Rilevazione delle frodi nelle carte di credito

Giorgio Carbone[1], Gianluca Cavallaro [2], Remo Marconzini[3], Gianluca Scuri[4]

Sommario
Nell'ambito finanziario, uno degli aspetti più rilevanti è l'individuazione di transazioni fraudolente sulle carte di credito. Si può parlare sia di frodi "autorizzate", dove il possessore della carta processa un pagamento verso un account sospetto, che di frodi "non autorizzate", dove il possessore della carta non fornisce alcuna autorizzazione alla transazione, che viene eseguita da una terza parte. Dal punto di vista economico, è essenziale per gli istituti finanziari riuscire a individuare preventivamente queste operazioni: nel 2018 il Regno Unito ha subito frodi sulle carte di credito per una ammontare di £844,8 milioni. Al giorno d'oggi, le transazioni operate mediante carte di credito risultano essere molto sicure, e questo è dovuto anche al grande sforzo portato dagli istituti finanziari in materia di individuazioni di "cyber crimini". Per riuscire a completare questo compito in maniera efficace, uno degli obiettivi da raggiungere è quello di individuare la migliore tecnica di Machine Learning, e in particolare il miglior classificatore, che sia in grado di riconoscere, e di conseguenza bloccare, questo tipo di transazioni. Il progetto si concentra sul contrastare gli effetti negativi che un dataset con classi sbilanciate, come in questo caso, ha sulla ricerca e stima del miglior classificatore ai fini di previsione delle transazioni fraudolente. Le prestazioni dei modelli saranno valutate non solo tramite le usuali metriche, ma anche tenendo conto dei costi di computazione e rispetto alla qualità dei servizi antrifrode forniti al titolare della carta di credito.

Keywords
Machine Learning – Fraud – Classification

[1,2,3,4] *Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli studi di Milano-Bicocca, Milano, Italia*

Indice

1. Dataset

Il dataset contiene le transazioni eseguite tramite carte di credito nel settembre 2013 da cittadini europei. Il dataset contiene le transazioni avvenute nell'arco di due giorni, per un totale di 284.807 operazioni.

Il dataset a disposizione è il risultato di una "Principal Component Analysis", e di conseguenza contiene solo attributi numerici. Per questioni di privacy non sono fornite informazioni sugli attributi originali. Gli unici attributi mantenuti fra quelli originali sono 'Time', che contiene i secondi trascorsi dalla prima transazione del dataset da ciascuna delle altre transazioni, e 'Amount', che fa riferimento all'importo della transazione.

La variabile di risposta è l'attributo 'Class', variabile binaria, che assume il valore 1 in caso di transazione fraudolenta e 0 in caso di transazione legittima.

2. Data exploration

Inizialmente, viene affrontata una fase di esplorazione dei dati al fine di familiarizzare con il dataset.

2.1 Descriptive Statistics

Come descritto in sezione 1, il dataset è frutto dell'applicazione di una procedura di Principal Component Analysis che ha generato 28 attributi numerici. Come si può notare in figura 1, gli attributi risultano avere tutti una distribuzione simile.
Tutte le variabili sono infatti standardizzate, centrate attorno allo zero e con diversi outlier. Per questo, non vengono eseguite ulteriori operazioni su di esse.

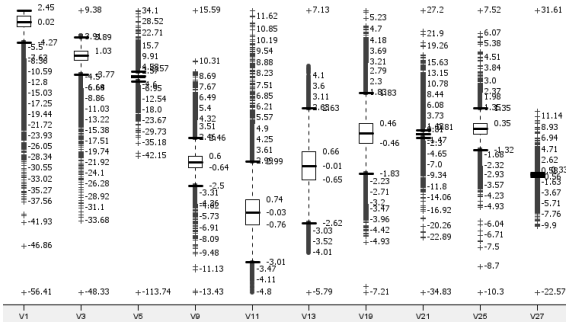

Figura 1. Distribuzione di alcune delle variabili ottenute da PCA

Gli unici due attributi mantenuti fra quelli originali sono l'attributo 'Time' e l'attributo 'Amount'.
Per quanto riguarda l'attributo 'Time' la distribuzione è la seguente (figura 2):

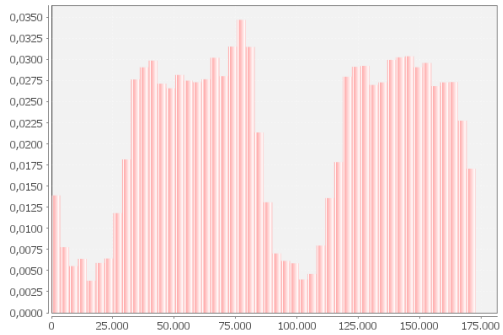

Figura 2. Distribuzione della variabile Time

7

Come descritto in precedenza, le transazioni sono riferite ad un arco temporale di due giorni, come si può notare dalla distribuzione.
Invece, per quanto riguarda l'attributo 'Amount' (figura 3):

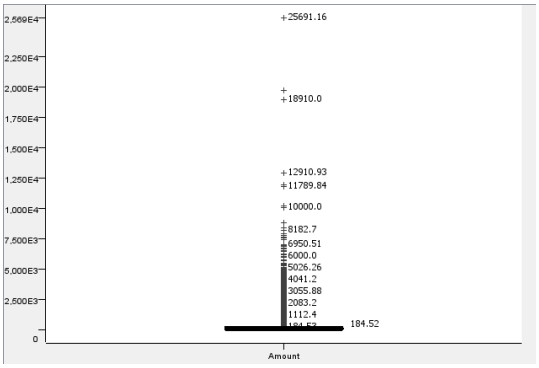

Figura 3. Distribuzione della variabile Amount

Si nota come la distribuzione sia fortemente asimmetrica, con la presenza di un ampio numero di outlier. Peraltro, si nota come nel dataset siano presenti 1825 transazioni a cui corrisponde un importo pari a zero. Di queste 1825, 27 osservazioni appartengono alla classe minoritaria. Tali transazioni potrebbero riferirsi a operazioni di verifica quali integrità della carta di credito, verifica di informazioni del titolare della carta ecc.., e quindi di un importo trascurabile (es: £0.01), ma con il rischio che, per le transazioni classificate come illegittime, tali operazioni abbiano lo scopo di ottenere informazioni sensibili. Non avendo informazioni aggiuntive sul significato di questo tipo di transazioni, si decide di mantenerle all'interno del dataset.

Analisi correlazioni

Andiamo ad analizzare la correlazione fra gli attributi presenti nel dataset (figura 4):

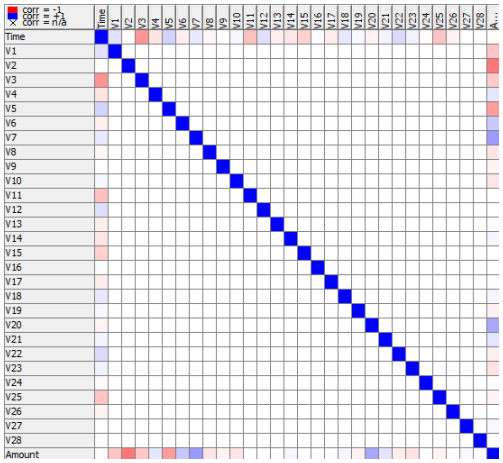

Figura 4. Matrice di correlazione

Come ci si poteva attendere, la correlazione fra le variabili ottenute da una PCA è trascurabile. Le uniche due variabili che presentano delle correlazioni significative con le 28 variabili ottenute dalla PCA sono 'Time' e 'Amount'. I valori più elevati sono $-0,43$ (Time-V3) e $-0,51$ (Amount-V2). Ciononostante, tali valori non rappresentano comunque una correlazione così rilevante da pensare di eliminare gli attributi coinvolti, considerando anche la difficoltà d'interpretazione degli attributi introdotta dall'uso di una PCA.

2.2 Data Partitioning

Il dataset non presenta valori mancanti.
Per quanto riguarda il partizionamento, si decide di seguire uno schema Training - Validation - Test. Il dataset viene pertanto suddiviso in due partizioni: partizione A (training e validation set, 67% del dataset iniziale) e partizione B (test set, 33% del dataset iniziale). Per poter mantenere in ogni partizione la distribuzione originale rispetto all'attributo 'Class' è stato effettuato un campionamento stratificato rispetto alla variabile di risposta. La partizione A verrà utilizzata per l'addestramento dei modelli, per la fase di feature selection e per l'ottimizzazione del modello. La partizione B, invece, sarà utilizzata per la fase di validazione dei risultati dei diversi modelli, che verranno testati su dati mai utilizzati in precedenza, in modo da valutare anche l'eventuale presenza del fenomeno di overfitting.

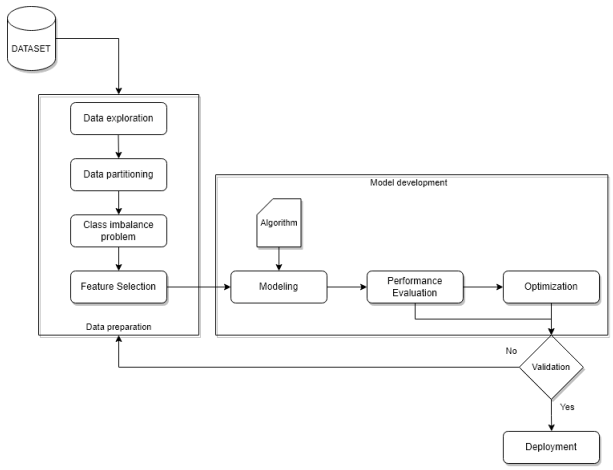

Figura 5. Project Workflow

3. Class Imbalance Problem

Dall'analisi dell'attributo target, è facile rendersi conto di quanto esso sia fortemente sbilanciato (figura 6):

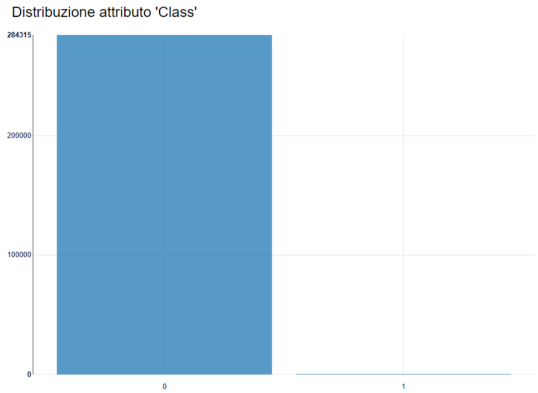

Figura 6. Distribuzione variabile target

In particolare, delle 284.807 transazioni contenute all'interno del dataset solo 492 sono fraudolente, per una percentuale dello $0,17\%$.
Si rende quindi necessario applicare delle opportune tecniche che tengano conto del forte sbilanciamento presente nel dataset, di seguito brevemente discusse.

3.1 Equal size sampling

Si tratta di una tecnica di random undersampling che va a ridimensionare la classe maggioritaria, andandone a mantenere soltanto un suo sottinsieme. L'equal size sampling consiste nel campionare randomicamente i record appartenenti alla classe maggioritaria fino ad ottenere un dataset composto dallo stesso numero di eventi rari (frodi) ed eventi frequenti (transazioni legittime).

3.2 SMOTE

Rappresenta una delle più popolari tecniche di oversampling. É l'acronimo di "Synthetic Minority Oversampling Technique". SMOTE, infatti, genera delle osservazioni "sintetiche" a partire dalla classe minoritaria, aggiungendole ai dati esistenti, fino a riequilibrare le due classi. Le nuove osservazioni sono generate secondo un criterio di similarità rispetto ai record esistenti.

3.3 Cost sensitive learning

Si tratta di un metodo che introduce all'interno del modello una specifica matrice di costo, ai fini di dare un peso maggiore o minore ai diversi errori che il modello può commettere. Nel caso in questione, ad esempio, l'errore che comporta il maggior costo per l'istituto finanziario è quello della classificazione di una transazione fraudolenta come una transazione legittima. Ciononostante, non bisogna trascurare il fatto che un numero elevato di transazioni legittime processate come frodi potrebbero avere un impatto negativo sulla reputazione, in termini di rischio, dell'istituto finanziario o sulla "customer satisfaction", la soddisfazione dei clienti.

Struttura matrice di costo

Il punto critico, dunque, è quello di scegliere la corretta struttura della matrice.

Come detto, nel nostro caso di interesse il problema più grande si verifica quando una frode non viene correttamente identificata, venendo classificata come transazione legittima. In questi casi, la transazione viene autorizzata senza effettuare controlli, causando un danno equivalente all'ammontare della transazione. Pertanto, sembra ragionevole considerare come costo associato ai falsi negativi il valore economico della transazione: nel nostro caso è stato considerato il valore medio di tutte le transazioni fraudolente presenti nel dataset, pari a 122€.

Decisamente più complesso è stimare il costo delle operazioni di controllo da eseguire in presenza di frode, indipendentemente dal fatto che si tratti di veri positivi o di falsi positivi, per cui andiamo a considerare lo stesso costo. Diversi esperti di dominio concordano sul fatto che tale costo è circa 1/100 del costo per i falsi negativi. In generale, in letteratura, si possono trovare considerazioni più ampie. Il dataset contiene i dati relativi a transazioni risalenti al settembre del 2013 in territorio europeo. Assumendo che tali dati siano rappresentativi della situazione generale, ipotizziamo che le transazioni siano gestite dai poli economici più rilevanti a livello europeo, come Londra, Parigi, Bruxelles, Lussemburgo, Amsterdam, Francoforte o Dublino. Per ciascuna di queste città, possiamo recuperare il salario medio annuale al 2013 facendo riferimento ai dati dell'OECD[1]. Facendo una media dei valori ottenuti si ottiene un salario medio annuo di 43.358€. A questo punto, assumendo che il tempo necessario al controllo di una transazione, grazie ai moderni sistemi informatici, sia di 5 minuti (12 controlli all'ora), possiamo calcolare il costo medio per il singolo controllo. Coi dati a disposizione, esso risulta di circa 1,75€. Tale valore viene considerato come il costo legato a veri e falsi positivi.

Infine, per quanto riguarda le transazioni legittime classificate correttamente, i veri negativi, è lecito immaginare che

[1] Organizzazione Europea per la Cooperazione e lo Sviluppo Economico

esse siano gestite da un software in maniera automatica. Per questo, il costo associato ai veri negativi è 0.

Riassumendo, la matrice di costo considerata assume la seguente forma (figura 7) :

MATRICE DI COSTO		CLASSE PREVISTA	
		-1	+1
CLASSE REALE	-1	0	1,75
	+1	122	1,75

Figura 7. Matrice di costo

3.4 Performance measures

Per poter valutare il miglior classificatore utile al nostro obiettivo è necessario calcolare delle misure di performance, per poter effettuare un confronto fra diversi modelli. Normalmente, la prima misura di performance considerata è l'accuratezza, che viene definita nel modo seguente:

$$\textbf{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

dove:

- TP e TN rappresentano il numero di record appartenenti, rispettivamente, alla classe positiva e alla classe negativa ed effettivamente classificati come tali;
- FP e FN rappresentano il numero di record appartenenti, rispettivamente, alla classe positiva e alla classe negativa ma classificati come appartenenti alla classe opposta.

Più alto è il valore di accuratezza, più alto è il numero di record correttamente classificati dal modello.
Tuttavia, nel caso di classi sbilanciate, l'accuratezza non è una misura consistente. In un caso come il nostro, dove la classe minoritaria rappresenta solo lo $0,17\%$ del dataset completo, il classificatore tende a concentrarsi solo sui valori appartenenti alla classe maggioritaria, classificando tutti i record come appartenenti a quella classe. Nonostante questo, visto il basso numero di eventi rari, l'accuratezza del modello risulta comunque molto elevata. Pertanto, al fine di affrontare il problema delle classi sbilanciate, è necessario individuare misure di valutazione delle performance alternative.
In particolar modo, le quantità considerate sono Recall, Precision, F1-measure e AUC, l'area sottesa alla curva ROC.

$$\textbf{Recall} = \frac{TP}{TP+FN}$$

rappresenta la frazione di record appartenenti alla classe positiva correttamente classificati. Più alto è il valore di recall, meno sono i record erroneamente attribuiti alla classe negativa;

$$\textbf{Precision} = \frac{TP}{TP+FP}$$

rappresenta la frazione dei record effettivamente appartenenti alla classe positiva fra tutti quelli classificati come tali. Più alto è il valore di precision, minore sarà il numero di falsi positivi;

$$\textbf{F1-measure} = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

è una misura che assume valori compresi nell'intervallo $[0,1]$ e che rappresenta una media armonica di Recall e Precision. Ad elevati valori di F1-Measure corrispondono valori ragionevolmente alti sia di recall che precision;
AUC: un modo per confrontare fra di loro diversi modelli è quello di valutare l'area sottesa alla curva ROC, curva che mette in relazione la percentuale di TP con la percentuale di FP. Più alto è il valore dell'AUC, minore è il numero di falsi positivi che dobbiamo "sopportare" per avere un certo numero di veri positivi.

3.5 Procedure assessment

La tecnica di gestione del problema delle classi sbilanciate che maggiormente si adatta al nostro problema viene scelta attraverso una analisi di benchmark, ovvero una preliminare stima di un classificatore individuato come benchamrk e una valutazione delle relative performance in termini di Recall, Precision, costo computazionale e considerazioni su caratterstiche proprie di ogni tecnica. Il classificatore scelto come benchmark è il *Random Forest*, implementato tramite nodo *Weka*. Tale modello è stato scelto in quanto:

- lavora bene con dataset di grandi dimenioni e con attributi target nominali;
- non richiede una fase di *feauture engineering*; (ovvero scaling e normalization)
- modello robusto rispetto a *outliers*.

Il classificatore viene addestrato prima sul dataset originale sbilanciato, e successivamente mediante le tre strategie descritte nella sezione 3. Per fare questo vengono utilizzati i nodi *Equal size sampling* e *Smote* di Knime e il nodo *CostSensitiveLearner* di Weka. In ciascun caso vengono calcolate le diverse misure di performance, che vengono di seguito riportate:

	Recall	Precision	F1-Measure	Accuracy	AUC	Cost
Class Unbalanced	0.704	0.927	0.800	0.999	0.910	-
Equal size sampling	0.852	0.052	0.098	0.973	0.963	-
SMOTE	0.784	0.847	0.814	0.999	0.937	-
Cost sensitive	0.821	0.289	0.427	0.996	0.909	4460

Tabella 1. Analisi benchmark su classificatore benchmark, *Random Forest*

L'obiettivo è quello di massimizzare la *Recall* mantenendo un valore accettabile di *Precision*. Così facendo, infatti, si limitano il più possibile i *falsi negativi*, che come detto sono gli errori più rilevanti. Tuttavia anche i *falsi positivi* non sono da trascurare: per quanto non siano particolarmente onerosi in termini di costo, possono essere comunque dannosi per

l'immagine dell'istituto finanziario.
Sulla base dei valori ottenuti,vengono fatte le seguenti considerazioni. La tecnica di *Equal size sampling*, pur avendo il più alto valore di *Recall*, ha una *Precision* pessima. Inoltre, opera una drastica riduzione della dimensionalità del dataset eliminando le osservazioni della classe maggioritaria, comportando la potenziale perdita di informazioni utili alla corretta identificazione delle transazioni legittime. La tecnica *Smote* ha una *Precision* molto elevata, ma il più basso valore di *Recall*. Oltretutto, genera istanze "fittizie", andando ad aumentare ulteriormente la complessità di un dataset già di grandi dimensioni. La tecnica *Cost Sensitive*, invece, ha un alto valore di *Recall* e un valore di *Precision* accettabile, lascia inalterato il dataset di partenza e, soprattutto, consente di tenere in considerazione aspetti propri di dominio, come ad esempio il costo.

Pertanto, la scelta della tecnica di gestione del problema delle classi sbilanciate ricade sull'approccio di *Cost Sensitive Learning*.

4. Model Evaluation

Determinata la tecnica ottimale per la gestione del problema delle classi sbilanciate, lo step successivo si concentra nell'individuazione del modello più adatto, andando a effettuare un confronto fra diversi classificatori. I modelli presi in considerazione rientrano nelle seguenti famiglie:

- **Modelli euristici**: quali *J48*, di *Weka*, *Random Forest* (RF), *k-Nearest Neighbour* (KNN). Sebbene non garantiscano di giungere a risultati ottimali, tali modelli sono molto comuni in quanto flessibili e non particolarmente difficili da interpretare.
- **Modelli probabilistici**: quali *Naive Bayes* (NB). Permettono, a partire da una elevata quantità di dati, di ottenere risultati molto accurati sotto ipotesi di indipendenza degli attributi.
- **Modelli di regressione**: basati sulla *regressione logistica* (LR). Il loro vantaggio è rappresentato dalla possibilità di considerare qualsiasi tipo di input, risultando così estremamente flessibili.
- **Modelli di separazione**: quali *Support vector machine* (SVM) e *Multilayer Perceptron* (MLP). Si basano sulla sepazione dello spazio degli attributi.

Tutti i modelli sono implementati tramite metanodo Weka, *CostSensitiveClassifier*.

4.1 Feature selection

Inizialmente si procede con una fase di feature selection. Per quanto le variabili siano frutto dell'applicazione di una PCA, e pertanto si abbiano variabili esplicative incorrelate fra loro, si desidera comunque verificare l'eventuale presenza di attributi ridondanti o irrilevanti per evitare il fenomeno di *overfitting*. Il processo di feature selection è stato implementato tramite metodo *wrapper*, seguendo un approccio di tipo *backward feature elimination*. L'algoritmo iterativo parte dall'intero insieme di attributi e ad ogni step toglie l'attributo meno rilevante, calcolando le misure di performance del classificatore relative a quel sottoinsieme, fino ad arrivare alla stima con un solo attributo. Lo step finale consiste nel selezionare il sottoinsieme ottimale secondo una certa metrica da massimizzare o minimizzare. Nel nostro caso il criterio selezionato è stata la massimizzazione della Recall. La procedura è stata eseguita sui dati della partizione A.
La procedura è stata eseguita per ciascuno dei 7 classificatori considerati.
Per ciascun classificatore, il subset di attributi ottimali individuato risulta essere:

Classificatore	Nr di attributi
J48	28
Random Forest	27
Naive Bayes	23
Logistic regression	21
SVM	28
Multilayer Perceptron	23
K-Nearest Neighboor	16

Tabella 2. Feature selection

4.2 Cross Validation

Dopo aver individuato per ciascun classificatore il rispettivo sottoinsieme di variabili rilevanti, si vanno a valutare le performance di ognuno di essi. Si applica, quindi, una *Stratified k-fold Cross Validation*, con $k = 10$, sui dati della partizione A, seguita da una validazione dei risultati tramite *Holdout* sui dati della partizione B, ai fini di evitare il fenomeno di overfitting. In questo caso viene riportato anche il tempo necessario per l'addestramento del classificatore, in modo da avere un'ulteriore dimensione di confronto fra i vari modelli. Una sintesi delle performance di ogni modello è riportata nella tabella 6:

Model	Recall	Precision	F1-Measure	Accuracy	AUC	Cost	Ex. Time (ms)
LR	0.864	0.563	0.680	0.999	0.931	**651**	8661
MLP	0.848	0.638	0.721	0.999	/	**702**	423802
RF	0.871	0.317	0.463	0.996	0.934	**707**	44773
J48	0.815	0.667	0.719	0.999	0.907	**830**	25760
SVM	0.792	0.867	0.825	0.999	0.896	**878**	17739
KNN	0.787	0.858	0.817	0.999	0.894	**939**	5556
NB	0.885	0.069	0.127	0.979	0.932	**1335**	5025

Tabella 3. Performance evaluation k-fold Cross Validation

Seguendo il nostro obiettivo di business, il parametro maggiormente d'interesse risulta il costo. In particolare:

- **Regressione Logistica**, **Multilayer Perceptron** e **Random Forest** risultano essere i modelli caratterizzati dal costo più basso, grazie a valori di Recall elevati e da valori di Precision accettabili. Inoltre, presen-

tano i valori di AUC più elevati[2]. Tra questi modelli il Multilayer Perceptron risulta essere il modello computazionalmente più dispendioso.

- **J48**, **Support vector machine** e **K-Nearest Neighoor** fanno registrare un costo più elevato, seppur non in maniera così evidente, a causa di una Recall più bassa. L'aumento del costo non è così evidente in quanto tali modelli presentano valori di Precision più alti rispetto ai modelli presi in considerazione al punto precedente. Dal punto di vista computazionale nessun modello risulta particolarmente dispendioso (si noti l'efficenza computazione del modello K-Nearest-Neighboor).

- Per quanto riguarda il modello **Naive Bayes**, nonostante presenti la Recall più elevata, si notano pessime performance in termini di Precision, sintomo di una errata classificazione delle istanze della classe maggioritaria. A causa di ciò presenta il quindi il costo pià elevato.

La procedura di Cross Validation è stata svolta sulla partizione A. Ai fini di **validazione** i vari modelli sono stati inoltre stimati sui dati della **partizione B**, mai usati in precedenza, tramite una procedura di *Holdout*, come riportato in tabella 4.

Model	Recall	Precision	F1-Measure	Accuracy	AUC	Cost	Ex. Time (ms)
RF	0.833	0.292	0.433	0.996	0.915	**4218**	55847
LR	0.809	0.537	0.645	0.998	0.904	**4270**	9898
MLP	0.809	0.300	0.438	0.996	/	**4654**	382135
KNN	0.728	0.837	0.779	0.999	0.864	**5650**	5000
SVM	0.722	0.830	0.772	0.999	0.861	**5772**	11576
J48	0.722	0.807	0.762	0.999	0.861	**5780**	26696
NB	0.833	0.063	0.117	0.978	0.906	**7570**	6077

Tabella 4. Holdout Validation

Le osservazioni precedentemente fatte rimangono valide anche dopo la procedura di Holdout sulla partizione B. L'unica differenza significativa risulta nel calo del valore di *Precision* per il modello Multilayer Perceptron.

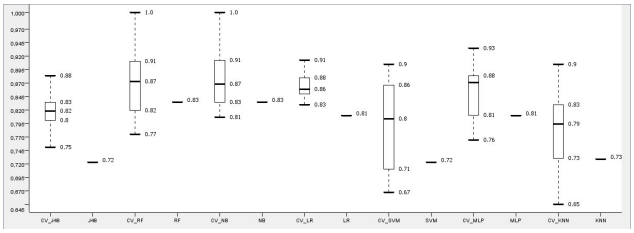

Figura 8. Recall Cross Validation vs Recall Holdout

Come evidenziato dai boxplot (figura 8 e 9), potrebbe esserci un leggero *overfitting* nei modelli **J48**, **Logisti Regression** e **Multilayer Perceptron**, aspetto che andrebbe maggiormente approfondito.

[2]Ad eccezione del Multilayer Perceptron, per cui non è stato possibile eseguire il calcolo. Infatti, aggiungendo le colonne relative alla probabilità di appartenenza alle classi di risposta utili al calcolo dell'AUC, il classificatore risultava incapace di classificare correttamente le istanze.

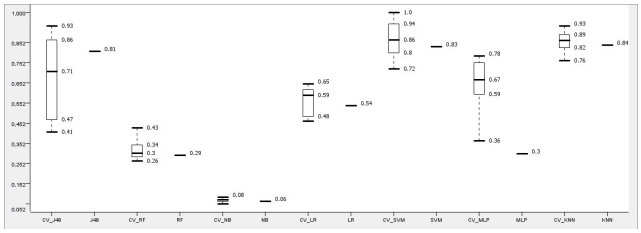

Figura 9. Precision Cross Validation vs Precision Holdout

4.3 Model optimization

Una volta analizzate le performance, in particolare costo e Recall, è stato applicato un algoritmo iterativo di ottimizzazione degli iperparametri con l'obiettivo di incrementare ulteriormente le performance. Per le considerazioni fatte in precedenza si è scelto di escludere il modello **Naive Bayes**, che peraltro non presenta iperparametri da poter ottimizzare. In particolare, per ogni modello sono stati identificati i parametri da ottimizzare e gli intervalli di valori da valutare. La strategia di ottimizzazione dei parametri considerata è *Bayesian Optimization*. Tale tecnica prevede due fasi: una prima fase di *warm up* in cui le combinazioni dei parametri sono scelte causalmente e valutate, e una seconda fase, che fa uso dell'algoritmo *Tree-structured Perzen Estimation*, che sulla base delle performance della fase di warm up cerca di trovare le combinazioni migliori e le valuta ulteriormente. In seguito alla procedura, i valori ottimali per i parametri considerati sono riportati nella seguente tabella, come evidenziano i grafici 10, 11, 12, 13, 14, 15 :

Classificatore	Parametro	Range	Valore ottimale
J48	minNumObj	0 - 100	10
RF	maxDepth	1-25	5
LR	ridge	0-1000	11
SVM	c (complexity)	0-100	85
MLP	hiddenLayer	1-13	2
KNN	k (nr. neighboors)	1-10	9

Tabella 5. Ottimizzazione iperparametri

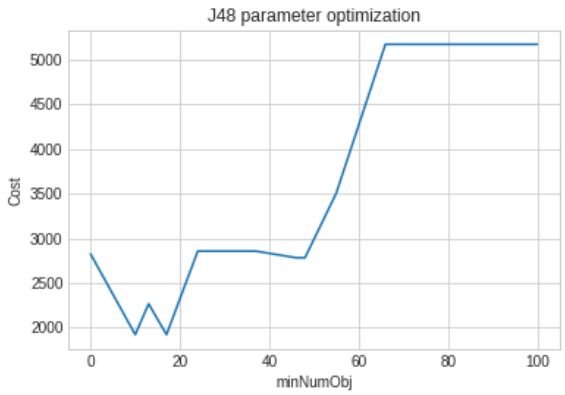

Figura 10. Ottimizzazione minNumObj, J48

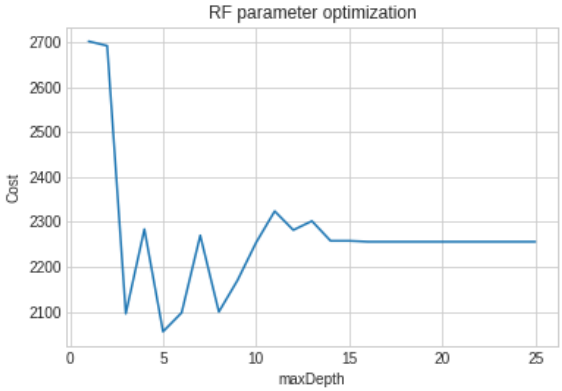

Figura 11. Ottimizzazione maxDepth, RF

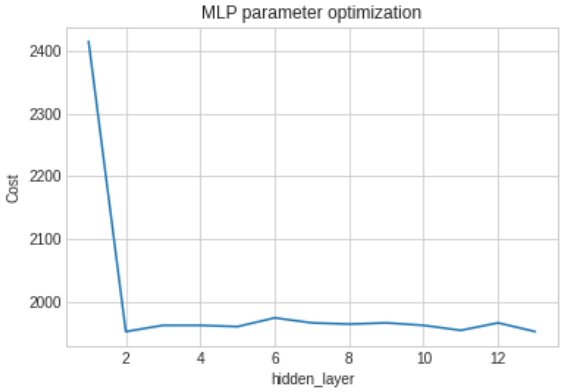

Figura 14. Ottimizzazione hidden layer, MLP

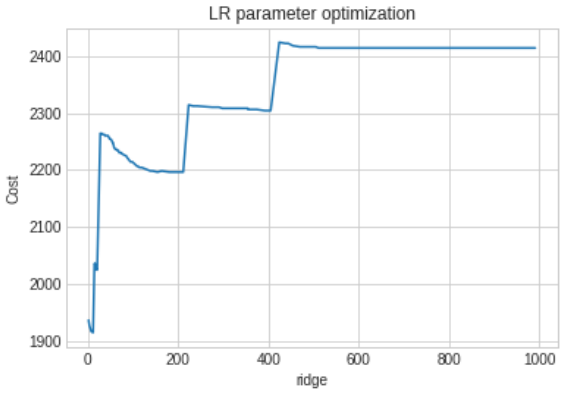

Figura 12. Ottimizzazione ridge, LR

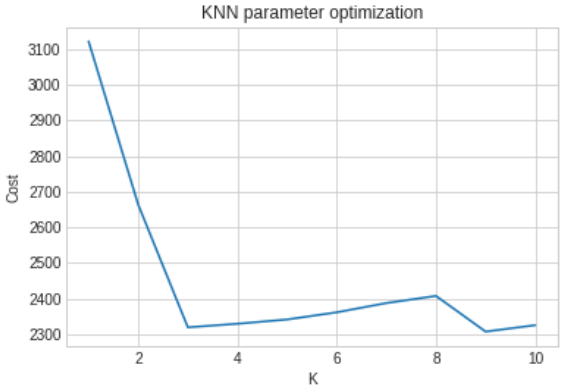

Figura 15. Ottimizzazione K, KNN

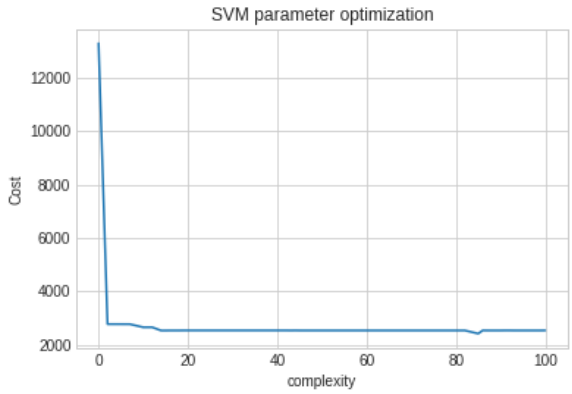

Figura 13. Ottimizzazione complexity, SVM

5. Model Validation

Come per la fase di model evaluation, anche in questo caso viene applicato una procedura di *Holdout* con addestramento sull'intera **partizione A** e test sui dati della **partizione B** ai fini di validazione dei modelli. In particolare si confrontano i risultati ottenuti con set di parametri di default con quelli ottenuti dopo il processo di ottimizzazione.

Rispetto ai risultati ottenuti con i parametri di default si nota un generale miglioramento delle perforamance per tutti i modelli, prevalentemente in termini di costo e Recall:

- **Regressione logistica** e **Random Forest** si confermano i classificatori migliori in termini di costo, Recall e AUC. In particolare per il modello Random Forest, nonostante un leggero peggioramento della Recall, migliora notevolmente la Precision, con conseguente diminuzione del costo. Inoltre, per tale modello si nota come l'ottimizzazione riduca notevolmente il tempo di esecuzione.

- **K-Nearest Neighboor** mostra un notevole miglioramento in termini di costo, dovuto ad un miglioramento della Recall anche a fronte di un peggioramento della Precision.

Model	Recall	Precision	F-Measure	Accuracy	AUC	Cost	Ex. Time (ms)
LR	0.815	0.559	0.663	0.998	0.907	**4132**	10603
RF	0.815	0.512	0.629	0.998	0.907	**4176**	24869
KNN	0.809	0.453	0.581	0.998	0.903	**4360**	4889
MLP	0.784	0.648	0.709	0.999	/	**4662**	80433
J48	0.784	0.460	0.580	0.998	0.891	**4822**	21126
SVM	0.728	0.825	0.774	0.999	0.864	**5654**	375247

Tabella 6. Holdout ottimizzato

- **Multilayer Perceptron**, **J48** e **Support Vector Machine** non presentano miglioramenti evidenti in termini di Recall, Precision e costo. Tuttavia, si nota un drastico calo del costo computazionale per il Multilayer Perceptron, dovuto a un minor numero di strati nascosti considerato, e al contrario si nota un deciso aumento del tempo di esecuzione per il modello Support Vector Machine a causa dell'aumento della complessità.

6. Conclusioni

L'obiettivo di questo lavoro è quello di identificare la tecnica e il modello di Machine Learning adatti alla classificazione delle transazioni fraudolente. Il problema principale che si è manifestato è stato quello delle classi sbilanciate. Tale problematica ha richiesto di utilizzare delle tecniche opportune per essere gestita. A tale scopo sono state utilizzate tecniche di undersampling (Equal Size Sampling), oversampling (SMOTE) e Cost Sensitive Learning.

Inizialmente è stata condotta una analisi di benchmark, tramite il modello Random Forest, al fine di selezionare il metodo più appropriato per la gestione del problema delle classi sbilanciate. I risultati, e la relativa analisi, hanno portato a selezionare l'approccio Cost Sensitive in quanto, oltre alle buone performance, non altera il dataset e permette di sfruttare informazioni di dominio. Dopodichè si è passati alla fase di feature selection, tramite metodo wrapper, dove si è andati a selezionare, per ciascuno dei diversi classificatori, solo il sottoinsieme di attributi rilevanti.

A questo punto, individuati gli opportuni classificatori, ne sono state valutate le performance al fine di individuare il più adatto per il nostro scopo. I modelli sono stati valutati tramite una K-fold Cross validation sull'intera partizione A dei dati, e successivamente i risultati sono stati validati tramite Holdout sulla partizione B dei dati, mai utilizzata in precedenza. Tale analisi porta all'esclusione del modello Naive Bayes, in quanto presenta performance notevolmente peggiori rispetto agli altri modelli.

I restanti modelli sono stati sottoposti ad un processo di ottimizzazione degli iperparametri al fine di migliorare ulteriormente le performance, nel tentativo di individuare il miglior classificatore. Dai risultati emergono Regressione Logistica e Random Forest come i modelli maggiormente performanti, con particolare riferimento a Recall e costo. In ogni caso non c'è una chiara evidenza delle migliori performance rispetto agli altri modelli, k-Nearest Neighboor, Multilayer Perceptron, J48 e Support vector machine.

Concludendo, nell'ottica di migliorare ulteriormente l'analisi, sarebbe opportuno indagare il fenomeno di overfitting precedentemente riscontrato e approfondire maggiormente la fase di ottimizzazione, con particolare focus su quali parametri ottimizzare e su quale range di valori.

Riferimenti bibliografici

[1] A. J. Figueredo and P. S. A. Wolf. Assortative pairing and life history strategy - a cross-cultural study. *Human Nature*, 20:317–330, 2009.

[2] J. M. Smith and A. B. Jones. *Book Title*. Publisher, 7th edition, 2012.

[3] Kha Shing Lim, Lam Hong Lee, and Yee-Wai Sim. A review of machine learning algorithms for fraud detection in credit card transaction. *International Journal of Computer Science & Network Security*, 21(9):31–40, 2021.

[4] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166. IEEE, 2015.

[5] David J Hand, Christopher Whitrow, Niall M Adams, Piotr Juszczak, and Dave Weston. Performance criteria for plastic card fraud detection tools. *Journal of the Operational Research Society*, 59(7):956–962, 2008.

[6] Alejandro Correa Bahnsen, Aleksandar Stojanovic, Djamila Aouada, and Björn Ottersten. Cost sensitive credit card fraud detection using bayes minimum risk. In *2013 12th international conference on machine learning and applications*, volume 1, pages 333–338. IEEE, 2013.

[7] Carlo Lauro, Massimo Aria, and Marina Marino. Tecniche di ricampionamento per dataset con classi di risposta sbilanciate. una proposta metodologica per dataset con predittori di natura numerica e categorica. 2014.

[8] Matthias Feurer, Jost Springenberg, and Frank Hutter. Initializing bayesian hyperparameter optimization via meta-learning. 29(1), 2015.

[9] Ritu Ratra, Preeti Gulia, and Nasib Singh Gill. Performance analysis of classification techniques in data mining using weka. *Available at SSRN 3879610*, 2021.

[10] Haihui You, Zengyi Ma, Yijun Tang, Yuelan Wang, Jianhua Yan, Mingjiang Ni, Kefa Cen, and Qunxing Huang. Comparison of ann (mlp), anfis, svm, and rf models for the online classification of heating value of burning municipal solid waste in circulating fluidized bed incinerators. *Waste Management*, 68:186–197, 2017.