La disuguaglianza di ricchezza e la differenza sociale: un approccio predittivo

Ermellino Andrea, Kolyszko Matteo, Serino Antonio, Fasani Alessandro, Valente Sofia

Dipartimento di Informatica Sistemistica e Comunicazione, DISCO , Università degli Studi di Milano Bicocca

Piazza dell'Ateneo Nuovo, 1, 20126 Milano MI

Abstract: È possibile prevedere se il reddito annuo di un cittadino risulti essere superiore ai $50mila conoscendo alcune caratteristiche individuali quali età, sesso, titolo di studio, situazione familiare, tipologia di impiego e altre informazioni demografiche? La notevole disuguaglianza di ricchezza e reddito è una grande preoccupazione soprattutto negli Stati Uniti. La probabilità di una diminuzione della povertà è una valida ragione per ridurre il crescente livello mondiale di disuguaglianza economica. Il principio dell'uguaglianza morale universale garantisce lo sviluppo sostenibile e migliora la stabilità economica di una nazione. I governi di diversi paesi hanno fatto del loro meglio per affrontare questo problema e fornire una soluzione ottimale. Questo studio mira a mostrare l'uso delle tecniche di Machine Learning nel fornire una soluzione al problema della parità di reddito. A tale scopo è stato utilizzato un dataset estratto dal database del Census Bureau degli USA. È stata effettuata una classificazione per prevedere se il reddito annuo di una persona negli Stati Uniti rientrasse nella categoria di reddito superiore a $50.000 in base a un determinato insieme di attributi.

Key words: Machine Learning, Predictive Analysis, Meta-Classifiers

Indice

1 Dataset e Preprocessing

Il dataset selezionato per l'analisi contiene dati estratti nel 1994 dal database dell'Ufficio del censimento degli Stati Uniti d'America. Il dataset contiene le informazioni di 32561 cittadini americani relative alla provenienza, all'età, al tipo di istruzione, alla situazione familiare e all'impiego nel mondo del lavoro. In particolare, le 15 colonne che compongono il dataset sono:

- *age (Numeric - Int)*: Età;
- *workclass (String)*: Categoria di impiego (Privato, Statale, Federale etc.);
- *fnlwgt (Numeric - Int)*: Attributo che descrive un cittadino all'interno della popolazione. Il peso viene assegnato tenendo conto dell'etnia, del sesso e dell'età del cittadino. Cittadini con caratteristiche demografiche simili avranno un peso simile;
- *education (String)*: Titolo di studio;
- *education.num (Numeric - Int)*: Codice del titolo di studio;

- *marital.status (String)*: Stato civile;
- *occupation (String)*: Tipo di impiego;
- *relationship (String)*: Stato della relazione familiare;
- *race (String)*: Etnia di appartenenza;
- *sex (String)*: Sesso;
- *capital.gain (Numeric - Int)*: Utile in conto capitale;
- *capital.loss (Numeric - Int)*: Perdita in conto capitale;
- *hours.per.week (Numeric - Int)*: Ore di lavoro settimanali;
- *native.country (String)*: Paese di provenienza;
- *income (String)*: Reddito annuale, attributo binario ($<= 50k, > 50k$).

Gli attributi *education* e *education.num* forniscono la medesima informazione, il primo in formato testuale e il secondo in formato numerico, per questo motivo in fase di preprocessing dei dati l'attributo *education.num* è stato rimosso in quanto informazione ridondante.
Il dataset presenta dei valori mancanti in 3 colonne differenti. I dati mancanti non sono segnalati come tali (NULL) ma sono indicati dalla stringa "?" e sono distribuiti nelle colonne *workclass, occupation* e *native.country*, rispettivamente con il 5.6%, 0.02% e 1.7% del totale dei missing values. Vista la percentuale minima di valori mancanti, si è deciso di rimuovere i record che presentassero anche un solo valore mancante in una qualsiasi delle 3 colonne citate. Alla fine del processo di rimozione dei missing values, i record "utili" sono 30162.

2 Metodologia

L'analisi è basata sull'applicazione di diverse tecniche di classificazione binaria per poter predire il valore della variabile *income*. L'obiettivo è stato predire quando un cittadino riesce ad ottenere un reddito annuale superiore ai $50k, ovvero il valore della variabile *income* che è stata identificata come variabile target, sulla base delle variabili di input.
I modelli di classificazione utilizzati seguono un approccio al problema differente:

- **Modelli di regressione**: basati sulla regressione logistica, il loro vantaggio è rappresentato dalla possibilità di considerare qualsiasi tipo di input, risultando così estremamente flessibili. Dopo aver individuato l'attributo di classe, INCOME, e gli attributi di input, il metodo intende misurare l'effetto di ciascun input sull'output e la risposta sarà identificata sulla base del valore assunto dagli input che la influenzano maggiormente. Per l'apprendimento del modello è stato utilizzato il nodo *Logistic* messo a disposizione dalla libreria Weka;

- **Modelli euristici**: fanno riferimento, in particolar modo, ai Decision Trees ed alle Random Forest, un tipo di modello che deriva dai primi. La classificazione avviene sulla base della classe maggioritaria all'interno del nodo foglia. Questi modelli possono ricevere input di qualunque tipo, per questo sono estremamente flessibili. Tra questi modelli si è concentrata l'attenzione sul classificatore *Best-First Decision Tree*, l'albero di regressione *J48* della libreria Weka, *Random Tree* e *Random Forest*;

- **Modelli probabilistici**: includono le Bayesian Network, un modello di classificazione caratterizzato da un grafo con vincolo di aciclicità che consente di rappresentare la distribuzione di probabilità e la dipendenza tra le variabili. Il vantaggio di questo modello è di poter effettuare inferenza anche con con un dominio composto da molte variabili. Il modello utilizzato è *BayesNet* della libreria di Weka, con l'utilizzo di differenti algoritmi di ricerca : *K2, HillClimb, SearchAlgorithm, TAN*;

- **Modelli di separazione**: la teoria alla base di questi modelli fu introdotta da Vladimir Vapnik nel 1994 e raffinata successivamente nel 1995. I modelli di separazione, invece di stimare le densità di probabilità delle classi per effettuare la classificazione, risolvono direttamente il problema di determinare le superfici decisionali tra le classi (classification boundaries). Gli algoritmi appartenenti a questa classe selezionati sono le Support Vector Machines (SVM) e il Multilayer Perceptron. Per le SVM sono stati utilizzati i nodi forniti dalla libreria Weka *SPegasos* e *SMO*, quest'ultimo con l'utilizzo dei kernel *PolyKernel, Puk*

e RBFKernel. Per lo sviluppo di Multilayer Perceptron sono stati utilizzati i nodi *MultylayerPerceptron* di Weka ed *RProp MLP Learner* nativo di Knime, andando a modificare il numero di strati intermedi.

3 Valutazione delle performance

Per valutare la bontà dei modelli sono stati presi in considerazione i principali indicatori di performance: Accuracy, Precision, Recall, F-measure, AUC. Gli indicatori vengono calcolati a partire dal numero di osservazioni correttamente classificate dal modello, e dal numero di osservazioni che invece vengono classificate in modo errato:

- *TP e TN*: numero di istanze appartenenti alla classe positiva (TP) o alla classe negativa (TN) che vengono correttamente classificate;
- *FP e FN*: numero di istanze appartenenti alla classe positiva (FN) o negativa (FP) che vengono classificate in modo errato.

L'Accuracy di un modello di classificazione indica il numero di predizioni corrette effettuate dal modello e assume valore nell'intervallo [0,1]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Nonostante sia un indicatore estremamente diffuso, poiché in questo caso l'obiettivo dell'analisi è predire quando le osservazioni appartengono ad una classe specifica (valore di *income* >50k), l'Accuracy non è un indicatore che da solo può indicare efficacemente la bontà di un modello. Decisamente più adatti al caso specifico in esame, sono gli indicatori di Precision e Recall di un modello. La Precision indica la frazione di record effettivamente positivi tra tutti quelli predetti come tali:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

La Recall invece, definita anche come True Positive Rate, indica la porzione di record positivi classificati correttamente dal modello:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Entrambi gli indicatori assumono valori nell'intervallo [0,1].

Tuttavia, spesso ci si trova davanti ad un tradeoff tra i due indicatori, un miglioramento della Recall porta ad un peggioramento della Precision e viceversa. Per questo motivo si ricorre all'utilizzo di misure alternative che riassumono i valori di Precision e Recall, come la F-Measure.
La F-Measure è la media armonica tra Recall e Precision e anche questa, come gli indicatori descritti in precedenza, assume valori nell'intervallo [0,1]:

$$F - Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

L'ultimo indicatore numerico di performance preso in considerazione fa riferimento alla curva ROC: una curva che mette in relazione il rate di falsi positivi con il rate dei veri positivi. Tracciata la curva, l'area sottesa al grafico (AUC) risulta essere un indicatore di performance di particolare rilevanza.
Oltre agli indicatori numerici, è stato valutato anche il tempo di apprendimento dei singoli modelli e relazionato con gli score numerici di performance ottenuti, in modo tale da effettuare un'analisi completa della bontà del modello.

4 Analisi e Risultati

Dopo la fase di preprocessing in cui è stata rimossa la colonna *education.num* e sono stati rimossi i record contenenti missing values nelle colonne *workclass, occupation* e *native.country*, è stato adottato l'approccio holdout per applicare i metodi di classificazione selezionati. Il metodo holdout prevede la divisione del dataset in training set (67% dei record) e test set (il 33% rimanente dei record). La partizione del dataset è stata effettuata mediante un campionamento casuale stratificato considerando *income* come variabile target, in modo tale da migliorare la precisione e minimizzando gli errori di campionamento.
Per poter effettuare un confronto veritiero delle performance dei diversi modelli di classificazione, i record appartenenti al training set e al test set sono identici per tutti gli algoritmi.
Sono stati adottati diversi approcci di addestramento dei modelli: il primo passo è stato fatto addestrando i modelli considerando tutte le 13 variabili come variabili di input senza effettuare dunque una selezione delle features, successivamente sono stati addestrati i modelli selezionando preventivamente un gruppo di features con un approccio di tipo filter e, infine, sono

stati addestrati i modelli selezionando un gruppo di features con un approccio di tipo wrapper.
Di seguito sono trattati con maggior dettaglio i diversi approcci di addestramento dei modelli.

4.1 Classificazione senza selezione delle features

Il primo approccio di addestramento dei modelli di classificazione è un addestramento utilizzando tutte le 13 variabili a disposizione come variabili di input per predire il valore della variabile target *income*, in particolare quando il valore di *income* è $> 50k$. Nella Tabella 1 sono riportati i valori degli indicatori di performance dei classificatori addestrati.

Classificatore	Accuracy	Precision	Recall	F-Measure	AUC
SVM	0.843	0.748	0.554	0.637	0.896
Logistic	0.850	0.738	0.614	0.670	0.903
Multilayer Perceptron	0.751	/	0.000	/	0.896
Decision Tree	0.855	0.758	0.614	0.679	0.855
Bayesian Network	0.823	0.610	0.803	0.694	0.907

Tabella 1: No Features Selection

In particolare, la SVM a cui si fa riferimento nella tabella è ottenuta mediante l'utilizzo del nodo *SPegasos*, il Multilayer Perceptron mediante l'utilizzo del nodo *MultilayerPerceptron*, il Decision Tree è un Best-First Decision Tree ottenuto mediante l'utilizzo del nodo *BFTree* e la Bayesian Network è ottenuta mediante l'utilizzo del nodo *BayesNet* con algoritmo di ricerca *K2* (tutti i nodi citati sono forniti dalla libreria Weka). Andando ad analizzare i valori degli indicatori di performance, tutti i classificatori ad eccezione del Multilayer Perceptron hanno un'Accuracy superiore a 0.82. Focalizzando l'attenzione sugli indicatori di Precision e Recall, si riesce a notare che il Multilayer Perceptron si comporta come una *ZeroRule* (il più semplice dei classificatori, classifica ogni osservazione come appartenente alla classe di maggioranza) classificando tutte le osservazioni come appartenenti alla classe $<= 50k$. La Bayesian Network riporta il valore di Precision più basso di tutti mentre il valore di Recall più alto, andando a classificare correttamente l'80% delle osservazioni appartenenti alla classe $> 50k$. Spostando l'attenzione sulla curva ROC, e dunque sull'area sottesa al grafico, il classificatore con l'AUC maggiore è sempre la Bayesian Network, che risulta essere il miglior classificatore tra quelli confrontati se si segue l'approccio di addestramento senza features selection anche in termini di velocità di apprendimento.

4.2 Classificazione con selezione delle features mediante approccio di tipo Filter

Il secondo approccio di addestramento dei modelli prevede una selezione delle features di input mediante approccio di tipo Filter. L'approccio di selezione Filter consiste nella selezione delle features di input prima che il modello venga addestrato. Il Filter può essere uni-variato o multi-variato:

- *Uni-variato*: viene scelta una misura di associazione e viene calcolato il suo valore tra l'attributo di classe e gli attributi candidati alla selezione, successivamente vengono ordinati gli attributi candidati in base al valore della misura di associazione e vengono selezionati i primi *K* attributi che diventeranno dunque gli attributi di input per l'addestramento del classificatore;
- *Multi-variato*: vengono identificati gli attributi irrilevanti e ridondanti con l'obiettivo di ottenere un insieme di attributi fortemente associati all'attributo di classe e incorrelati fra loro.

Per il Filter Uni-variato sono state selezionate le misure di associazione *InfoGain* e *GainRatio*, mentre per il Filter Multi-variato è stato utilizzato il metodo *CfsSubsetEval* che permette di selezionare un subset di attributi in base all'influenza che questi hanno sulla predizione del valore dell'attributo di classe. In entrambi i casi, sia per Filter Uni-variato sia per Filter Multi-variato, sono stati selezionati 5 attributi dei 13 candidati.
Per poter effettuare questo tipo di features selection è stato utilizzato il nodo *AttributeSelectedClassifier* della libreria Weka. All'interno della Tabella 2 sono riportati i valori degli indicatori di performance dei migliori classificatori per ogni tipologia.

Classificatore	Accuracy	Precision	Recall	F-Measure	AUC
SVM	0.831	0.673	0.625	0.648	0.762
Logistic	0.839	0.729	0.563	0.635	0.886
Multilayer Perceptron	0.815	0.668	0.510	0.579	0.865
Decision Tree	0.855	0.781	0.581	0.667	0.898
Bayesian Network	0.789	0.551	0.822	0.659	0.886

Tabella 2: Features selection mediante Filter

I classificatori a cui si fa riferimento nella Tabella 2 sono i seguenti:

- **SVM**: *SMO* di Weka con kernel *PolyKernel* in seguito alla features selection mediante Filter Univariato con misura di associazione *IngoGain* in cui sono stati selezionati gli attributi *relationship, marital.status, capital.gain, age e occupation*;

- **Logistic**: *Logistic* di Weka in seguito alla features selection mediante Filter Multi-variato in cui sono stati selezionati gli attributi *education, marital.status, relationship, capital.gain e capital.loss*;

- **Multilayer Perceptron**: *MultylayerPerceptron* di Weka in seguito alla features selection mediante Filter Uni-variato con misura di associazione *IngoGain* in cui sono stati selezionati gli attributi *relationship, marital.status, capital.gain, age e occupation*;

- **Decision Tree**: *BFTree* di Weka in seguito alla features selection mediante Filter Multi-variato in cui sono stati selezionati gli attributi *education, marital.status, relationship, capital.gain e capital.loss*;

- **Bayesian Network**: *BayesNet* di Weka con algoritmo di ricerca *K2* in seguito alla features selection mediante Filter Uni-variato con misura di associazione *IngoGain* in cui sono stati selezionati gli attributi *relationship, marital.status, capital.gain, age e occupation*.

Confrontando i valori degli indicatori di performance mostrati nella Tabella 2 con quelli mostrati nella Tabella 1, si nota subito che la selezione delle features ha portato ad un miglioramento netto del Multilayer Perceptron. Infatti, il classificatore in questione non performa più come una *ZeroRule* in seguito alla features selection anche se non ottiene buoni risultati in termini di Recall, classificando correttamente solamente il 51% delle osservazioni appartenenti alla classe $> 50k$. Sia la SVM che il modello di regressione logistica peggiorano di poco in termini di Accuracy, Precision e AUC in seguito alla features selection ma la SVM migliora in Recall, andando a classificare correttamente una porzione più ampia di osservazioni della classe d'interesse. Il Decision Tree ottiene la stessa Accuracy anche dopo la selezione degli attributi di input, migliorando in Precision ma peggiorando in Recall e conseguentemente in F-Measure, sintomo di un aumento del numero di osservazioni appartenenti alla classe $< 50k$ che vengono classificate in modo errato. La Bayesian Network è il classificatore che più risente della selezione delle features. L'Accuracy peggiora di molto, così come la Precision ma viene registrato un aumento della Recall, per cui nonostante il peggioramento dei valori degli altri indicatori di performance, la Bayesian Network risulta essere ancora una volta il modello che classifica correttamente il più alto numero di osservazioni appartenenti alla classe d'interesse. Si noti che, sia nel caso di Filter Uni-variato sia nel caso di Filter Multi-variato, gli attributi *relationship, marital.status e capital.gain* vengono sempre selezionati come attributi di input poiché sono gli attributi che più influenzano il valore della variabile target.

4.3 Classificazione con selezione delle features mediante approccio di tipo Wrapper

Oltre al Filter, un altro approccio per selezionare le features e diminuire la dimensionalità del training set è il metodo Wrapper: attraverso l'utilizzo di un classificatore viene trovato un subset di attributi ottimale tra tutti gli attributi candidati al fine di ottimizzare una data misura di performance. In questo caso specifico, tutti i modelli sono stati addestrati ottimizzando prima la F-Measure e poi l'AUC.

4.3.1 Wrapper con massimizzazione della F-Measure

Nella Tabella 3 sono riportati i valori degli indicatori di performance ottenuti dai diversi modelli in seguito alla features selection con metodo Wrapper massimizzando la F-Measure.

Classificatore	Accuracy	Precision	Recall	F-Measure	AUC
SVM	0.849	0.746	0.598	0.664	0.765
Logistic	0.852	0.744	0.616	0.674	0.900
Multilayer Perceptron	0.842	0.796	0.493	0.609	0.883
Decision Tree	0.846	0.748	0.577	0.651	0.863
Bayesian Network	0.845	0.683	0.705	0.694	0.906

Tabella 3: Wrapper con massimizzazione della F-Measure

I classificatori a cui si fa riferimento nella Tabella 3 sono i seguenti:

- **SVM**: *SMO* con kernel *PolyKernel* di Weka, utilizzando il classificatore *SPegasos* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi;

- **Logistic**: *Logistic* di Weka, utilizzando il classificatore *SPegasos* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi;
- **Multilayer Perceptron**: *MultylayerPerceptron* di Weka, utilizzando il classificatore *MultylayerPerceptron* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi;
- **Decision Tree**: *J48* di Weka, utilizzando il classificatore *SPegasos* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi;
- **Bayesian Network**: *BayesNet* con algoritmo di ricerca *K2* di Weka, utilizzando il classificatore *SPegasos* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi.

Tutti i classificatori presentano un'Accuracy superiore a 0.84. La Bayesian Network riporta il più basso valore di Precision ma il più alto valore di Recall e di F-Measure, risultato che indica una tendenza maggiore rispetto agli altri classificatori di classificare le istanze come appartenenti alla classe positiva. Rispetto alla features selection mediante Filter, tutti i classificatori ad eccezione del *J48* presentano un valore di F-Measure maggiore in seguito alla features selection con l'approccio Wrapper, il *J48* invece ha performato meglio con l'approccio Filter. A conferma di questa osservazione, i valori di AUC dei classificatori, escluso il *J48*, sono superiori in seguito alla features selection con Wrapper rispetto a quella effettuata con Filter.

4.3.2 Wrapper con massimizzazione della AUC

Nella Tabella 4 sono riportati i valori degli indicatori di performance ottenuti dai diversi modelli in seguito alla features selection con metodo Wrapper massimizzando la AUC.

Classificatore	Accuracy	Precision	Recall	F-Measure	AUC
SVM	0.843	0.712	0.621	0.663	0.769
Logistic	0.845	0.734	0.591	0.655	0.892
Multilayer Perceptron	0.847	0.796	0.520	0.629	0.895
Decision Tree	0.856	0.775	0.591	0.671	0.861
Bayesian Network	0.829	0.623	0.792	0.697	0.910

Tabella 4: Wrapper con massimizzazione della AUC

I classificatori a cui si fa riferimento nella Tabella 4 sono i seguenti:

- **SVM**: *SMO* con kernel *PolyKernel* di Weka, utilizzando il classificatore *SPegasos* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi;
- **Logistic**: *Logistic* di Weka, utilizzando il classificatore *SPegasos* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi;
- **Multilayer Perceptron**: *MultylayerPerceptron* di Weka, utilizzando il classificatore *MultylayerPerceptron* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi;
- **Decision Tree**: *BFTree* di Weka, utilizzando il classificatore *SPegasos* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi;
- **Bayesian Network**: *BayesNet* con algoritmo di ricerca *K2* di Weka, utilizzando il classificatore *BayesNet* e l'algoritmo di ricerca *GreedyStepwise* per trovare il subset ottimale di attributi.

Confrontando i risultati mostrati nella Tabella 4 con quelli mostrati nella Tabella 2, relativi alle performance dei classificatori in seguito alla features selection con Filter, tutti i classificatori presentano un'Accuracy migliore in seguito alla features selection con Wrapper massimizzando la AUC. Anche i valori di Precision sono migliori in seguito al Wrapper per tutti i classificatori, tranne per il Decision Tree che invece presenta valori peggiori anche in F-Measure e AUC. Un risultato molto simile è stato riscontrato nel confronto effettuato in precedenza nella sezione 5.3.1, per cui il Decision Tree aveva fatto registrare performance migliori con una features selection tramite Filter. La Bayesian Network fa registrare ancora una volta il più basso valore di Precision e il più alto di Recall, anche in questo caso il modello tende eccessivamente a classificare osservazioni come appartenenti alla classe positiva.

4.4 Meta-Classificatori

I meta-classificatori sono una classe particolare di classificatori. L'idea comune a tutti i modelli di questo tipo è quella di creare un classificatore sulla base dell'aggregazione di più classificatori di base. Nel caso specifico, sono stati addestrati tre meta-classificatori differenti:

- **RotationForest**: addestrato utilizzando J48, la RotationForest è un metodo per generare insiemi di classificatori basati sull'estrazione di feature. Per creare i training set per un classificatore di base, il feature set viene suddiviso casualmente in K sottoinsiemi e la Principal Component Analysis viene applicata a ciascun sottoinsieme. Tutti i componenti principali vengono conservati al fine di preservare le informazioni sulla variabilità nei dati, inoltre vengono effettuate k rotazione dell'asse per formare le nuove feature per un classificatore di base;

- **RealAdaBoost**: addestrato utilizzando BayesNet, il RealAdaBoost costruisce una lista di classificatori assegnando, in maniera iterativa, un peso ad ogni nuovo classificatore considerando la sua capacità di riconoscere campioni non correttamente identificati dagli altri classificatori già coinvolti nell'addestramento. Tutti questi classificatori coinvolti voteranno con il peso loro assegnato e la scelta finale avverrà per maggioranza;

- **RandomCommittee**: addestrato utilizzando MultilayerPerceptron come classificatore di base, nella RandomCommitee un numero prestabilito di classificatori viene addestrato utilizzando un seed diverso ogni volta. La media delle previsioni generate dai singoli classificatori di base costituisce la previsione finale. Si tratta di un'aggregazione di classificatori di base randomizzabili.

I risultati ottenuti in termini di performance sono riportati nella Tabella 5.

Classificatore	Accuracy	Precision	Recall	F-Measure	AUC
RotationForest	0.858	0.756	0.636	0.690	0.912
RealAdaBoost	0.831	0.624	0.812	0.705	0.914
RandomCommittee	0.838	0.690	0.632	0.660	0.889

Tabella 5: Meta-classificatori

Nessun Decision Tree aveva ottenuto valori di Accuracy, Recall, F-Measure e AUC migliori di quelli riportati dalla *RotationForest*. Allo stesso modo, nessuna Bayesian Network aveva ottenuto valori migliori di F-Measure e AUC rispetto a quelli registrati da *RealAdaBoost*, così come nessun Multilayer Perceptron aveva fatto registrare valori di Recall e F-Measure migliori di quelli ottenuti da *RandomCommittee*.

Tutti i meta-classificatori hanno reso i classificatori di base più "completi" aggregandoli.

5 Tempo di apprendimento

Il tempo di apprendimento dei modelli addestrati varia da pochi secondi (per la maggior parte dei modelli) a 1-2 ore. La SVM *SPegasos*, il modello di regressione Logistica, i Decision Trees e le Bayesian Network impiegano pochi secondi per completare l'apprendimento senza effettuare features selection, i Multilayer Perceptron impiegano poco meno di 30 secondi per lo stesso task mentre le SVM *SMO* con kernel *PolyKernel, Puk, RBFKernel* completano il processo di apprendimento in $\sim$5 minuti.
Per quanto riguarda la features selection tramite Filter, sia nel caso di Filter Uni-variato che nel caso di Filter Multi-variato, il tempo di apprendimento per tutti i modelli è molto breve, di pochi secondi, a causa della riduzione della dimensionalità del training set.
Nel caso di features selection con Wrapper, il discorso è inverso. Il costo computazionale di questa operazione è molto alto, per questo motivo il tempo di apprendimento risulta essere maggiore in questo caso. Il modello di regressione Logistica e la SVM *SPegasos* sono i più veloci a completare l'apprendimento, impiegando pochi secondi. I restanti modelli, soprattutto per quanto riguarda il Multilayer Perceptron, il tempo di apprendimento del classificatore stimato è di 1-2 ore.
Per quanto riguarda i meta-classificatori invece, il tempo di apprendimento di RotationForest e RealAdaBoost è di pochi secondi, il RandomCommittee invece richiede più di 1 ora di apprendimento a causa della sua natura che rende il costo computazionale dell'operazione estremamente elevato.

6 Conclusioni

L'obiettivo dell'analisi è stato trovare il miglior classificatore per predire il valore della variabile target binaria *income*, con particolare attenzione per la classe positiva, ovvero quando l'*income* è $> 50k$. Per poter ricercare il miglior classificatore sono stati adottati diversi approcci di addestramento dei modelli anche prevedendo una fase di features selection. Sono stati addestrati modelli di regressione Logistica, Support Vector Machine, Multilayer Perceptron, Decision Tree e Bayesian Network, infine sono stati utilizzati anche i meta-classificatori Rotation Forest, RealAdaBoost e Random Committee addestrati rispettivamen-

te su classificatori di base J48, Bayesian Network e Multilayer Perceptron. Per valutare le performance dei classificatori ed effettuare un confronto, sono stati presi in considerazione gli indicatori di Tempo di addestramento, Accuracy, Precision, Recall, F-Measure e AUC, in particolare gli ultimi tre sono stati gli indicatori che più hanno influenzato la scelta del miglior modello.
Il modello che ha fatto registrare le performance migliori è la Bayesian Network, un tradeoff ottimo tra tempo di addestramento e performance, al netto però di una tendenza maggiore rispetto agli altri classificatori nel classificare osservazioni come appartenenti alla classe positiva. Infatti, sia nel caso di apprendimento senza features selection, sia nel caso di features selection con Filter che con Wrapper, la Bayesian Network ha sempre riportato il più basso valore di Precision e il più alto valore di Recall tra tutti i classificatori. Questo classificatore ha spesso riportato il più alto valore di AUC, nella Figura 1 è possibile vedere il valore per ogni approccio di apprendimento utilizzato.

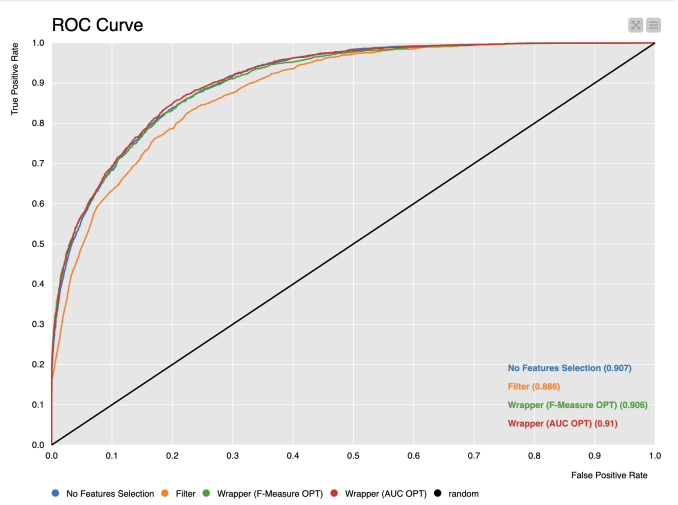

Figura 1: Curva ROC di Bayesian Network con i diversi approcci di apprendimento

Nella Tabella 6 sono riportati invece i valori degli indicatori di performance.

Classificatore	Tipo di apprendimento	Accuracy	Precision	Recall	F-Measure	AUC
Bayesian Network	No Features Selection	0.823	0.610	0.803	0.694	0.907
Bayesian Network	Filter	0.789	0.551	0.822	0.659	0.886
Bayesian Network	Wrapper (F-Measure OPT)	0.845	0.683	0.705	0.694	0.906
Bayesian Network	Wrapper (AUC OPT)	0.829	0.623	0.792	0.697	0.910

Tabella 6: Indicatori di performance di Bayesian Network

I meta-classificatori invece necessitano di un discorso a parte. I risultati ottenuti in un tempo di apprendimento minimo da RotationForest e RealAdaBoost permettono di capire le effettive potenzialità di questa tipologia di algoritmi. L'idea di aggregare più classificatori di base per costituirne uno unico permette di colmare le carenze dei singoli classificatori di base, ottenendo un modello in grado di svolgere operazioni dal costo computazionale elevato in tempi ragionevoli. Il RandomCommittee invece, per sua natura, richiede un tempo di addestramento decisamente più elevato e non confrontabile con gli altri due modelli.
Per perfezionare l'analisi in questione, possibili sviluppi futuri potrebbero portare all'utilizzo di altri meta-classificatori per poter esplorare a pieno le loro potenzialità. In alternativa, sarebbe utile sfruttare più a fondo le potenzialità delle Bayesian Network, modello che è risultato essere il migliore per questo task tra quelli utilizzati.

Riferimenti bibliografici

[1] P. Medici et al. "AdaBoost e le sue varianti" *http://www.ce.unipr.it/ medici/geometry/ node106.html* (2017).

[2] A. Niranjan, D. H. Nutan, A. Nitish, P. Deepa Shenoy, K. R. Venugopal "ERCRTV: Ensemble of Random Committee and Random Tree for Efficient Anomaly Classification using Voting" *3rd International Conference for Convergence in Technology* (2018).

[3] Juan J. Rodrıguez, Ludmila I. Kuncheva Carlos J. Alonso et al. "Rotation Forest: A New Classifier Ensemble Method" *IEEE transaction on pattern analysis anche machine intelligence* (2006).