training dando un peso maggiore

agli anni precedenti alla crisi in modo tale da sviluppare un modello in grado di fare previsioni a più anni dalla crisi stessa.

In conclusione, gli approcci non parametrici, in particolare *Decision Tree, Random forest e SVM linear*, si sono mostrati più efficienti in termini sia di previsioni corrette, sia di previsioni errate delle due classi rispetto alla regresisone logistica e questo va a confermare la maggior parte degli studi effettuati in letteratura. Tuttavia l'utilizzo di *undersampling* ed *oversampling* ha però permesso a tutti i classificatori di ottenere miglioramenti sulla base della *Recall* ed in generale della *F-Measure*, andando a confermare dunque l'obiettivo proposto all'inizio del presente lavoro di tesi.

Bibliografia

A. Antunes, D. Bonfim, N. Monteiro, and P. M. Rodrigues. Forecasting banking crises with dynamic panel probit models. *International Journal of Forecasting*, 34(2):249–275, 2018.

J. Beutel, S. List, and G. von Schweinitz. Does machine learning help us predict banking crises? *Journal of Financial Stability*, 45:100693, 2019.

K. Bluwstein, M. Buckmann, A. Joseph, S. Kapadia, and Ö. Simsek. Credit growth, the yield curve and financial crisis prediction: evidence from a machine learning approach. 2021.

J. Brownlee. *Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery, 2020.

G. Caggiano, P. Calice, L. Leonida, and G. Kapetanios. Comparing logit-based early warning systems: Does the duration of systemic banking crises matter? *Journal of Empirical finance*, 37:104–116, 2016.

E. J. Casabianca, M. Catalano, L. Forni, E. Giarda, S. Passeri, et al. An early warning system for banking crises: From regression-based analysis to machine learning techniques. *EconPapers. Orebro: Orebro University*, 2019.

T. Datascience. Svm hyperparameters explained with visualizations. https://towardsdatascience.com/svm-hyperparameters-explained-with-visualizations-143e48cb701b, 2020a.

T. Datascience. Decision trees explained. https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6, 2020b.

E. P. Davis and D. Karim. Comparing early warning systems for banking crises. *Journal of Financial stability*, 4(2):89–120, 2008.

A. Demirgüç-Kunt and E. Detragiache. The determinants of banking crises in developing and developed countries. *Staff Papers*, 45(1):81–109, 1998.

Z. Ding. Diversified ensemble classifiers for highly imbalanced data learning and their application in bioinformatics. 2011.

P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164, 1999.

R. Duttagupta and P. Cashin. The anatomy of banking crises. *IMF Working Papers*, 2008(093), 2008.

M. Holopainen and P. Sarlin. Toward robust early-warning models: A horse race, ensembles and model uncertainty. *Quantitative Finance*, 17(12):1933–1963, 2017.

R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

G. Kaminsky, S. Lizondo, and C. M. Reinhart. Leading indicators of currency crises. *Staff Papers*, 45(1):1–48, 1998.

N. KDnuggets. Support vector machines: a simple explanation. https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html, 2016.

M. L. Laeven and M. F. Valencia. *Systemic banking crises revisited*. International Monetary Fund, 2018.

C. Lauro, M. Aria, and M. Marino. Tecniche di ricampionamento per dataset con classi di risposta sbilanciate. una proposta metodologica per dataset con predittori di natura numerica e categorica. 2014.

C. Pigini. Penalized maximum likelihood estimation of logit-based early warning systems. *International Journal of Forecasting*, 37(3):1156–1172,

2021. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2021.01.004. URL https://www.sciencedirect.com/science/article/pii/S0169207021000042.

H. Rhys. *Machine Learning with R, the tidyverse, and mlr*. Simon and Schuster, 2020.

M. Schularick and A. M. Taylor. Credit booms gone bust: Monetary policy, leverage cycles, and financial crises, 1870-2008. *American Economic Review*, 102(2):1029–61, 2012.

L. S. Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson Education India, 2016.

B. Thehindu. What is the gdp deflator? https://www.thehindu.com/, 2018.

E. Tölö. Predicting systemic financial crises with recurrent neural networks. *Journal of Financial Stability*, 49:100746, 2020.

T. Worldbank. Banking crisis. https://www.worldbank.org/en/publication/gfdr/gfdr-2016/background/banking-crisis, 2016.