Frodi finanziarie: un modello per scoprirle

Bekollari Fatjona[1], Ceccotti Alex[2], Giampino Alice[3], Missineo Nicholas[4], Pirotta Elisa[5]

Abstract
Autocertificare un reddito più basso di quello effettivo costituisce un reato di falso ideologico. Obiettivo di molte aziende investigative è quello di effettuare puntuali e precise indagini di riscontro volte alla scoperta dei tentativi di truffare lo stato dichiarando redditi fittizi. In generale lo Stato, il cui obiettivo è quello di garantire il benessere dei cittadini che vi fanno parte, difronte a coloro che dichiarano una situazione economica di difficoltà, applica una serie di agevolazioni tra cui ticket sanitari, riduzione delle tasse universitarie, contributo all'affitto e borse di studio. Non in tutti i paesi però le manovre pubbliche sono le stesse. Ad esempio, negli Stati Uniti d'America, che è poi lo stato da cui proviene il campione di popolazione che ha permesso la creazione di questo studio, la sanità è basata su criteri di natura essenzialmente privatistica e non sono previste agevolazioni universitarie per coloro che hanno un reddito ritenuto basso. È comunque nell'interesse dello Stato scoprire, attraverso diverse metodologie, chi dichiara dati non rispondenti al vero al fine di beneficiare degli aiuti pubblici oppure come unico tornaconto personale in quanto il contributo statale versato tramite le tasse sarebbe inferiore. Per questo motivo, assumendo veritieri i dati presenti nel dataset in esame, nell'elaborato sarà illustrato il procedimento utilizzato per creare un algoritmo di previsione avente come obiettivo principale l'identificazione dei soggetti più ricchi in base a determinate caratteristiche socio-economiche. Sarà poi onere dello Stato effettuare gli accertamenti opportuni.

Keywords
Reddito — Evasione — Modelli Predittivi — Mirsure di Performance

[1] *Università degli Studi di Milano Bicocca, CdLM CLAMSES*
[2] *Università degli Studi di Milano Bicocca, CdLM Data Science*
[3] *Università degli Studi di Milano Bicocca, CdLM CLAMSES*
[4] *Università degli Studi di Milano Bicocca, CdLM CLAMSES*
[5] *Università degli Studi di Milano Bicocca, CdLM CLAMSES*

Contents

Introduzione

Per raggiungere il nostro scopo abbiamo deciso di analizzare il dataset "*Adult*" presente sulla piattaforma Kaggle il quale è reso disponibile in forma ridotta da "*USA Census*"[1].
Il dataset originario è composto da 32.561 osservazioni e 15 variabili:

- *Age*: l'età degli individui
- *Workclass*: il tipo di lavoro che svolge l'individuo
- *Fnlwgt*: un peso assegnato ad ogni individuo in base ad alcune caratteristiche demografiche; a individui con valori simili corrispondono caratteristiche analoghe
- *Education*: il più alto livello di istruzione raggiunto da quell'individuo
- *Education.num*: il più alto livello di istruzione in forma numerica raggiunto dall'individuo in questione (si assume che le modalità di "education" siano equidistanti)
- *Marital.status*: lo stato civile dell'individuo
- *Occupation*: l'occupazione dell'individuo

- *Relationship*: contiene i valori relativi alle relazioni familiari
- *Race*: la descrizione dell'etnia dell'individuo
- *Sex*: il sesso biologico (uomo/donna)
- *Capital gain*: la plusvalenza degli investimenti effettuati dall'individuo
- *Capital loss*: la minusvalenza degli investimenti effettuati dall'individuo
- *Hours per week*: le ore lavorate a settimana
- *Native country*: il paese di origine dell'individuo
- *Income*: il reddito degli individui suddiviso in >50K oppure <=50K

Per una descrizione più accurata delle variabili e delle condizioni applicate per costruire il dataset si rimanda alla piattaforma Kaggle[2].
La variabile dipendente della nostra analisi è *income* e, di conseguenza, l'obiettivo è prevedere se un individuo abbia un reddito superiore a $50.000 annui. È bene evidenziare che in tutta la nostra analisi, quando si parla di individui con reddito superiore a $50.000 annui, li definiamo come classe positiva in quanto rappresentano la classe di interesse dell'analisi. Analogamente coloro che hanno un reddito inferiore a $50.000 annui appartengono alla classe negativa.
La relazione è suddivisa nel modo seguente:

- Nel primo paragrafo è stata affrontata un'analisi preliminare del dataset in cui sono state escluse alcune variabili in quanto poco utili ai fini del progetto, sono stati imputati alcuni valori mancanti e sono state ricodificate le variabili nominali.
- Nel secondo paragrafo abbiamo valutato diversi modelli ponendo una particolare attenzione alle misure di performance che verranno utilizzate.
- Nel terzo paragrafo abbiamo sviluppato tali modelli applicando i metodi di *hold out* e *cross validation*, restringendo, quindi, la nostra attenzione ai soli modelli più performanti. Quest'ultimi sono stati infine testati sul *validation set* per decidere quale di questi risultasse il migliore.

Infine, abbiamo proceduto commentando i risultati ottenuti nel processo di analisi dei dati, esponendo gli elementi più importanti dell'indagine. Tali risultati sono riportati dettagliatamente nelle conclusioni finali di questo elaborato.

1. *Preprocessing*

Delle 15 variabili iniziali non ne utilizzeremo 4 a priori. In particolare:

- *fnwgt* perché non sono presenti le informazioni necessarie per un appropriato utilizzo
- *capital.loss* e *capital.gain* dato che per l'87% delle osservazioni assumono entrambe valore 0 (di conseguenza, data anche la difficile interpretazione, risulterebbero soltanto una complicazione)
- *native.country* poiché per il 90% delle osservazioni ha come modalità '*United-States*' e contiene 583 *missing values* di difficile imputazione

1.1 *Missing Replacement*

Una volta eliminati questi attributi, passiamo all'imputazione dei dati mancanti contenuti esclusivamente in due variabili: *workclass* e *occupation*. La presenza di questi *missing* non è casuale, infatti tra i livelli delle due esplicative non è presente nessuna modalità riconducibile agli inattivi (studenti, pensionati, ...). Abbiamo perciò scelto di considerare solo i record aventi valore mancante in entrambe le variabili in questione e aventi un'età minore di 23 (studenti) o maggiore di 64 anni (pensionati). I record con valori mancanti aventi un'età intermedia sono stati eliminati (939 osservazioni) poiché non sarebbe stato possibile assegnarli a nessuna classe specifica. Abbiamo quindi imputato opportunamente alla variabile *occupation* i valori "studente" o "pensionato" e alla variabile *workclass* "inattivo" per entrambe le fasce di età. Dopo tale processo, ci siamo accorti che 7 osservazioni avevano ancora valori mancanti nella variabile *occupation*, poiché avevano già assegnato il valore "never-worked" per *workclass*. Anche queste osservazioni sono state eliminate dal dataset, che ora contiene 31.615 record.

1.2 Aggregazione

Dovendo utilizzare algoritmi che necessitano di binarizzazione, abbiamo deciso di aggregare le modalità di alcune variabili per evitare problemi di alta dimensionalità. A tal fine, oltre a considerare aggregazioni logiche (per esempio nella variabile *workclass* tutti i lavori statali), ci siamo basati anche sulla connessione delle diverse modalità di ogni covariata con la variabile risposta *income*. Ad esempio, la tabella 1 è quella su cui si è basata l'aggregazione per quanto riguarda la variabile *occupation*. "N" indica la numerosità della modalità di *occupation* avente *income* >50K e "% MOD" indica la percentuale di classe positiva interna alla modalità di *occupation*. Avendo a disposizione questa tabella, è stato facile accorpare le osservazioni in 4 modalità aggregate. Lo stesso procedimento è stato fatto per le variabili *workclass*, *education*, *marital status*, *relationship* e *race*.

1.3 Ricampionamento

Svolto questo procedimento, dal dataset è stato estratto il *validation set* (10%) e il restante 90% delle osservazioni è stato diviso in *training set* (67%) e *test set* (33%). Questa divisione multipla è necessaria affinché le misure di accuratezza utilizzate per valutare i modelli non siano minimamente influenzate dal processo che porterà alla stima dei modelli stessi.

Occupation	N	% MOD	AGG
Exec-managerial	1968	48,40%	Spec Work
Prof-specialty	1859	44,90%	
Protective-serv	211	32,51%	Tecnical Work
Tech-support	283	30,50%	
Sales	983	26,93%	
Craft-repair	929	22,66%	
Transport-moving	320	20,04%	
Retired	51	14,78%	Mid Salary
Adm-clerical	507	13,45%	
Machine-op-inspct	250	12,49%	
Farming-fishing	115	11,57%	
Armed-Forces	1	11,11%	
Handlers-cleaners	86	6,28%	Low Salary
Other-service	137	4,16%	
Priv-house-serv	1	0,67%	
Student	1	0,18%	

Tabella 1. Income >50K

Prima di procedere allo sviluppo dei modelli, sono necessarie ancora alcune operazioni. In particolare, dato che i record "positivi" rappresentano il 24% della variabile *income*, per sviluppare modelli che prevedano bene soggetti con *income* >50K è opportuno ricampionare adeguatamente il dataset. La tecnica che abbiamo scelto di utilizzare è l'*oversampling*[3], la quale consiste nel campionare casualmente (con reinserimento) dalla classe minoritaria un numero di osservazioni tale da pareggiare la cardinalità della classe maggioritaria. Nel nostro caso, il *training set* conterrà esattamente 14.419 record positivi e altrettanti record negativi. Le osservazioni non utilizzate durante il ricampionamento verranno aggregate al *test set*. Inoltre, un procedimento analogo è stato utilizzato per creare il dataset che verrà utilizzato per la *cross-validation*.

1.4 *Feature Selection* e Binarizzazione

Gli ultimi passi prima di sviluppare i modelli predittivi consistono nella selezione delle variabili rilevanti e nella binarizzazione delle variabili nominali, necessaria per la stima di alcuni modelli (reti neurali, ...). Per quanto riguarda la *feature selection* è stata adottata la tecnica CFS, che consiste nella stima di un modello, nel nostro caso un albero di classificazione[1], e nell'identificazione delle variabili più rilevanti per tale modello. Il risultato ci ha portato a tenere come esplicative solamente *age*, *education.num*, *hours.per.week*, *marital* (ricodificata), *relationship* (ricodificata) e *occupation* (ricodificata). Successivamente, dato che *relationship* e *marital* sono quasi perfettamente correlate tra loro, abbiamo scelto di tenere tra le due soltanto la prima. Inoltre, prendendo in considerazione il caso in cui la nostra aggregazione abbia in qualche modo influenzato il procedimento di selezione, abbiamo applicato lo stesso algoritmo alle variabili non ricodificate e al dataset non ricampionato, ottenendo comunque gli stessi risultati. Infine, come già anticipato, abbiamo binarizzato le variabili *occupation* (4 *dummies* - *spec_work*, *tecnical_work*, *mid_salary* e *low_salary*) e *relationship* (una *dummy* - *spouse*).

[1] Sono stati provati anche altri modelli ottenendo risultati analoghi

2. Modelli e Misure di Valutazione

2.1 Modelli Utilizzati

Per trarre le conclusioni necessarie dai dati osservati, abbiamo utilizzato 7 modelli predittivi differenti: *Random Forest*, J48, *Naive Bayes*, *Multilayer Perceptron*, *Logistic Regression*, *Support Vector Machine* e *Sequential Minimal Optimization*. Gli attributi considerati nei nostri modelli sono i seguenti: *age*, *education.num*, *hour.per.week*, *spouse*, *mid_salary*, *spec_work*, *low_salary*, *tecnical_work*.

Random Forest Il *Random Forest*[4] è un classificatore d'insie-me composto da molti alberi di decisione e dà in uscita la classe che ha ottenuto la percentuale maggiore tra tutte le classi previste dai singoli alberi di decisione presi individualmente. Gli alberi di decisione che compongono il *Random Forest* sono un metodo di apprendimento supervisionato non parametrico utilizzato sia per la classificazione che per la regressione. Esso utilizza tecniche per analizzare la relazione tra una variabile dipendente (target) ed altre variabili indipendenti (covariate di qualsiasi tipo). I vantaggi di questo metodo sono molteplici: innanzitutto richiede una scarsa preparazione dei dati. Si noti tuttavia che questo modulo non supporta i valori mancanti. In secondo luogo questo modello è in grado di gestire sia dati numerici che categoriali. Altre tecniche sono solitamente specializzate nell'analisi di set di dati che hanno un solo tipo di variabile. Infine esegue bene anche se le sue ipotesi sono in qualche modo violate dal vero modello da cui sono stati generati i dati. Uno degli svantaggi di questa tecnica consiste nella creazione di alberi troppo complessi che porta al rischio di non generalizzare bene i dati (*overfitting*). Meccanismi quali la potatura (non attualmente supportato), l'impostazione del numero minimo di osservazioni richiesto su un nodo foglia o l'impostazione della profondità massima dell'albero sono necessari per evitare questo problema. Noi abbiamo dunque deciso di inserire i seguenti parametri: 4 ramificazioni massime, 4 variabili per ogni albero di decisione e un totale di 100 alberi decisionali.

J48 L' algoritmo J48[5] è l'implementazione Weka dell'albero di decisione C4.5. Uno dei vantaggi di questo algoritmo è che si possono classificare sia dati numerici che nominali, ma l'attributo di output deve essere categoriale; inoltre, non sono necessarie assunzioni a priori sulla natura dei dati. Resta il fatto che, purtroppo, molteplici attributi output non sono consentiti e che gli alberi di decisione sono instabili poiché leggere variazioni nei dati di *training* possono produrre differenti selezioni di attributi ad ogni punto di scelta all'interno dell'albero. Abbiamo settato il numero minimo di osservazioni in una foglia pari a 20.

Naive Bayes Il *Naive Bayes*, come tutti i classificatori bayesiani, si basa sull' applicazione del teorema di Bayes. Ciò che viene richiesto per implementare l'algoritmo è la conoscenza

della probabilità condizionata ed a priori relative al problema che, generalmente, non sono note, ma tipicamente stimabili. L' etichetta ai record viene associata con il valore della classe che massimizza la probabilità a posteriori. L' algoritmo può essere utilizzato sia con dati categoriali (ordinali o nominali) sia con dati numerici (discreti e continui) ed è possibile combinarli insieme. Uno degli svantaggi è che gli attributi che descrivono le istanze sono condizionalmente indipendenti data la classificazione (anche se spesso questa ipotesi può essere violata).

Multilayer Perceptron Questo classificatore utilizza dei neuroni artificiali che comunicano unidirezionalmente dagli attributi di input X all' attributo di classe. L' MLP è composto da:

- Neuroni di input associati alle variabili esplicative
- Neuroni nascosti corrispondenti alle diverse trasformazioni che collegano le variabili input all'output implementando il compito di classificazione che si vuole svolgere
- Neurone di output associato alla classe degli attributi etichetta

Ogni neurone di input viene collegato ad ogni neurone presente nello strato nascosto attraverso un link diretto in un'unica direzione: dal neurone di input al neurone dello strato nascosto (direzione del segnale). Una volta che ogni neurone di input ha apportato il segnale, i neuroni dello strato nascosto si attivano. Da quest' ultimi il segnale verrà inviato ai neuroni di output. Non esiste una regola di decisione riguardo al numero di strati nascosti e al numero di neuroni in ognuno di questi strati. Al variare di questi numeri la complessità della nostra architettura varia e ciò si traduce in algoritmi più lenti e computazionalmente onerosi. Si crea una completa connessione tra i neuroni dei diversi strati nascosti: il segnale infatti deve passare dai neuroni di ogni strato, mantenendo la condizione che le informazioni vengano trasferite solamente in un'unica direzione e sotto la condizione di non collegare i neuroni dello stesso strato. Per la scelta del numero degli strati nascosti abbiamo agito di conseguenza: siamo partiti con un solo *hidden layer* ed abbiamo osservato "Precision", "Recall" e "F-measure". Abbiamo continuato ad aggiungere *hidden layer* e neuroni fino a quando le suddette misure di performance hanno raggiunto il valore ottimale. Nel nostro caso sono presenti 3 *hidden layer* contenenti rispettivamente 3, 7 e 10 neuroni.

Logistic Regression La regressione logistica è un caso particolare di modello lineare generalizzato avente come funzione link la funzione *logit*. Si tratta di un modello di regressione applicato nei casi in cui la variabile dipendente sia di tipo dicotomico riconducibile ai valori 0 e 1.

Support Vector Machine Le macchine a vettori di supporto (SVM, dall'inglese *Support Vector Machines*)[6], o macchine *kernel*, sono delle metodologie di apprendimento supervisionato per la regressione e la classificazione di pattern. L'obiettivo è trovare l'iperpiano col massimo margine che divide i due gruppi di punti differenti per il valore assunto dalla variabile risposta in modo che la distanza tra l'iperpiano e il punto più vicino di entrambi i gruppi sia massima.

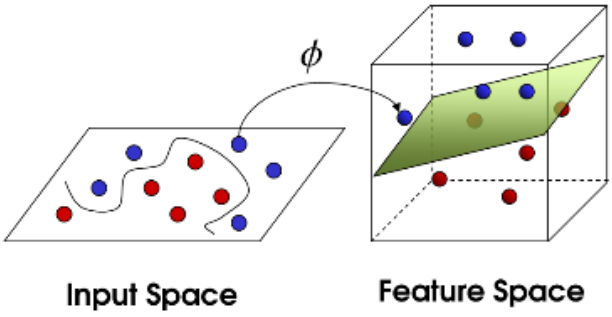

Figura 1. Funzione SVM

Noi ci troviamo nel caso di *non-linearly separable data* ovvero nel caso in cui i nostri dati non possono essere suddivisi in classi mediante una retta. Questo vuole dire che il *learning algorithm* usa la seguente funzione:

$$K(x1,x2) = \phi(x1)\phi(x2)$$

dove K è una funzione di similarità calcolata nello spazio degli attributi originale X ed è denominata *kernel function*. La *Support Vector Machine* è un algoritmo efficiente ed è in grado di rappresentare funzioni non lineari complesse.

Sequential Minimal Optimization *Sequential Minimal Optimization* è un algoritmo per risolvere efficientemente il problema di ottimizzazione che emerge durante l'addestramento di una macchina a vettori di supporto. Questa implementazione sostituisce i *missing values*, trasforma gli attributi nominali in binari ed infine normalizza tutte le variabili esplicative.

2.2 Misure di Valutazione

Per gli Stati Uniti d'America gli individui di maggiore interesse sono coloro che hanno un reddito >\$50.000 in ottica fiscale. La suddetta classe è una classe rara poiché dal nostro dataset appare che solamente il 24% dei cittadini possiede un reddito di questo tipo. Essendo la classe rara definita positivamente possiamo costruire una tabella del tipo:

		Inducer Prediction	
		-1	+1
Actual Class	-1	TN	FP
	+1	FN	TP

Tabella 2. Confusion Matrix

Dove con TN (TP) indichiamo i *true negative* (*true positive*) ovvero la porzione di classe negativa (positiva) predetta

correttamente e con FN (FP) indichiamo i *false negative* (*false positive*) ovvero la porzione di classe negativa (positiva) predetta erroneamente. Attraverso queste quantità possiamo calcolare degli indici che ci consentono di valutare l'effettiva bontà di un modello. Tra questi abbiamo:

- *Precision*: $P = \frac{TP}{TP+FP}$
  Frazione di osservazioni che sono effettivamente positive nel gruppo della classe prevista positiva
- *Recall*: $R = \frac{TP}{TP+FN}$
  Frazione delle osservazioni positive correttamente previste dal modello
- *F-measure*: $F = \frac{2*R*P}{R+P}$
  Media armonica tra Recall e Precision. Un valore alto ci indica che sia Recall che Precision assumono valori alti
- *Accuracy*: $A = \frac{TP+TN}{TP+FP+TN+FN}$
  Percentuale di osservazioni positive e negative predette correttamente

Queste sono le misure che generalmente vengono utilizzate come indici di bontà del modello. Essendo interessati maggiormente alla parte di popolazione avente un reddito alto, non ci siamo basati sull'*Accuracy* poichè non distingue i record positivi da quelli negativi tra quelli predetti correttamente. Classificare come 'povero' un individuo con reddito $>$\$50.000 è l'errore più grave che lo stato possa commettere in ottica fi-scale poiché l'individuo pagherebbe meno tasse del dovuto.
L' errore contrario, invece, farebbe pagare ad un cittadino povero più tasse del dovuto, ma in questo caso il cittadino stesso farebbe ricorso e l'errore verrebbe risolto. Di conseguenza le nostre valutazioni saranno basate principalmente su *Recall* e *F-meausure*.

3. *Hold out*, *Cross Validation* e *Validation*

Per ottenere una valutazione delle performance dei classificatori scelti, abbiamo deciso di utilizzare due differenti metodi: *Hold out* e *Cross Validation*.

3.1 Hold out

Il primo approccio utilizzato è il metodo dell'*Hold out* che si basa sulla partizione del dataset in due sottoinsiemi disgiunti. Il dataset viene scisso seguendo un procedimento di *random sampling* dove abbiamo deciso di suddividere i record in $\frac{2}{3}$ per il *training set* e $\frac{1}{3}$ per il *test set*. La scelta è stata fatta in seguito ad alcune prove dove questa partizione è risultata essere la migliore. Dopo aver applicato l'*Hold out* al dataset non binarizzato è risultato che i classificatori avessero delle performance migliori sul dataset che presenta i dati binarizzati. Il metodo dell'*Hold out* è stato eseguito su tutti e 7 i modelli precedentemente descritti. Utilizzando i risultati dei classificatori ottenuti tramite il nodo "Scorer" ed unendoli, abbiamo potuto confrontare le performance dei modelli attraverso *box plot*, *lift chart* e *ROC curve*. Siccome la nostra attenzione è rivolta principalmente a misure di *Recall* e *F-measure* alte, abbiamo preso in considerazione i modelli con i valori superiori rispetto agli altri.

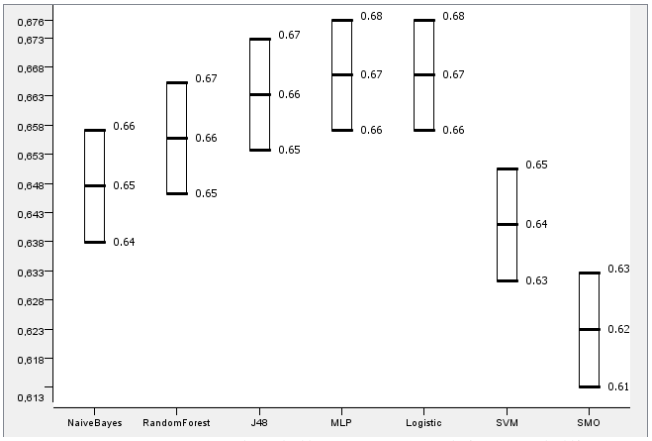

Figura 2. *Box plot* dell'*F-measure* dei 7 modelli

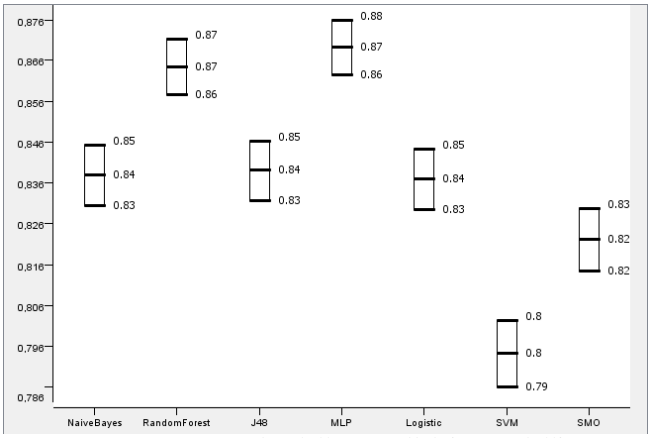

Figura 3. *Box plot* della *Recall* dei 7 modelli

I *box plot* sono stati cotruiti ponendo oltre che la stima della misura di nostro interesse ottenuta per ogni classificatore, anche estremo superiore e inferiore dell'intervallo di confidenza di Wilson a livello di confidenza del 95%[7].

$$\frac{acc+\frac{Z^2_{1-\frac{\alpha}{2}}}{2\cdot N}-Z_{1-\frac{\alpha}{2}}\cdot\sqrt{\frac{acc}{N}-\frac{acc^2}{N}+\frac{Z^2_{1-\frac{\alpha}{2}}}{4\cdot N^2}}}{\left(1+\frac{Z^2_{1-\frac{\alpha}{2}}}{N}\right)},\ \frac{acc+\frac{Z^2_{1-\frac{\alpha}{2}}}{2\cdot N}+Z_{1-\frac{\alpha}{2}}\cdot\sqrt{\frac{acc}{N}-\frac{acc^2}{N}+\frac{Z^2_{1-\frac{\alpha}{2}}}{4\cdot N^2}}}{\left(1+\frac{Z^2_{1-\frac{\alpha}{2}}}{N}\right)}$$

Figura 4. Formula dell'intervallo di confidenza di Wilson

Le *lift chart* e la *ROC curve* non verranno riportate in quanto le differenze sono poco identificabili tra i migliori modelli, viene invece riportata la tabella delle aree sottese dalla curva di ROC.

Applicando questo metodo, i modelli che risultano essere i migliori sono *Random Forest*, J48, MLP e *Logistic Regression*.

Modelli	AUC
Random Forest	0.797
J48	0.799
Naive Bayes	0.788
Multilayer Perceptron	0.806
Logistic Regression	0.801
Support Vector Machine	0.778
SMO	0.768

Tabella 3. Area sotto la curva (AUC)

Essendo questo approccio di semplice applicazione è anche molto probabile introdurre della distorsione dovuta al campionamento effettuato per suddividere le osservazioni nei due sottoinsiemi. Per questo abbiamo deciso di utilizzare anche un secondo approccio.

3.2 *Cross Validation*

L'approccio basato sulla *Cross Validation* permette di valutare le performance dei classificatori partizionando il dataset iniziale in k sottoinsiemi disgiunti dove tutti i record vengono utilizzati sia nel *training set* che nel *test set* grazie alle k iterazioni che vengono effettuate scambiando di volta in volta il sottoinsieme utilizzando come *test set* per k volte. Nel nostro caso k è stato scelto pari a 10 dopo diverse prove. Siccome abbiamo constatato che SVM e SMO sono modelli computazionalmente onerosi sono stati scartati dalla nostra analisi non essendo risultati comunque tra i modelli migliori nell'*Hold out* e considerando i lunghi tempi di applicazione della *Cross Validation*. Anche questo metodo è stato applicato sui dati binarizzati perché si ottenevano delle performance migliori da parte dei classificatori. Attraverso questo se-condo approccio utilizzando lo stesso procedimento del metodo dell'*Hold out* abbiamo confrontato i modelli sulle misure di nostro interesse. Quindi considerando i *box plot* i modelli migliori sono risultati essere *Random Forest*, J48 e MLP.

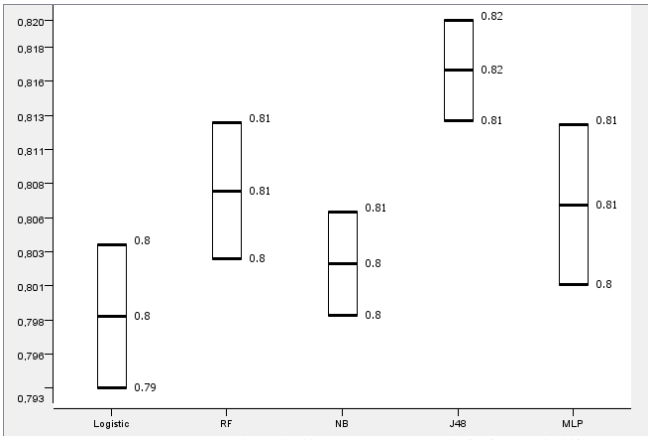

Figura 5. Box plot dell'F-measure dei 5 modelli

Gli stessi modelli risultano essere tra i migliori anche tramite il procedimento dell'*Hold out*. Quindi sono i modelli che abbiamo poi utilizzato nella validazione.

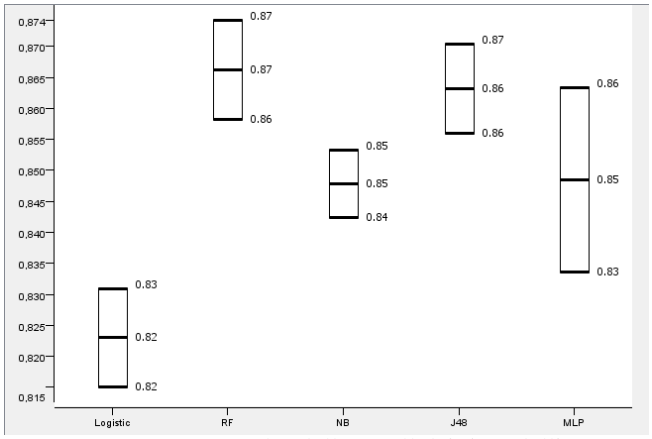

Figura 6. Box plot della recall dei 5 modelli

3.3 *Validation*

In quest'ultima sezione poniamo l'attenzione sulla parte dei dati che non è ancora stata utilizzata. Essa costituisce il *validation set* dove è contenuto il 10% delle osservazioni del dataset originale. Essendo questi record completamente indipendenti dal dataset di partenza, li utilizziamo come fossero nuovi casi da prevedere. I modelli uscenti vincitori dalla fase di test sono, in questo passo, i candidati per effettuare lo *score* sui casi non ancora presi in considerazione e il loro compito è la profilazione degli individui rispetto ai livelli della variabile risposta. Se il modello è generalizzabile, ossia non si specializza sui dati di *training* siamo in presenza di un buon classificatore che è possibile replicare su ulteriori dataset. Come anticipato nella sezione precedente, i modelli che possiamo utilizzare per la validazione sono la *Random Forest*, il J48 e il *Multilayer Perceptron*. Anche in questa fase è stato necessario apportare agli attributi tutte le modifiche già apportate nei *training* e *test set* , nella fase di *preprocessing*, ovvero *feature selection* e binarizzazione degli attributi nominali. Una volta sistemato il dataset, ha avuto inizio l'implementazione dei tre modelli sopracitati per fare previsione. Siamo partiti inserendo in input le 3.162 osservazioni, finora inutilizzate, nei tre modelli migliori (disponibili in Weka) per poi andare a considerare gli output dei relativi nodi "Inducer" e "Scorer": le tre matrici di confusione create indicano che il modello con l'*Accuracy* più alta è MLP poiché classifica correttamente 2.467 osservazioni.

	<=50K	>50K
<=50K	1817	575
>50K	120	650

Tabella 4. Confusion Matrix MLP

Correct classified: 2467
Accuracy: 78,02%

Tuttavia, poiché siamo interessati a predire bene la modalità "positiva" (*income* >50K), abbiamo confrontato le misure della performance dando maggior rilievo a *Recall*. Il modello

che identificava il maggior numero di individui con reddito elevato è risultato il *Random Forest*.

	RF	J48	MLP
Recall	0.862	0.842	0.844
Precision	0.522	0.521	0.531
F-measure	0.650	0.644	0.652
Accuracy	0.774	0.773	0.780

Tabella 5. Performance *measures* dei 3 modelli

Avendo presupposto normale ciascuna misura, per avere risultati statisticamente significativi, abbiamo calcolato anche i relativi intervalli di confidenza di Wilson.

	RF	J48	MLP
Recall	0.862	0.842	0.844
Low est	0.855	0.834	0.837
Sup est	0.869	0.849	0.851

Tabella 6. Intervallo di confidenza per *Recall* dei 3 modelli

In aggiunta a ciò, abbiamo confrontato le diverse probabilità di corretta classificazione degli eventi al variare dell'errata classificazione dei non-eventi (*income* <=50K) tramite la *Roc curve*. Anche qui, come nei due casi precedenti, la probabilità di prevedere bene coloro che possiedono un reddito elevato è più alta nel modello RF, anche se il distacco non è così evidente.

	Area Under Curve
P(*income* = >50K) - RF	0.804
P(*income* = >50K) - J48	0.796
P(*income* = >50K) - MLP	0.802

Tabella 7. Aree sotto la curva (AUC) riferite ai 3 modelli

In ultima analisi, considerando i *box plot* relativi ad ogni misura della performance, attraverso il *Random Forest* la modalità "positiva" viene predetta nel modo più corretto. Nella figura 7 presentiamo la *Recall* che per la RF risulta significativamente maggiore rispetto a quella degli altri due classificatori.

Conclusioni

In conclusione, mettendoci nell'ottica fiscale del governo statunitense, un modello come il *Random Forest*, che con una probabilità dell'86% classifica correttamente i cittadini come lavoratori aventi reddito elevato, si rivela utile affinché le manovre finanziarie siano eque. Un cittadino americano situato nella fascia di reddito elevato che viene profilato dal modello come appartenente alla classe di reddito >$50.000, si attenderà una certa somma di contributi da pagare, proporzionale al reddito. Caso contrario, un cittadino americano situato nella fascia di reddito basso che viene profilato dal modello come appartenente alla classe di reddito <$50.000, si attenderà di pagare meno tasse di quante ne paghi il cittadino con reddito elevato. Tuttavia, se il modello

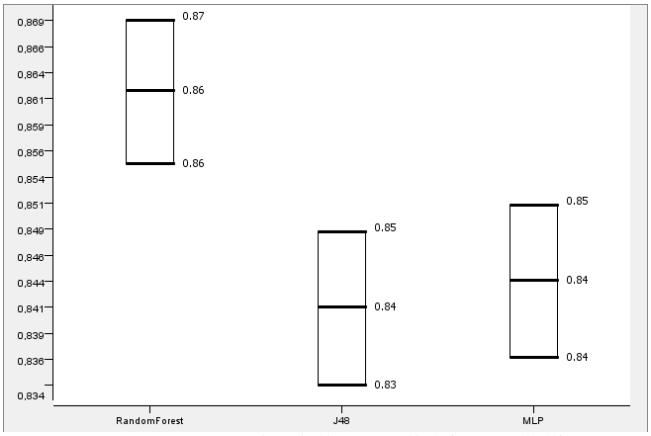

Figura 7. *Box plot* della *Recall* dei 3 modelli

dovesse fallire classificando un individuo che guadagna più di $50.000 annui nella fascia bassa, allora, ingiustamente, questo pagherebbe meno del dovuto e continuerebbe a farlo approfittando dell'errore e sconfinando nella frode. Dunque, sulla base della nostra analisi, per un futuro censimento statunitense si potrebbe pensare di utilizzare tale modello che con una probabilità dell'86% stabilisce tra chi ha un reddito >$50.000 dichiara il vero. Il *Random Forest*, difatti, riuscirebbe a garantire, con un errore del solo 14% la veridicità dell'applicazione. Un possibile sviluppo futuro potrebbe essere quello di collaudare in maniera ottimale tale classificatore sulla totalità dei dati estratti da ogni censimento passato così da poter meglio prevedere, per le dichiarazioni dei redditi future, la quota di cittadini con reddito elevato che dichiara il falso.
Questa operazione è fondamentale poichè riesce a punire gli evasori. Inoltre, nel caso in cui l'algoritmo classificasse erroneamente un individuo con reddito basso, quest'ultimo farebbe ricorso e l'errore verrebbe risolto. Al contrario, se l'errata classificazione riguardasse l'individuo con reddito elevato, l'errore non verrebbe mai colmato poichè risulterebbe vantaggioso per il cittadino stesso.

Riconoscimenti

Ringraziamo il Prof. Stella per aver reso disponibile il materiale degli argomenti svolti a lezione oltre a libri di testo e articoli aggiuntivi che ci sono tornati utili per la stesura di questo articolo. Un riconoscimento dovuto va anche agli ideatori dei software Knime e R che ci hanno permesso di sviluppare il *workflow* su cui si è basata la nostra analisi.

Riferimenti

[1] `https://www.census.gov/en.html`.

[2] `https://www.kaggle.com/uciml/adult-census-income`.

[3] Brownlee J. 8 tactics to combat imbalanced classes in your machine learning dataset. `https://machinelearningmastery.com/`, 2015.

[4] Polamuri S. Introduction to random forest algorithm. http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing, 2017.

[5] Mathuria M. Machine learning for language technology. http://stp.lingfil.uu.se/~santinim/ml/2016/Lect_03/Lab02_DecisionTrees.pdf, 2016.

[6] Burges. *A tutorial on support vector machines for pattern recognition.* C.J.C., 1998.

[7] Tan P.-N. Steinbach M. Kumar V. Introduction to data mining. http://www.uokufa.edu.iq/staff/ehsanali/Tan.pdf, 2006.