Studio del reddito: un'analisi predittiva per la ricerca delle frodi fiscali

Andrea Piancone[1], Marcello Pichini, [1], Valentina Nelli[1], Mariam Savadogo[1], Ardalan Mehraram[1]

Sommario
Dichiarare un reddito inferiore a quello effettivo è un illecito che negli Stati Uniti comporta oggi anno perdite per le casse erariali di diversi miliardi di dollari. Pertanto, è interesse dell'autorità fiscale riuscire a sviluppare sistemi in grado di identificare in maniera efficace i possibili evasori. Di fatto, la riscossione dei tributi è uno strumento fondamentale per redistribuire la ricchezza riducendo così le disparità economiche e sociali presenti all'interno del tessuto sociale. Alla luce di queste considerazioni ci si è posti la seguente domanda: è possibile prevedere il reddito di un cittadino sulla base delle informazioni raccolte tramite censimenti al fine di aiutare le autorità fiscali ad individuare i possibili evasori? Quindi, supposti veri i dati a disposizione, dopo un'accurata fase di valutazione e comparazione di diversi modelli, sono stati individuati i metodi con le migliori performance al fine di rispondere alla domanda di ricerca che ci si è posti. I risultati potranno costituire un valido strumento di supporto per l'autorità fiscale nel complesso ed importante compito di individuazione dei possibili evasori.

Keywords
Machine Learning — Dati sbilanciati — Evasione fiscale — Reddito

[1] *Dipartimento di Informatica, Università degli Studi di Milano-Bicocca, CdLM: Data Science*

Indice

Introduzione

Per rispondere alla domanda di ricerca è stato analizzato il dataset *Adult Census Income*, disponibile sulla piattaforma Kaggle. Esso contiene i dati relativi al censimento svolto nel 1994 negli Stati Uniti. Si compone di 32561 records, dove ogni record rappresenta un individuo censito, descritto da una serie di fattori sociali e demografici. In particolare, su ogni persona censita sono stati rilevati i seguenti attributi:

- *Age*: età del cittadino;
- *Workclass*: classe lavorativa del cittadino;
- *Fnlwgt*: attributo che esprime con quale misura l'individuo considerato rappresenta la propria classe sociale;
- *Education*: livello di istruzione del cittadino;
- *Education.num*: livello di istruzione del cittadino, ma rappresentato in termini numerici;
- *Marital.status*: stato civile del cittadino;
- *Occupation*: lavoro svolto dal cittadino;
- *Relationship*: ruolo del cittadino all'interno del nucleo familiare;
- *Race*: etnia del cittadino;
- *Sex*: sesso del cittadino;
- *Captal.gain*: utili di capitali;
- *Capital.loss*: perdita di titoli di capitale;
- *Hours.per.week*: ore di lavoro settimanali;
- *Native.country*: nazione di nascita dell'individuo;

- *Income*: classe di reddito del cittadino.

La variabile target dell'analisi è *income*, che presenta due modalità: reddito maggiore di 50 000 dollari e reddito minore o uguale di 50 000 dollari. La prima viene considerata come classe positiva in quanto considerata d'interesse per l'analisi, mentre la seconda viene considerata come classe negativa. L'analisi è articolata nel seguente modo: dopo una prima fase di preprocessing si è proceduto con l'impiego di diversi modelli di classificazione tramite i metodi holdout e cross validation, accompagnati da tecniche di feature selection e metodi basati sul ricampionamento al fine di ridurre le distorsioni causate dall'attributo di classe sbilanciato. L'analisi si è conclusa con l'utilizzo del Cost Sensitive Learning realizzato con il metodo brute force.

1. Preprocessing

Prima di procedere con l'implementazione delle tecniche di Machine Learning, è stata eseguita una fase di preprocessing.

1.1 Valori mancanti

Dalle statistiche descrittive è emerso che sono presenti 4262 valori mancanti, di cui 1836 nell'attributo *workclass*, 1843 nell'attributo *occupation* e 583 nell'attributo *native.country*. Si è deciso di rimuovere i records che presentavano almeno un valore mancante su un attributo in forza di due ragioni. La prima consiste nell'elevato numero records di cui si compone il dataset, infatti solo 2399 records (circa il 7% del dataset originario), presentano valori mancanti. La seconda motivazione consta nel rischio concreto di alterare in maniera significativa i dati, tramite una sostituzione con la tecnica del *most frequent value*, attribuendo ad alcuni individui valori non veritieri.

1.2 Rimozione attributi superflui

Successivamente, la fase di preprocessing è proseguita con l'eliminazione di due attributi superflui: *education* e *fnlwgt*. Il primo è ridondante in quanto replica le medesime informazioni contenute nell'attributo *education.num*, giacché si è deciso di conservare ai fini dell'analisi quest'ultimo per esprimere numericamente il livello di istruzione del cittadino, scartando l'attributo *eudcation*. Il secondo invece mostra il grado di rappresentatività della propria classe sociale da parte del cittadino considerato. Questo grado è espresso tramite il numero di rilevatori del censimento che credono che l'osservazione rappresenti la propria classe. Questa feature e stata scartata direttamente, e non sarà più considerata.

1.3 Feature transformation

Infine, la fase di preprocessing si è conclusa con l'aggregazione delle modalità delle variabili *native.country*, *education.num* e *occupation*. Il raggruppamento delle modalità di tali variabili è stata dettata dall'elevata numerosità delle modalità stesse, riducendo così la variabilità. Le modalità della variabile *native.country* sono state raggruppate in due gruppi: *USA* se il cittadino è nativo degli Stati Uniti, *other* se il cittadino proviene da un'altra nazione del mondo. In questo modo le modalità sono state ridotte da 41 a 2.

Modalità	**%>50K**	**Gruppo**
Exec-managerial	48.52%	High-income-occupation
Prof-specialty	44.85%	
Protective-serv	32.61%	Middle-income-occupation
Tech-support	30.48%	
Sales	27.06%	
Crafr-repair	22.53%	
Transport-moving	20.29%	
Adm-clerical	13.38%	Middle-low-income-occupation
Machine-op-inspct	12.46%	
Farming-fishing	11.63%	
Armed-Forces	11.11%	
Handlers-cleaners	6.15%	Low-income-occupation
Other-service	4.11%	
Priv-house-serv	0.70%	

Tabella 1. Aggregazione delle modalità della variabile occupation

In merito ad *education.num*, per aggregarne le modalità si è adottato il seguente criterio:

- i valori di *education.num* al massimo pari a 8 sono stati raggruppati in gruppo chiamato *Dropout*. In tale gruppo rientrano i cittadini che hanno abbandonato gli studi;
- per *education.num* uguale a nove si è creato un unico gruppo chiamato *High-school-graduate*, nel quale rientrano i cittadini censiti che hanno concluso il ciclo di istruzione secondaria;
- per *education.num* uguale a dieci si è creato un unico gruppo chiamato *Some-college*, nel quale rientrano i cittadini che al momento del censimento frequentavano un college senza averlo ancora terminato o che hanno frequentato un college in passato, ma senza terminarlo;
- i valori di *education.num* pari undici o dodici sono stati raggruppati in nuovo gruppo chiamato *Associate-graduate*;
- i valori di *education.num* almeno pari a tredici sono stati raggruppati in gruppo chiamato; *Bachelor-degree-or-higher*, in cui rientrano tutti i cittadini che hanno conseguito un titolo di laurea triennale o superiore.

Così facendo le modalità sono state ridotte da 16 a 5.
Infine, per aggregare le modalità dell'attributo *occupation* non si è riusciti a trovare una fonte di conoscenza esterna da sfruttare per l'operazione di aggregazione. Pertanto si è deciso di sfruttare la variabile di risposta, calcolando per ogni modalità di *occupation* la percentuale di classe rara all'interno della modalità stessa. Il risultato di questa operazione è rappresentato in Tabella 1.

2. Dataset sbilanciato

Il dataset è caratterizzato da una distribuzione sbilanciata della variabile target, in quanto la classe positiva è rappresentata

solo dal 24% del totale. Per ridurre le distorsioni causate dallo sbilanciamento della variabile di classe sono stati impiegati due approcci: *euqal size undersampling* e *cost sensitive learning*. La prima è una tecnica che consiste nel mantenere tutti i records della classe rara, per poi selezionare in maniera casuale un uguale numero di records della classe abbondante, ottenendo quindi un nuovo dataset bilanciato. Questo metodo, nonostante la sua semplicità, presenta una serie di problematiche legate all'eliminazione di dati che potrebbero rivelarsi utili nella fase di classificazione. La seconda tecnica prevede l'utilizzo della matrice dei costi che consente di assegnare ad ogni istanza della matrice di confusione uno specifico peso.

3. Modelli e misure di performance

3.1 Modelli

Nel seguente lavoro sono state applicate diverse tecniche di classificazione al fine di individuare quella migliore sulla base dei dati disponibili. In particolare:

- *Modelli euristici*: tra questi modelli si è scelto di adottare l'albero di decisione **J48** e il classificatore **Random Forest**, entrambi implementati da Weka;
- *Modelli di regressione*: tra i modelli appartenenti a questa famiglia si è deciso di applicare la **regressione logistica**;
- *Modelli di separazione*: tra i diversi classificatori appartenenti a questa categoria si è scelto di utilizzare due tipi di **Support Vector Machine**, quali **SPegasos** e **SMO poly**, che sfrutta come funzione di kernel un polykernerl;
- *Modelli probabilistici*: sfruttano il teorema di Bayes, tra i classificatori appartenenti a questa famiglia sono stati utilizzati i modelli **Naive Bayes** e **NBTree**.

3.2 Misure di performance

Come anticipato nella Sezione 2, il dataset presenta un attributo di classe sbilanciato. A causa di questo sbilanciamento, l'Accuracy non è sufficiente a valutare adeguatamente i modelli di classificazione. Pertanto essa deve essere accompagnata da altre misure performance quali Precision, Recall, F_1-measure e AUC (Area Under Curve). Di seguito si descrivono brevemente le sopracitate misure di performance.
L'Accuracy indica la percentuale di records positivi e negativi classificati correttamente. Essa è data dalla seguente formula:

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

Dove TP e TN indicano rispettivamente il numero di istanze positive e negative classificate correttamente, mentre FP e FN indicano rispettivamente il numero di istanze positive e negative classificate erroneamente.
L'indicatore di Precision rappresenta la percentuale di records positivi che sono realmente positivi tra tutti quelli predetti come tali. La Precision è data dalla seguente formula:

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

Un valore elevato della Precision comporta pochi falsi positivi. L'indicatore di Recall, invece, rappresenta la percentuale di records positivi correttamente classificati dal modello. Esso è dato dalla seguente formula:

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

Un alto valore di Recall, implica che pochi records positivi sono stati erroneamente classificati dal modello. Tuttavia, in alcuni casi, queste ultime due misure sono tra loro in conflitto: al crescere dei veri positivi della classe rara migliora la Recall, ma questo potrebbe portare ad un peggioramento della Precision a causa di un incremento dei falsi positivi. Per evitare questo problema si utilizzano misure alternative come la F_1-measure, data dalla media armonica tra Precision e Recall. Tale metrica è data dalla seguente formula:

$$F_1 - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Un valore elevato della F_1-measure garantisce indicativamente che sia la Recall che la Precision siano elevate. Un ulteriore strumento per misurare la performance di un modello di classificazione consiste nella curva ROC, che mette in relazione la percentuale di falsi positivi con quella di veri positivi. Dalla curva ROC è possibile ricavare l'AUC, ossia l'area sottesa alla curva ROC. Maggiore è il valore di tale metrica, migliore è il modello.
In generale questi sono gli indici che vengono impiegati per valutare la bontà di un classificatore. Come già anticipato, l'Accuracy nel caso in cui la variabile target è sbilanciata non è un buon indicatore in quanto non attua alcuna distinzione tra record positivi e negativi, tra quelli predetti correttamente. In particolare, in questo lavoro, si possono commettere due errori: classificare un individuo con un reddito superiore ai 50 000 dollari nella classe opposta (in questo caso si avrebbe un falso negativo) oppure classificare un individuo con un reddito minore o uguale a 50 000 dollari nella classe opposta (in questo caso si avrebbe un falso positivo).
Il primo errore è più grave, in quanto potrebbe portare lo Stato a non identificare un potenziale evasore fiscale, il quale continuerebbe a mantenere il suo comportamento fraudolento. Al contrario, il secondo errore è meno grave in quanto significa classificare erroneamente un individuo con un reddito inferiore ai 50 000 dollari. Tale errore tuttavia, verrebbe facilmente individuato in quanto il cittadino presenterebbe ricorso, a fronte di una cartella esattoriale errata. Tuttavia, per lo Stato ci sarebbero una serie di costi legati ad un eventuale contenzioso tributario.
Pertanto, la valutazione della bontà dei classificatori si baserà principalmente sull'indice di Recall e F_1-measure.

4. Analisi e risultati

4.1 Classificazione con metodo holdout

In questa prima fase è stato applicato il metodo holdout che si basa sulla scomposizione del dataset in due sottoinsiemi esaustivi ed esclusivi, tramite una procedura di campionamento stratificato, dove l'attributo di stratificazione è *income*. Grazie a questa procedura si è ottenuto il training set (67% dei records) e il test set (33% dei records). I classificatori precedentemente descritti, sono stati addestrati con il training set e validati con il test set. I risultati così ottenuti sono riportati nella Tabella 2.

Classificatore	Recall	Precision	F_1-measure	Accuracy	AUC
J48	0.605	0.769	0.677	0.856	0.884
Random Forest	0.601	0.694	0.644	0.835	0.865
Logistic	0.600	0.735	0.660	0.846	0.901
SMO poly	0.567	0.747	0.644	0.844	0.752
SPegasos	0.495	0.750	0.596	0.833	0.720
Naive Bayes	0.458	0.731	0.564	0.823	0.887
NBTree	0.654	0.754	0.700	0.861	0.911

Tabella 2. Classificatori con holdout

In particolare si osserva che:

- tutti i classificatori presentano valori inferiori della Recall rispetto alla Precision, sintomo che i modelli tendono a classificare negativamente i records in maniera eccessiva;
- i valori di F_1-measure sono sistematicamente inferiori alle relative Accuracy, ma ne seguono l'andamento. Questo non sorprende a causa dello sbilanciamento della variabile di risposta;
- in termini di AUC tutti i classificatori sono caratterizzati da valori generalmente elevati, ad eccezione dei modelli **SMO poly** e **SPegasos**, i quali assumono valori significativamente inferiori rispetto agli altri che modelli.

4.2 Feature selection ed equal size sampling

Al fine di ridurre le distorsioni generate dalla presenza di un attributo di classe sbilanciato si è deciso di accompagnare i metodi impiegati nella Sezione 4.1 con tecniche basate sul campionamento. In particolare si è scelto di adottare la tecnica nota come *equal size undersampling* (Sezione 2). Una volta scomposto il dataset in training set (67% dei records) e test set (33% dei records), si è applicato solo al primo l'equal size undersampling. Successivamente, con lo scopo di ridurre il numero di attributi del training set ed individuare i più significativi si è applicata una procedura di feature selection. In particolare si è deciso di utilizzare il filtro multivariato CfsSubsetEval, grazie al quale è possibile individuare quali sono gli attributi che maggiormente influenzano la variabile di classe, senza trascurare la correlazione tra gli stessi. Gli attributi così selezionati sono: *age*, *education.num* (modificato), *marital.status*, *occupation* (modificato), *relationship*, *capital.gain*, *capital.loss*, *hours.per.week*. Nella Tabella 3 sono riportate le misure di performance ottenute con tale approccio.

Classificatore	Recall	Precision	F_1-measure	Accuracy	AUC
J48	0.868	0.547	0.671	0.788	0.888
Random Forest	0.787	0.546	0.645	0.784	0.864
Logistic	0.831	0.557	0.667	0.794	0.897
SMO poly	0.847	0.531	0.653	0.776	0.800
SPegasos	0.845	0.530	0.651	0.775	0.798
Naive Bayes	0.564	0.707	0.628	0.833	0.887
NBTree	0.849	0.577	0.687	0.808	0.909

Tabella 3. Equal size sampling con CfsSubsetEval e holdout

In generale i classificatori registrano valori elevati dell'indice di Recall e valori ridotti dell'indice di Precision. Sintomo che tendenzialmente i classificatori portano ad numero contenuto di falsi negativi, mentre commettono un numero elevato di falsi positivi. L'unica eccezione è Naive Bayes, il quale presenta presenta il valore più elevato in termini di Precision e Accuracy, ma il più basso in termini di Recall, sintomo di una sua incapacità a classificare la classe rara.
Poiché fra attributi selezionati dal filtro multivariato CfsSubsetEval ne sono presenti alcuni categoriali (*education.num*, *marital.status*, *occupation* e *relationship*), si è deciso di addestrare i classificatori con tali attributi binarizzati. La binarizzazione viene realizzata tramite il nodo **One to Many** grazie al quale è possibile creare un attributo binario per ciascuna modalità del carattere qualitativo. Tuttavia, i risultati ottenuti non sono particolarmente differenti rispetto a quelli ottenuti senza binarizzazione, pertanto non sono riportati ulteriormente.
Il metodo holdout, dipendendo dalla scelta del test, può portare a sottostimare o a sovrastimare la performance di un classificatore. Per superare questo limite si è scelto di ricorrere all'uso del 10 folds cross validation, che divide il dataset in dieci partizioni di approssimativamente uguale numerosità, ognuna delle quali viene utilizzata nove volte come training set ed una volta come test set. Le misure di performance che si ottengono alla fine del processo sono una media aritmetica delle misure ottenute ad ogni iterazione. Le misure di performance così registrate sui classificatori sono riportate in Tabella 4.

Classificatore	Recall	Precision	F_1-measure	Accuracy	AUC
J48	0.834	0.564	0.673	0.798	0.886
Random Forest	0.794	0.549	0.649	0.786	0.870
Logistic	0.834	0.566	0.675	0.800	0.899
SMO poly	0.846	0.532	0.653	0.777	0.800
SPegasos	0.844	0.531	0.652	0.776	0.799
Naive Bayes	0.520	0.697	0.596	0.824	0.884
NBTree	0.876	0.559	0.683	0.797	0.911

Tabella 4. Equal size sampling con CfsSubsetEval e Cross Validation

L'addestramento dei classificatori tramite cross validation non ha portato a risultati particolarmente differenti rispetto al metodo holdout in termini di AUC e F_1-measure. Per quanto riguarda la Recall, gli unici classificatori su cui si registrano

differenze rilevanti tra il metodo holdout (Tabella 3) e la cross validation (Tabella 4) sono **J48** e **NBTree**. In particolare, il metodo holdout porta a registrare sul classificatore **J48** un valore dell'indice di Recall superiore rispetto a quello ottenuto tramite cross validation, sintomo che l'holdout commette una sovrastima di tale indice. Sul classificatore **NBTree** si verifica la situazione opposta, sintomo quindi di una sottostima da parte del metodo holdout per l'indice di Recall.

4.3 Validazione e intervalli di confidenza

Al fine di svolgere un'analisi più approfondita per individuare quale sia il miglior classificatore, si è deciso di dividere il dataset in due partizioni: Partizione A (80% dei records) e Partizione B (20% dei records). Per ottenere tali partizioni si è fatto ricorso ad un procedimento di stratified sampling, dove l'attributo di stratificazione è *income*. In seguito si è divisa la Partizione A in training set (67% dei records) e test set (33% dei records), sempre tramite una procedura di campionamento stratificato. I classificatori sono stati addestrati sul training set della Partizione A e validati sia sul test set della Partizione A, che sulla Partizione B. In questa fase dell'analisi si è focalizzata l'attenzione sugli indici di Recall e F_1-measure, in quanto considerati più interessanti ai fini dell'analisi. Si è deciso di ignorare il classificatore **Naive Bayes** a causa delle sue scarse performance in termini di Recall. Con lo scopo di comparare i classificatori in termini di Recall si è realizzato un line plot (Figura 1), grazie al quale è possibile mostrare sullo stesso grafico i valori dell'indice di Recall riscontrati sulle due partizioni. In Figura 1, sulla sinistra è presente il valore relativo alla partizione A, mentre sulla destra è presente il valore relativo alla partizione B.

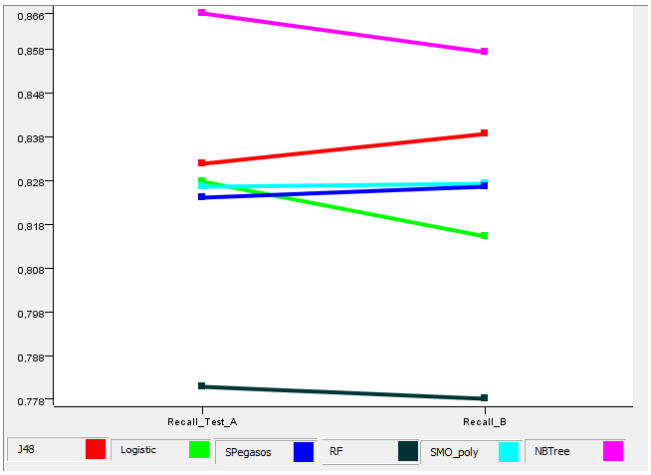

Figura 1. Recall sulle due partizioni

Dalla Figura 1 si osserva che **NBTree**, in termini di Recall domina gli altri classificatori su entrambe le partizioni, seguito dal classificatore **J48**. Fra i classificatori **SPegasos**, **Logistic** e **SMO poly** non si riesce ad identificare un classificatore migliore. Infine, **Random Forest** presenta le peggiori performance in termini di Recall su entrambe le partizioni. Nella Tabella 5 sono riportati i valori di Recall e la differenza tra le due partizioni.

Classificatore	Recall Test A	Recall B	Differenza
J48	0.832	0.839	-0.007
Random Forest	0.781	0.778	0.003
Logistic	0.828	0.816	0.012
SMO poly	0.827	0.828	-0.001
SPegasos	0.824	0.827	-0.003
NBTree	0.866	0.858	0.008

Tabella 5. Confronto valori di Recall

Con lo scopo di indagare le differenze visibili sia dalla Tabella 5 che dalla Figura 1 si è deciso di calcolare l'intervallo di confidenza sulla Recall riscontrata nella prima partizione, per poi confrontare l'intervallo di confidenza con la Recall registrata sulla seconda partizione. L'intervallo di confidenza è stato calcolato secondo Wilson, con un livello di confidenza del 99%.
Gli intervalli di confidenza sulla prima partizione e i valori della Recall sulla seconda partizione sono rappresentati nella Figura 2.

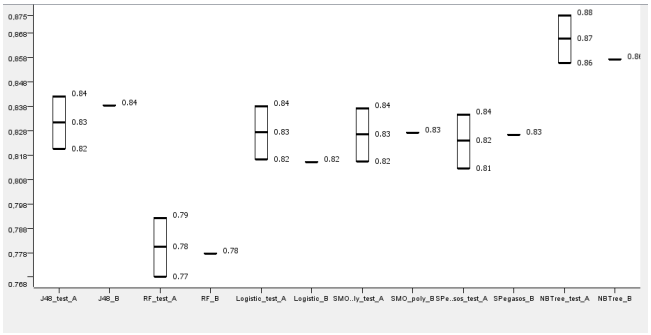

Figura 2. Intervalli di confidenza per l'indice di Recall

Dall'osservazione della Figura 2 si nota che per il modello **Logistic**, la Recall ottenuta sulla Partizione B non ricade nell'intervallo di confidenza stimato sulla Recall riscontrata sul test set della Partizione A. Infatti il line plot (Figura 1) per questo modello mostra la linea con la pendenza più elevata. Per gli altri modelli, il valore della Recall registrato sulla Partizione B è compreso all'interno dell'intervallo di confidenza stimato sull'indice di Recall registrato sul test set della Partizione A. Infatti, il line plot (Figura 1) mostra per gli altri classificatori una linea con una pendenza moderata. Infine, sempre dalla Figura 2, si può concludere che in base alla Recall, il miglior classificatore risulta essere **NBTree** seguito da **J48**.
Successivamente, si è mantenuto il medesimo impianto metodologico per la F_1-measure. In Tabella 6 sono riportate le misure di F_1-measure registrate sia sul test set della Partizione A che sulla Partizione B. In Figura 3 sono rappresentati gli intervalli di confidenza costruiti sulla F_1-measure registrata sul test set della Partizione A e il valore registrato sulla Partizione B.
Dalla Figura 3 si osserva che per tutti i classificatori in esame

Classificatore	F_1-measure Test A	F_1-measure B	Differenza
J48	0.672	0.671	0.001
Random Forest	0.641	0.643	-0.002
Logistic	0.667	0.663	0.004
SMO poly	0.635	0.647	-0.012
SPegasos	0.636	0.648	-0.012
NBTree	0.669	0.667	0.002

Tabella 6. Confronto valori di F_1-measure

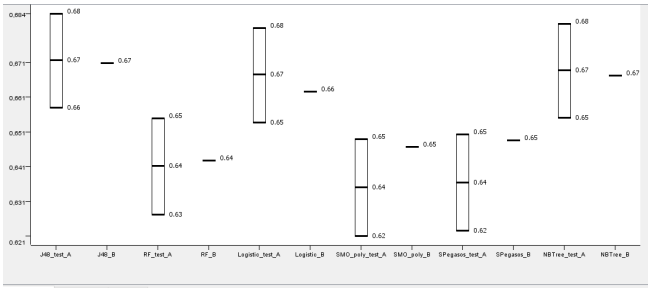

Figura 3. Intervalli di confidenza per l'indice di F_1-measure

i valori dell'indice di F_1-measure riscontrati sulla Partizione B, ricadono all'interno dell'intervallo di confidenza costruito sull'indice di F_1-measure registrato sul test set della Partizione A. Questo risultato è in linea con le piccole differenze mostrate all'interno della Tabella 6. Infine dalla Figura 3 si può concludere che i migliori classificatori risultano essere **J48** e **NBTree**, pertanto i risultati sono in linea con l'indice di Recall.

4.4 Cost Sensitive Learning

Il secondo metodo impiegato per contrastare le distorsioni causate dalla classe sbilanciata consiste nell'analisi dei costi. Per svolgere tale analisi si è usato il nodo weka **CostSensitiveClassifier**. L'analisi dei costi è stata accompagnata dall'uso del 3 folds cross validation. Questo numero di fogli di cross validazione è stato scelto sia per garantire la scalabilità dell'algoritmo, che a causa delle differenze trascurabili con i risultati ottenuti utilizzando dieci o cinque fogli di cross validazione. Inoltre, si è deciso di utilizzare un approccio brute force, con cui si provano diverse matrici di costo, al fine di individuare quale è quella che si adatta meglio ai dati. In particolare si è deciso di provare la seguente coppia di matrici di costo:

$$A = \begin{bmatrix} 0 & 2 \\ 6 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -1 & 2 \\ 12 & 0 \end{bmatrix}$$

In entrambe le matrici si è scelto di assegnare un costo superiore ai falsi negativi, rispetto ai falsi positivi. Il falso negativo è stato considerato un errore più grave, in quanto significa classificare erroneamente un individuo con un reddito superiore ai 50 000 dollari e quindi tale errore potrebbe portare lo Stato a non identificare un potenziale evasore fiscale, il quale continuerebbe a mantenere il suo comportamento fraudolento. Al contrario, un falso positivo è un errore meno grave in quanto significa classificare erroneamente un individuo con un reddito inferiore ai 50 000 dollari. Tale errore tuttavia verrebbe facilmente individuato in quanto il cittadino presenterebbe ricorso, a fronte di una cartella esattoriale errata. Tuttavia, per lo Stato ci sarebbero una serie di costi legati ad un eventuale contenzioso tributario.
In Tabella 7 sono riportati i risultati ottenuti utilizzando la matrice di costo A e la matrice di costo B.

Classificatori	Recall	Precision	F_1-measure	Accuracy	AUC
Matrice di costo A					
J48	0.803	0.561	0.661	0.795	0.798
Random Forest	0.778	0.570	0.658	0.799	0.792
Logistic	0.839	0.573	0.681	0.804	0.816
SMO poly	0.555	0.747	0.637	0.843	0.747
SPegasos	0.525	0.747	0.616	0.837	0.733
Naive Bayes	0.570	0.682	0.621	0.827	0.741
NBTree	0.806	0.628	0.706	0.833	0.824
Matrice di costo B					
J48	0.832	0.542	0.656	0.783	0.799
Random Forest	0.825	0.525	0.641	0.770	0.789
Logistic	0.888	0.537	0.669	0.781	0.817
SMO poly	0.555	0.747	0.637	0.843	0.747
SPegasos	0.525	0.747	0.616	0.837	0.733
Naive Bayes	0.600	0.668	0.632	0.826	0.751
NBTree	0.857	0.592	0.700	0.817	0.831

Tabella 7. Cost sensitive learning con Matrice di costo A e Matrice di costo B

Dalla Tabella 7 si osserva che in generale la seconda matrice di costo ha portato a valori dell'indice di Recall migliori rispetto alla prima matrice di costo, ad eccezione dei modelli SVM, per i quali tale misura è rimasta invariata. In particolare, concentrandosi sulla seconda matrice di costo si può osservare che:

- i classificatori **SPegaosos** e **SMO poly** presentano i più alti valori di Accuracy, ma i più bassi di Recall. Sintomo di incapacità a classificare la classe minoritaria. Questo è in contrasto con i risultati ottenuti sia tramite il metodo holdout che con la cross validation (Sezione 4.2), che avevano portato a registrare su tali modelli ottimi valori della Recall;

- **Naive Bayes**, continua a presentare valori ampiamente insufficienti dell'indice di Recall, in accordo con le analisi condotte nella Sezione 4.2;

- i classificatori **J48**, **Logistic**, **Random Forest** e **NBTree**, presentano ottimi valori dell'indice di Recall. In particolare, confrontando i risultati ottenuti nella Sezione 4.2 si nota che **J48** presenta un valore dell'indice di Recall inferiore rispetto a quello ottenuto con il metodo holdout (Tabella 3), ma è molto vicino a quello ottenuto tramite cross validation. In merito a **NBTree** si registrano valori inferiori rispetto a quelli mostrati in Tabella 3

e Tabella 4. Infine, **Random Forest** e **Logistic** hanno ottenuto risultati decisamente superiori grazie al Cost Sensitive Learning rispetto a quelli ottenuti con l'equal size undersampling.

In conclusione, il Cost Sensitive Learning implementato tramite le matrici di costo precedentemente descritte, si è rivelato un approccio di successo per i classificatori **J48**, **Random Forest**, **Logistic** e **NBTree**, sui quali sono stati registrati ottimi valori dell'indice di Recall. Al contrario, per i classificatori **SPegasos** e **SMO poly** tale approccio non ha portato a risultati soddisfacenti. Questo può essere dovuto al fatto che le matrici di costo che sono state utilizzate non sono ottimali per questi modelli, altra causa può essere una mancata feature selection che può avere inificiato le performance di tali modelli. In generale, l'approccio Cost Sensitive Learning per una sua applicazione ottimale richiede il supporto di un esperto di dominio, in grado di individuare i costi più realistici per le istanze della matrice di confusione.

Conclusioni

Al fine di rispondere alla domanda di ricerca, in primo luogo si è proceduto con un'analisi utilizzando tutte le features senza ribilanciare il training set. Successivamente, a causa delle performance deludenti, si è effettuata un'analisi ribilanciando il training set ed implementando una feature selection con il filtro multivariato CfsSubsetEval. Questo secondo approccio ha portato a registrare ottimi valori dell'indice di Recall su tutti i classificatori ad eccezione di **Naive Bayes**. Con lo scopo di approfondire l'analisi si è deciso di dividere il dataset in tre partizioni: A_train, A_test e B, procedendo al calcolo degli intervalli di confidenza per la Recall e la F_1-measure. Da questa ulteriore analisi è emerso che i migliori classificatori sono **NBTree** e **Naive Bayes**. Su questi modelli sono stati registrati alti valori di Recall, ma bassi valori di Precision, sintomo che tali modelli tendono a classificare erroneamente pochi individui con reddito superiore a 50 000 dollari, ma tendono a classificare erroneamente in maniera eccessiva gli individui con un reddito inferiore a 50 000 dollari, etichettandoli come cittadini aventi un reddito superiore a tale soglia.
Infine, si è utilizzato del Cost Sensitive Learning come approccio alternativo per ridurre gli effetti distorsivi dell'attributo di classe sbilanciato. In particolare sono state sperimentate due differenti matrici di costo, al fine di di individuare quale si adattava meglio ai dati disponibili. Il Cost Sensitive Learning ha portato a risultati buoni per i classificatori **48**, **Logistic**, **Random Forest** e **NBTree**. Su tali classificatori sono stati registrati i risultati migliori in termini di Recall e F_1-measure, confermando la bontà dei modelli **J48** e **NBTree** nel rispondere alla domanda di ricerca che ci si è posti.
Un possibile sviluppo futuro consiste nel cercare dei gruppi all'interno popolazione statunitense per consentire al Policy Maker di individuare quali sono le caratteristiche delle fasce più povere della popolazione, sviluppando di conseguenza strumenti in grado di colmare la diseguaglianza economico-sociale esistente fra i diversi gruppi della popolazione. In tal senso le tecniche di Machine Learning come la Cluster Analysis possono rivelarsi utili ad individuare raggruppamenti di persone con caratteristiche simili. Un altro sviluppo futuro può essere quello di applicare i modelli sviluppati a censimenti successivi al 1994, per verificare se con il passare del tempo, le tecniche implementate in tale progetto sono ancora valide per profilare i cittadini statunitensi in base al loro reddito.

Riferimenti bibliografici

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

[2] Ye Wu and Rick Radewagen. 7 Techniques to Handle Imbalanced Data. https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html.

[3] Alessia Pirolo. Il sistema scolastico americano. http://america24.com/news/il-sistema-scolastico-americano.

[4] Gary M. Weiss, Kate McCarthy, and Bibi Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In *DMIN*, 2007.

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