Alcol, condizioni sociali e performance degli studenti: un'analisi predittiva

Matteo Anedda[1], Diego Bartoli[1], Lorenzo Bruni[1], Niccolò Puccinelli[1]

Abstract

Secondo l'ultimo rapporto Eurostat del 2019 [1], l'8.4% degli Europei consuma alcol almeno una volta al giorno. Una percentuale che sale al 37.4% se consideriamo come intervallo temporale una settimana. In Portogallo circa un quinto della popolazione consuma alcol una volta al giorno, emergendo come la nazione europea con la quota percentuale più alta.
In questo lavoro di ricerca ci chiediamo se e come le condizioni sociali, unite al consumo di alcol, influiscano sulle performance scolastiche.
A tale scopo, abbiamo scelto di utilizzare un dataset della piattaforma Kaggle [2], contenente dati concernenti le condizioni sociali, il consumo di alcol e le performance scolastiche di 649 studenti portoghesi. In particolare, abbiamo applicato e confrontato diversi modelli di classificazione al fine di predire, date le condizioni sociali e il consumo di alcol, se lo studente in questione sia stato bocciato o meno.
Successivamente, provando a scavare più a fondo, abbiamo cercato di capire quali fossero le cause sociali principali del consumo di alcol tra gli studenti. Pertanto, abbiamo effettuato un'analisi comparativa tra diversi classificatori, allo scopo di predire il consumo di alcol sulla base di variabili relative alle condizioni sociali.

[1] CdLM Data Science, Università degli Studi di Milano Bicocca

Keywords
Machine Learning - classificazione - alcol - studenti

Sommario

1. Introduzione

Le performance degli studenti sono influenzate dal consumo di alcol e dalle condizioni sociali in cui versano? Il consumo di alcol stesso è in realtà determinato dalle condizioni sociali? Sono queste le principali domande di ricerca che ci poniamo durante la presente analisi. In particolare, vogliamo valutare, tramite diverse metriche, il grado di affidabilità di differenti modelli di classificazione , nel prevedere (a) se lo studente sia stato bocciato, (b) se l'assunzione di alcol dello studente sia sopra o sotto la mediana.

Riportiamo pertanto i risultati ottenuti e le metodologie impiegate durante l'analisi del problema.

2. Descrizione del dataset

Il dataset, liberamente scaricabile [2], si compone di due file: *student-mat.csv*, composto da 395 record e *student-por.csv*, composto da 649 record. Il primo dataset considera degli studenti di un corso di matematica, il secondo invece riguarda degli studenti di un corso di portoghese. I dataset sono stati tenuti separati al fine di evitare eventuali bias relativi alla materia studiata. Per questa ricerca abbiamo pertanto impiegato il dataset più numeroso, i.e. *student-por.csv*.

2.1 Descrizione degli attributi

Ogni record presenta diverse informazioni relative ad uno studente portoghese, per un totale di 33 attributi. Nello specifico:

- **school** ∈ {'GP', 'MS'}:
  Nome della scuola frequentata.
  - GP: *Gabriel Pereira*
  - MS: *Mousinho da Silveira*
- **sex** ∈ {'F', 'M'}:
  Sesso.
  - F: Femmina
  - M: Maschio
- **age** ∈ [15, 22]:
  Età.
- **address** ∈ {'U', 'R'}:
  Zona di abitazione.
  - U: Urbana
  - R: Rurale
- **famsize** ∈ {'LE3', 'GT3'}:
  Numero di componenti della famiglia.
  - LE3: ≤ 3
  - GT3: > 3
- **Pstatus** ∈ {'T', 'A'}:
  Indica se i genitori vivano insieme o separati.
  - T: Insieme
  - A: Separati
- **Medu** ∈ [0, 4]:
  Educazione della madre.
  - 0: Nessuna educazione
  - 4: Educazione più alta
- **Fedu** ∈ [0, 4]:
  Educazione del padre.
  - 0: Nessuna educazione
  - 4: Educazione più alta
- **Mjob** ∈ {teacher', 'health', 'services', 'at_home', 'other'}:
  Lavoro della madre.
- **Fjob** ∈ {'teacher', 'health', 'services', 'at_home', 'other'}:
  Lavoro del padre.
- **reason** ∈ {'home', 'school reputation', 'course', 'other'}:
  Motivo di scelta della scuola.
- **guardian** ∈ {'mother', 'father', 'other'}:
  Tutore.
- **traveltime** ∈ [1, 4]:
  Tempo impiegato per arrivare a scuola (minuti).
  - 1: < 15
  - 2: [15, 30]
  - 3: (30, 60]
  - 4: > 60
- **studytime** ∈ [1, 4]:
  Tempo di studio (ore).
  - 1: < 2
  - 2: [2, 5]
  - 3: (5, 10]
  - 4: > 10
- **failures** ∈ [1, 4]:
  Numero di bocciature.
  - 0: 0
  - 1: 1
  - 2: 2
  - 3: ≥ 3
- **schoolsup** ∈ {'yes', 'no'}:
  Supporto extra nell'educazione.
- **famsup** ∈ {'yes', 'no'}:
  Supporto familiare nell'educazione.
- **paid** ∈ {'yes', 'no'}:
  Insegnamento extra a pagamento.

- **activities** ∈ {‘yes’, ‘no’}:
  Attività extra-curricolari.
- **nursery** ∈ {‘yes’, ‘no’}:
  Frequentazione della scuola materna.
- **higher** ∈ {‘yes’, ‘no’}:
  Intenzione di proseguire gli studi.
- **romantic** ∈ {‘yes’, ‘no’}:
  Relazione romantica.
- **famrel** ∈ [1, 5]:
  Qualità delle relazioni familiari.
  - 1: molto bassa
  - 5: molto alta
- **freetime** ∈ [1, 5]:
  Tempo libero dopo la scuola.
  - 1: pochissimo
  - 5: tantissimo
- **goout** ∈ [1, 5]:
  Uscite con gli amici.
  - 1: pochissime
  - 5: tantissime
- **Dalc** ∈ [1, 5]:
  Consumo di alcol durante i giorni feriali.
  - 1: pochissimo
  - 5: tantissimo
- **Walc** ∈ [1, 5]:
  Consumo di alcol durante il weekend.
  - 1: pochissimo
  - 5: tantissimo
- **health** ∈ [1, 5]:
  Condizioni di salute attuali.
  - 1: gravi
  - 5: ottime
- **absences** ∈ [0, 93]:
  Numero di assenze.
- **G1** ∈ [0, 20]:
  Valutazione primo periodo.
- **G2** ∈ [0, 20]:
  Valutazione secondo periodo.
- **G3** ∈ [0, 20]:
  Valutazione finale.

2.2 Data exploration

Nel dataset non sono presenti valori nulli e tutti gli attributi numerici sono discreti. Presentiamo di seguito gli istogrammi relativi alle variabili oggetto di analisi predittiva, i.e. **Alc** (**Walc**+**Dalc**) e **G3**.

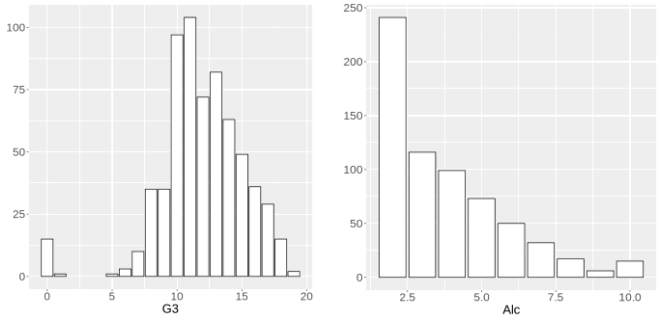

Figura 1: Distribuzione variabili target

Le variabili dipendenti sono state binarizzate in fase di pre-processing.

3. Modelli e metriche utilizzate

Sono state impiegate diverse tecniche di classificazione, allo scopo di individuare la più performante:

- **J48**: Albero di decisione, può essere rappresentato graficamente ed è facilmente interpretabile.
- **Random Forest**: Algoritmo composto da diversi alberi di decisione che riducono la variabilità. Genericamente ottiene performance migliori rispetto agli alberi di decisione, ma perde di interpretabilità rispetto a quest’ultimi.
- **Naive Bayes**: Algoritmo probabilistico, stima la probabilità della classe della variabile dipendente dati gli attributi esplicativi, basandosi sul teorema di Bayes.
- **SMO**: *Support Vector Machines* con *Sequential Minimal Optimization*, costruisce un iperpiano che separa i dati in classi.
- **Simple Logistic Regression**: Regressione logistica, la variabile dipendente è dicotomica ed è in grado di spiegare la relazione che intercorre tra essa e gli attributi esplicativi.
- **Multilayer Perceptron**: Rete neurale, composta da un *input layer* che riceve il segnale e un *output layer* che restituisce la previsione del segnale ricevuto. Tra questi, vi è una serie di *hidden layer*, vero motore computazionale del modello.

Al fine di selezionare il modello migliore, i classificatori sono stati addestrati combinando diversi approcci: Holdout, selezione delle feature, 10-Folds-Cross-Validation e ricerca

dei parametri migliori (tramite il nodo loop *Optimizazion Parameter* [3][4] di Knime).

Misure di Performance

Sono stati utilizzati, per ogni modello, diversi criteri di valutazione delle performance. In particolare:

- **Accuracy**: Percentuale di osservazioni positive e negative previste correttamente. Nel nostro caso di studio, l'accuracy è stata calcolata con un intervallo di confidenza del 95%.
- **Precision**: Un alto valore di precision indica un basso numero di falsi positivi.
- **Recall**: Un alto valore di recall indica un basso numero di falsi negativi.
- **F-Measure**: Media armonica tra precision e recall.

4. Pre-processing dei dati

Vengono inizialmente rimossi gli attributi irrilevanti ai fini delle nostre due domande di ricerca, i.e. non vogliamo che gli attributi che hanno meno a che fare con le condizioni sociali e il consumo di alcol influenzino i nostri modelli. Per di più, l'interpretabilità risulta migliorata.
Attributi rimossi:

- **school**
- **sex**
- **reason**
- **guardian**
- **traveltime**
- **studytime**
- **failures**
- **higher**
- **romantic**
- **absences**

Inoltre, consideriamo come variabile dipendente per la prima domanda di ricerca la valutazione finale (**G3**), escludendo quindi le due valutazioni intermedie (**G1** e **G2**).

Successivamente, introduciamo un nuovo attributo per la prima domanda di ricerca: **<16**, variabile binaria che indica se lo studente consumi alcol senza tuttavia avere l'età per bere. Rimuoviamo dunque **age** per la prima domanda di ricerca: non vogliamo introdurre un bias relativo all'età per la predizione del voto finale. Infatti, uno studente con un'età particolarmente avanzata molto probabilmente è già stato bocciato e potrebbe quindi presentare una maggiore tendenza degli altri ad essere nuovamente bocciato. Per quanto riguarda il consumo di alcol invece, riteniamo che l'età possa essere un fattore interessante da considerare.

Le variabili nominali sono state trasformate in variabili numeriche, assegnando un numero intero ad ogni modalità.

Il dataset è stato infine suddiviso in due partizioni: A (90%) e B (10%) tramite *stratified sampling*, sulla base della variabile target considerata. I classificatori sono stati addestrati sulla partizione A. In seguito, i modelli che sono risultati più efficienti sono stati testati sulla partizione B.

5. Domanda di ricerca #1

La variabile risposta scelta è **G3**. Nello specifico, vogliamo prevedere la bocciatura dello studente. Pertanto, sulla base del sistema scolastico portoghese [5], abbiamo binarizzato **G3**, nel seguente modo:

- **Pass** $\in$ {'yes', 'no'}:
  - 'no': [0,9]
  - 'yes': [10, 20]

La distribuzione tra le due modalità risulta sbilanciata a favore della classe 'yes', che copre circa l' 85% delle osservazioni.

5.1 Sbilanciamento tra classi

Il dataset presenta una distribuzione non equa tra le modalità della variabile risposta, sbilanciamento che può interferire nei risultati prodotti dai classificatori. In generale, l'impiego di modelli di classificazione su dataset sbilanciati comporta un elevato tasso di accuracy, ma questi tendono a concentrarsi sulla classe più frequente, ignorando quella più rara.

Diventa dunque necessario utilizzare misure di valutazione differenti e applicare diverse tecniche per risolvere il problema delle classi sbilanciate.

5.1.1 Holdout

Inizialmente è stato applicato il metodo Holdout. Il dataset è stato perciò suddiviso in training set (70%) e test set (30%),

mediante *stratified sampling*, al fine di mantenere la proporzione tra le due modalità di **Passed**.

I classificatori sono stati quindi addestrati sul training set e testati sul test set.

Classificatore	Recall	Precision	F-measure	Accuracy
J48	0.067	0.133	0.089	0.79
Random Forest	0.133	0.222	0.167	0.795
Naïve Bayes	0.2	0.214	0.207	0.764
SMO	0	0	0	0.846
Logistic Reg	0	0	0	0.846
MLP	0.2	0.286	0.235	0.8

Tabella 1: Holdout

La sola accuracy non è adatta a tale contesto, a causa della natura sbilanciata della variabile target. Per tale motivo, l'analisi di precision, recall e F-measure risulta assai più significativa.

Considerando questi risultati è possibile affermare che i modelli non sono in grado di classificare in maniera corretta. Il livello di accuracy costantemente sopra il 75% non deve trarci in inganno. Infatti, come indicato da precision e recall, l'elevata accuratezza dei modelli è data dal fatto che essi classifichino solo la classe negativa, i.e. quella più rappresentata (**Passed** = 'yes').

5.1.2 Feature selection

Dopo aver applicato il metodo Holdout, si è deciso di effettuare una selezione degli attributi. Nello specifico, abbiamo utilizzato un filtro multivariato, tramite il nodo Weka **AttributeSelectedClassifier** e metodo **cfsSubsetEval**. In questo modo, abbiamo scelto gli attributi maggiormente associati con l'attributo di classe, senza tuttavia essere correlati tra loro [6]. Ciò nonostante, il limite di questa tecnica si manifesta nel fatto che le interazioni coi singoli classificatori vengono completamente ignorate.

Il metodo ha selezionato **Mjob**, **famrel**, **goout**, **Dalc** e **Walc** come variabili ottimali.

I classificatori, a seguito della feature selection, rispondono in maniera pressoché equivalente rispetto all'impiego di tutti gli attributi, peggiorando leggermente nel caso di alcuni modelli. Pertanto, omettiamo la tabella relativa ai risultati ottenuti e proviamo a risolvere il problema dello sbilanciamento tra classi.

5.2 Oversampling

A causa dell'elevato sbilanciamento tra classi e delle ridotte dimensioni del dataset, è stato scelto di effettuare un sovra-ricampionamento della classe meno rappresentata, i.e. **Passed** = 'no'.

Il dataset è stato suddiviso in due partizioni, A (90%) e B (10%), per poi effettuare il ricampionamento sulla partizione A. Quest'ultima è stata poi suddivisa ulteriormente in due parti, contenenti rispettivamente il 70% e il 30% dei dati.

Consci del fatto che le tecniche di *undersampling* e *oversampling* rischiano di far allontanare molto dalla realtà osservata, i modelli finali sono stati testati sulla partizione B, estranea ad ogni processo di ricampionamento.

Per eseguire il sovra-ricampionamento è stato impiegato il metodo SMOTE [7], tramite l'omonimo nodo di Knime [8].

5.2.1 Holdout

E' stato dapprima applicato il metodo Holdout al dataset ricampionato, con le stesse modalità adottate per il dataset originale.

Classificatore	Recall	Precision	F-measure	Accuracy
J48	0.805	0.87	0.836	0.842
Random Forest	0.839	0.899	0.868	0.872
Naïve Bayes	0.638	0.674	0.655	0.663
SMO	0.772	0.714	0.742	0.731
Logistic Reg	0.705	0.739	0.722	0.727
MLP	0.826	0.755	0.788	0.778

Tabella 2: Oversampling - Holdout

I miglioramenti delle prestazioni sono evidenti. In particolare, i modelli **J48**, **Random Forest** e **Multilayer Perceptron** raggiungono valori ottimi (circa 80%), sia per l'accuracy, sia per la precision e la recall. Tuttavia, il miglioramento di quest'ultime due misure non deve trarci in inganno, poiché è dovuto al sovra-ricampionamento del dataset.

5.2.2 Feature selection

Effettuiamo una selezione degli attributi come nel caso precedente, tramite filtro multivariato **cfsSubsetEval**.

Il metodo ha selezionato tutte le variabili come ottimali, eccezion fatta per **health** e **<16**, segno di come le condizioni di salute e l'assunzione di alcol prima dei 16 anni non influenzino significativamente il voto finale.

Come era lecito aspettarsi dalla selezione della quasi totalità delle variabili, i risultati sono equiparabili alla tabella precedente. I modelli risultano leggermente migliorati esclusivamente dal punto di vista dell'interpretabilità, motivo per il quale, nel prosieguo dell'analisi relativa a questa prima domanda di ricerca, sarà comunque considerato il sottoinsieme di variabili di input rilevato dal filtro multivariato.

5.2.3 Ottimizzazione dei parametri

Dato il deludente risultato, in termini di performance, della feature selection, è stata inserita nel *workflow* una fase di ottimizzazione dei parametri di ciascun modello, utilizzando il loop *Optimizazion Parameter* di Knime [3][4]. Nello specifico, il loop effettua una ricerca *brute force* del migliore set di parametri numerici per ogni modello, al fine di massimizzare l'accuratezza sul test set. Partendo da un nodo di start, in cui sono specificati i valori possibili di ogni parametro, ogni modello viene addestrato su tutte le loro possibili combinazioni. Una volta finita la ricerca, il nodo di fine loop restituisce il miglior set di parametri, i.e. quello che massimizza la funzione obiettivo (l'accuratezza, nel nostro caso).

I parametri numerici testati sono i seguenti:

- **J48**:
  - *confidenceFactor*
  - *minNumObj*
  - *numFolds*
- **Random Forest**:
  - *maxDepth*
  - *numFeatures*
  - *numTrees*
- **Simple Logistic Regression**:
  - *maxBoostingIterations*
- **Multilayer Perceptron**:
  - *hiddenLayers*
  - *LearningRate*
  - *momentum*

Per quanto riguarda **Naïve Bayes** e **SMO**, è stato eseguita una ricerca manuale, a causa della natura non numerica dei parametri. In particolare, per **SMO** sono stati testati due tipi di kernel: **poly** e **puk**. Per ulteriori dettagli sui parametri dei modelli, rimandiamo all'Appendice.

I parametri ottenuti con questo procedimento per ciascun modello sono:

- **J48**:
  - *confidenceFactor* = 0.55
  - *minNumObj* = 1
  - *numFolds* = 2
- **Random Forest**:
  - *maxDepth* = 0
  - *numFeatures* = 0
  - *numTrees* = 20
- **Simple Logistic Regression**:
  - *maxBoostingIterations* = 60
- **Naïve Bayes:**
  - *useKernelEstimator* = False
  - *useSupervizedDiscretization* = True
- **SMO**:
  - *Kernel* = puk
- **Multilayer Perceptron**:
  - *hiddenLayers* = 9
  - *LearningRate* = 0.02
  - *Momentum* = 0.7

Una volta inseriti i parametri ottimizzati all'interno dei nostri modelli, abbiamo ottenuto i risultati che presentiamo di seguito.

Holdout

Nella seguente tabella è possibile osservare i valori delle varie metriche ottenuti testando i modelli ottimizzati.

Classificatore	Recall	Precision	F-measure	Accuracy
J48	0.785	0.854	0.818	0.825
Random Forest	0.819	0.968	0.887	0.896
Naïve Bayes	0.758	1	0.863	0.879
SMO	0.819	0.953	0.881	0.889
Logistic Reg	0.691	0.720	0.705	0.710
MLP	0.819	0.824	0.822	0.822

Tabella 3: Oversampling + parameter optimization – Holdout

Da questi risultati è possibile notare il miglioramento delle prestazioni della quasi totalità dei modelli. Una menzione particolare va fatta per il modello **Naïve Bayes**, le cui performance sono migliorate drasticamente.

10-Folds-Cross-Validation

Allo scopo di comparare le performance dei classificatori ottimizzati con un approccio maggiormente esaustivo, questi sono stati sottoposti a **10-Folds-Cross-Validation**. Si è ritenuto opportuno suddividere il training set in dieci parti (uguali), a causa delle dimensioni ridotte del dataset [9].

Confrontiamo, tramite i seguenti box plot, la media relativa all'accuratezza delle dieci iterazioni per ciascun classificatore, con un intervallo di confidenza del 95%.

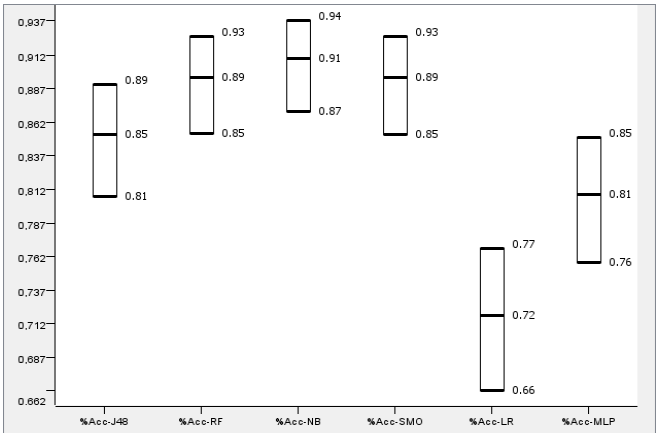

Figura 2: Oversampling + parameter optimization – 10-Folds-CV, box plot

5.3 Analisi dei risultati

Le performance dei modelli addestrati tramite *oversampling* e ottimizzazione dei parametri risultano soddisfacenti. Tuttavia, i classificatori non sono stati testati sui dati originali (i.e. con classi sbilanciate), per cui dobbiamo circoscrivere la validità dei notevoli miglioramenti di precision e recall (e, di conseguenza, F-measure) ad un contesto di distorsione del dataset originale.

Per tale motivo, i modelli sono stati testati anche sulla partizione B, in cui le classi della variabile target **Passed** sono sbilanciate, a favore della modalità 'yes'. L'accuracy ci interessa relativamente, perciò commentiamo solo la tabella concernente le metriche, senza riportare i box plot relativi all'accuratezza.

Classificatore	Recall	Precision	F-measure	Accuracy
J48	0.4	0.308	0.348	0.769
Random Forest	0.2	0.4	0.267	0.831
Naïve Bayes	0	0	0	0.846
SMO	0.1	0.25	0.143	0.815
Logistic Reg	0.4	0.148	0.216	0.554
MLP	0.3	0.188	0.231	0.692

Tabella 4: Test su partizione B

Come era lecito aspettarsi, le performance dei modelli, in termini di precision, recall e F-measure, sono peggiorate notevolmente. Questo a causa del significativo sbilanciamento tra classi della partizione B.

Ciò nonostante, rispetto alla valutazione iniziale (i.e. Holdout senza *oversampling*), diversi classificatori mostrano un significativo miglioramento. A parte il modello **Naïve Bayes** infatti, tutti gli altri indicano una migliore classificazione della classe positiva, seppure non sufficiente per poter affermare di aver raggiunto una buona capacità di predizione.

6. Domanda di ricerca #2

La variabile risposta scelta per la seconda domanda di ricerca è **Alc**, i.e. la somma di **Walc** (assunzione di alcol durante il weekend) e **Dalc** (assunzione di alcol durante i giorni feriali). Nello specifico, vogliamo prevedere il consumo di alcol a partire dalle condizioni sociali. Abbiamo dunque binarizzato la variabile target con l'obiettivo di garantire il miglior bilanciamento possibile tra le due classi, mediante una suddivisione basata sulla mediana della distribuzione.

- **Alc** ∈ {'yes', 'no'}:
  - 'no': [2, 3]
  - 'yes': [4, 10]

La distribuzione tra le due modalità risulta discretamente bilanciata (55% 'no', 45% 'yes').

6.1 Holdout

Per questa seconda domanda di ricerca è stato dapprima impiegato il metodo Holdout, con le stesse modalità di applicazione della prima domanda di ricerca.

Riportiamo i risultati ottenuti nella seguente tabella.

Classificatore	Recall	Precision	F-measure	Accuracy
J48	0.367	0.537	0.436	0.574
Random Forest	0.456	0.581	0.511	0.608
Naïve Bayes	0.532	0.56	0.545	0.602
SMO	0.62	0.636	0.628	0.67
Logistic Reg	0.481	0.655	0.555	0.653
MLP	0.038	0.75	0.072	0.562

Tabella 5: Holdout

I valori riportati in tabella non rilevano differenze particolari tra i diversi classificatori, eccezion fatta per il classificatore **Multi-Layer Perceptron**, il quale presenta una recall molto

bassa, indicando un alto numero di record positivi erroneamente classificati come negativi.

In generale, nonostante le migliori performance rispetto alla prima domanda di ricerca (probabilmente merito anche del bilanciamento tra classi), i modelli non riescono a predire correttamente una significativa percentuale di record. Il modello migliore, **SMO**, comunque non raggiunge il 70% di accuratezza.

6.2 Feature selection

Anche in questo caso è stata effettuata una feature selection, al fine di individuare un sottoinsieme di attributi ottimale. In questo caso, tuttavia, viene utilizzato un filtro univariato, i.e. il metodo **CorrelationAttributeEval** del nodo Weka **AttributeSelectedClassifier**. Questo tipo di filtro seleziona i migliori attributi di input in base ad una determinata misura di associazione (nel nostro caso, la correlazione di Pearson). La scelta di un filtro univariato è dovuta al fatto che il filtro multivariato **cfsSubsetEval**, che tiene conto della multicollinearità tra le variabili, non restituiva un insieme soddisfacente di attributi. Per questo motivo, si è deciso di valutare i singoli legami con l'attributo classe, scegliendo quelli maggiormente correlati. Questa tecnica, tuttavia, presenta una controindicazione in più rispetto al filtro multivariato, poiché ignora le intercorrelazioni tra le variabili [10].

Le variabili più correlate sono **famsize**, **schoolsup**, **famrel**, **freetime**, **goout**, **health** e **age**.

Holdout

In seguito alla feature selection, è stato applicato nuovamente il metodo Holdout.

Classificatore	Recall	Precision	F-measure	Accuracy
J48	0.481	0.613	0.539	0.631
Random Forest	0.506	0.548	0.526	0.591
Naïve Bayes	0.544	0.614	0.577	0.642
SMO	0.544	0.623	0.581	0.648
Logistic Reg	0.506	0.635	0.563	0.648
MLP	0.506	0.69	0.584	0.676

Tabella 6: Feature selection - Holdout

Abbiamo ottenuto una migliore recall per tutti i classificatori ad eccezione di **SMO**. Il miglioramento è particolarmente evidente per il modello **Multilayer Perceptron**. Per quanto riguarda la precision invece, si può notare un miglioramento per i classificatori **J48** e **Naive Bayes**. Nuovamente, la F-measure presenta un valore inferiore soltanto per il classificatore **SMO**. Quest'ultimo ha un valore inferiore anche per l'accuracy, così come **Random Forest** e **Simple Logistic Regression**.

Ad ogni modo, la selezione delle feature ci ha permesso di migliorare notevolmente l'interpretabilità dei modelli.

A differenza del caso precedente, in cui le classi erano molto sbilanciate, l'accuratezza è una misura molto significativa. Per questa ragione, approfondiamo l'analisi dei classificatori sulla base di suddetta metrica e utilizzando, come input, il sottoinsieme di attributi rilevato dal filtro uni-variato.

10-Folds-Cross-Validation

In maniera analoga a quanto fatto per la prima domanda di ricerca (post-*oversampling* del dataset), l'analisi è proseguita con un confronto più completo dell'accuracy dei vari classificatori, mediante **10-Folds-Cross-Validation**.

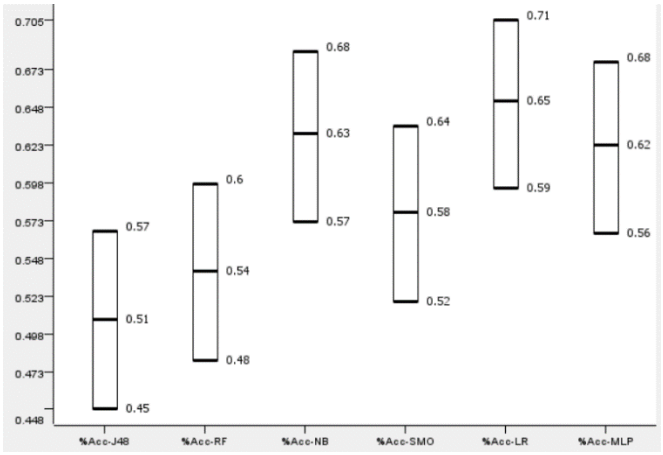

Figura 3: Feature selection – 10-Folds-CV, box plot

Dai boxplot in figura osserviamo che il classificatore **Simple Logistic Regression** presenta la migliore accuratezza media nelle dieci iterazioni, con un intervallo di confidenza del 95%.

6.3 Ottimizzazione dei parametri

Anche per questa domanda di ricerca è stata inserita una fase di ottimizzazione dei parametri, allo scopo di individuare un set di valori che massimizzasse l'accuratezza. I parametri dei vari modelli posti sotto controllo e la procedura impiegata sono analoghi rispetto alla prima domanda di ricerca. Di seguito sono mostrati i risultati ottenuti dal loop.

- **J48**:
  - *confidenceFactor* = 0.55
  - *minNumObj* = 1
  - *numFolds* = 2

- **Random Forest**:
  - *maxDepth* = 0
  - *numFeatures* = 0
  - *numTrees* = 30

- **Naïve Bayes:**
  - *useKernelEstimator* = True
  - *useSupervizedDiscretization* = False

- **SMO**:
  - *Kernel* = puk

- **Simple Logistic Regression**:
  - *maxBoostingIterations* = 10

- **Multilayer Perceptron**:
  - *hiddenLayers* = 6
  - *LearningRate* = 0.07
  - *Momentum* = 0.5

Tuttavia, una volta inseriti i parametri ottimizzati all'interno dei nostri modelli, le performance di quest'ultimi si sono rivelate peggiori. Per questo motivo, abbiamo scelto di tenere come modelli ottimali quelli individuati in precedenza dopo la fase di feature selection.

6.4 Analisi dei risultati

I modelli di classificazione ottenuti in seguito alla feature selection sono stati infine testati sulla partizione B. Analizziamo le performance raggiunte nella tabella sottostante.

Classificatore	Recall	Precision	F-measure	Accuracy
J48	0.75	0.711	0.73	0.692
Random Forest	0.583	0.618	0.6	0.569
Naïve Bayes	0.694	0.658	0.676	0.631
SMO	0.694	0.735	0.714	0.692
Logistic Reg	0.722	0.703	0.712	0.677
MLP	0.694	0.714	0.704	0.677

Tabella 7: Test su partizione B

I risultati ottenuti evidenziano dei valori generalmente più alti di precision e recall rispetto alla precedente validazione (Tabella 6). Nel complesso, vengono rispecchiate le performance ottenute tramite i test sulla partizione A. Tutti i modelli, ad eccezione di **Random Forest**, si attestano intorno ad una percentuale del 70%, sia per quanto riguarda l'accuratezza, sia per quanto concerne la F-measure. Il modello migliore, da questo punto di vista, è l'albero decisionale con algoritmo **J48**.

A differenza della domanda di ricerca precedente, qui abbiamo a che fare con un dataset dalle classi bilanciate. Di conseguenza, riponiamo particolare enfasi sull'accuratezza, rappresentata tramite appositi box plot con un intervallo di confidenza del 95%.

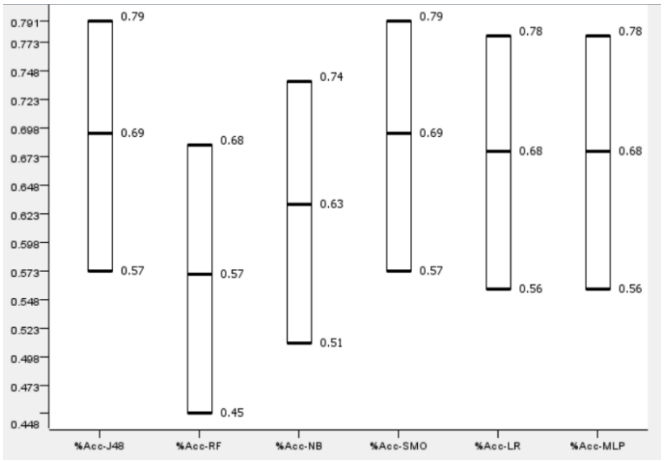

Figura 4: Test su partizione B, box plot

Come già evidenziato dalla tabella, ad eccezione del modello **Random Forest**, l'accuracy dei vari classificatori è molto simile. Tuttavia, l'intervallo di confidenza risulta piuttosto ampio (22% circa), probabilmente a causa delle dimensioni ridotte del set di dati.

In linea di massima, emerge una capacità predittiva variabile, poco sotto al 70%.

7. Conclusioni e sviluppi futuri

In questo progetto di ricerca sono state analizzate e comparate diverse tecniche di classificazione, allo scopo di rispondere alle nostre due domande di ricerca iniziali, i.e.:

- E' possibile prevedere la bocciatura di uno studente sulla base delle condizioni sociali e dell'assunzione di alcol?

- E' possibile prevedere il consumo di alcol stesso sulla base delle condizioni sociali?

Inizialmente, i classificatori sono stati addestrati e testati su tutti i dati, al fine di avere un'idea generale delle prestazioni. Successivamente, il dataset è stato partizionato in due parti, A e B. La partizione A è stata impiegata per l'addestramento e la scelta dei classificatori. Tramite, ad esempio, l'ottimizzazione

dei parametri e la **10-Folds-Cross-Validation**, siamo stati in grado di selezionare i migliori modelli, alla luce delle performance raggiunte sul 30% dei dati della partizione A. I modelli così ottenuti sono stati infine testati sulla partizione B.

Le performance appaiono altalenanti.

Per la prima domanda di ricerca, ci siamo dovuti confrontare col problema dello sbilanciamento tra classi, a causa del quale le prestazioni dei classificatori sono risultate essere insufficienti. Per tale motivo, abbiamo effettuato *oversampling* della classe minoritaria sulla partizione A. Di conseguenza, le prestazioni di tutti i modelli sono migliorate notevolmente (in particolare **Random Forest**, **Naive Bayes** e **SMO**), ma si tratta pur sempre di una distorsione del dataset originario. Infatti, una volta testati sulla partizione B, i classificatori hanno dato prova di una capacità predittiva insufficiente, seppure migliore rispetto al precedente test sui dati non sovra-ricampionati. Il classificatore **J48**, il migliore, raggiunge una F-measure di appena lo 0.348.

Per quanto riguarda la seconda domanda di ricerca, abbiamo binarizzato la variabile target in base al valore mediano, ottenendo una distribuzione delle classi discretamente bilanciata. In virtù di questo, rispetto alla precedente domanda di ricerca, le performance sono da subito risultate migliori in termini di precision e recall, seppure comunque non a livelli soddisfacenti. Anche in questo caso i modelli sono stati dapprima addestrati e testati sulla partizione A (per la quale l'ottimizzazione dei parametri si è rivelata infruttuosa), dopo di ché la partizione B è servita per la validazione finale. Trattandosi di classi bilanciate, ci siamo maggiormente concentrati sull'accuratezza predittiva e, da questo punto di vista, i modelli migliori sono risultati essere **J48** e **SMO**, con quasi il 70% di accuracy.

In sintesi, abbiamo riscontrato un'alta difficoltà predittiva per la prima domanda di ricerca, data dal cospicuo sbilanciamento tra le classi, osservando un significativo miglioramento solo in caso di distorsione dei dati tramite *oversampling* della classe minoritaria. Per quanto riguarda il secondo obiettivo, invece, la capacità predittiva con classi bilanciate si è rivelata migliore, seppure non del tutto soddisfacente.

In generale, questo dataset presenta numerose limitazioni, in primis la mancanza di un numero sufficiente di record (appena 649) su cui poter svolgere task di classificazione con risultati consistenti. Inoltre, vi è una sistematica mancanza di letteratura per quanto riguarda i lavori di analisi predittiva tramite classificazione, al contrario delle EDA (*Exploratory Data Analysis*), che invece sono numerose.

In futuro, potrebbe essere interessante e vantaggioso ampliare le dimensioni di questo dataset, o quanto meno indagare ulteriori aspetti relativi alla vita degli studenti. Dopo di ché, potrebbero essere sviluppati dei nuovi task di classificazione, rinnovando la ricerca dei modelli migliori per la predizione del voto finale e del livello di assunzione alcolica.

8. Riferimenti

[1]
https://www.federvini.it/studi-e-ricerche-cat/3557-eurostat,-italia-terza-in-ue-per-consumo-di-alcol-quotidiano#:~:text=18%20Agosto%202021-,Eurostat%2C%20Italia%20terza%20in%20Ue%20per%20consumo%20di,quotidiano%2C%20ma%20ultima%20per%20abuso&text=Nel%202019%2C%20l'8%2C,consumati%20negli%20ultimi%2012%20mesi.

[2]
https://www.kaggle.com/uciml/student-alcohol-consumption

[3]
https://hub.knime.com/knime/extensions/org.knime.features.optimization/latest/org.knime.optimization.internal.node.parameter.loopstart.LoopStartParOptNodeFactory

[4]
https://hub.knime.com/knime/extensions/org.knime.features.optimization/latest/org.knime.optimization.internal.node.parameter.loopend.LoopEndParOptNodeFactory

[5]
https://www.studyineurope.eu/study-in-portugal/grades

[6]
Hall, M.A., Correlation-based Feature Selection for Discrete and Numeric Class, Machine Learning, Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, USA, 2000, pages 359–366.

[7]
Chawla, Nitesh & Bowyer, Kevin & Hall, Lawrence & Kegelmeyer, W.. (2002). SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (JAIR).

[8]
https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.smote.SmoteNodeFactory

[9]
Raschka, Sebastian. (2018). Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning, pages 20-33.

[10]
S. Visalakshi and V. Radha, "A literature review of feature selection techniques and applications: Review of feature selection in data mining," 2014 IEEE International Conference on Computational Intelligence and Computing Research, 2014, pp. 1-6.

Appendice – breve descrizione dei parametri dei classificatori

J48

confidenceFactor: fattore di confidenza usato per il *pruning* dell'albero.

minNumObj: numero minimo di *instances* per nodo foglia dell'albero.

numFolds: ammontare di dati utilizzato per ridurre l'errore tramite *pruning*.

Random Forest

maxDepth: profondità massima di ciascun albero.

numFeatures: numero di features da prendere in considerazione al momento dello split.

numTrees: numero di alberi che formano la Random Forest.

Naïve Bayes

useKernelEstimator: utilizza una funzione pesata (*kernel*) per stimare gli attributi numerici.

useSupervizedDiscretization: il modello applica una discretizzazione supervisionata, allo scopo di convertire gli attributi numerici in attributi nominali.

SMO

kernel: funzione usata negli algoritmi di tipo SVM. Le tipologie di kernel sfruttate per questa analisi sono puk (*Pearson VII universal kernel*) e poly (*polynomial kernel*).

Simple Logistic Regression

maxBoostingIteration: numero massimo di iterazioni del LogitBoost (*boosting classification algorithm*).

Multilayer Perceptron

hiddenLayers: numero di hidden layers della rete neurale.

LearningRate: determina la dimensione del "passo" di ogni spostamento verso la minimizzazione di una funzione di perdita.

momentum: spesso combinato con il *learning rate*, è usato dalla rete per uscire dai minimi locali.