Wine Type & Quality Detection

Paolo Mariani[1], Dario Carolla[2], Matteo Licciardello[3], Federico Da Ronch[4], Mattia Franzè[5]

Sommario
La storia del vino è strettamente legata alla storia dell'uomo, tracce della sua coltivazione risalgono a oltre 5 millenni fa e rinvenute in più parti del globo: la bevanda viene associata ai termini di festività e convivialità, la sua presenza a tavola rallegra i pasti. La produzione del vino è importante dal punto di vista culturale ed economico soprattutto in Italia, Spagna, Portogallo, Grecia ed Argentina e permette di muovere oltre 300 milioni di ettolitri di merci ogni anno. Prodotto tramite la vinificazione, ossia un processo di trasformazione dell'uva in vino per via della fermentazione, viene spesso affinato con sostanze peculiari per ogni sua varietà, col fine di migliorarne il gusto. Ogni singola bottiglia di vino possiede delle proprietà organolettiche che vengono studiate e giudicate dagli enologi, permettono di determinarne la qualità e di conseguenza il suo valore nel mercato: proprio queste proprietà saranno oggetto di studio per lo scopo fondamentale di classificare il tipo e la qualità del vino attraverso tecniche di machine learning, che permettono di automatizzare il processo di etichettamento e distribuzione del prodotto presso i punti di vendita al pubblico.

Keywords
Machine Learning – Wine Type & Quality Detection Project – Project Report

[1,2,3,4,5] *Dipartimento di Informatica Sistemistica e Comunicazione, CdLM Data Science, Università degli Studi di Milano-Bicocca*
1 - p.mariani20@campus.unimib.it
2 - d.carolla@campus.unimib.it
3 - m.licciardello@campus.unimib.it
4 - m.franze1@campus.unimib.it
5 - f.daronch@campus.unimib.it

Indice

Introduzione

Il dataset selezionato per effettuare l'analisi di machine learning è "Wine Quality" [1], caricato dall'utente Raj Pramar sulla piattaforma Kaggle e originario della repository UCIML, che risulta composto da seimilaquattrocentonovantasette record e dai tredici seguenti attributi:

- Type: tipo di vino che viene considerato (Rosso, Bianco)
- Fixed Acidity: acidi tartarico, citrico, malico e succinico che caratterizzano il gusto del vino, misurati come g/dm^3
- Volatile Acidity: acidi acetico, lattico, formico e butirrico, eliminati durante il processo di produzione
- Citric Acid: acido citrico che fornisce il senso di freschezza al vino
- Residual Sugar: indica lo zucchero presente nell'uva che rimane a seguito della fermentazione
- Chlorides: indica la quantità di cloruro presente nel vino (il sale che principalmente si accumula in base al tipo di acqua che viene utilizzata nel processo di irrigazione)
- Free Sulfur Dioxide: rappresenta l'anidride solforosa libera (residuo non legato in seguito alla reazione con i solfiti)
- Total Sulfur Dioxide: quantità data dalla somma del totale dell'anidride solforosa legata e dell'anidride solforosa libera
- Density: indica la densità espressa in g/cm^3
- PH: misura standard per specificare l'acidità o la basicità del vino
- Sulphates: sali minerali che forniscono sapore e aroma al vino, sono utilizzati per limitare le reazioni con ossigeno e batteri che altererebbero la qualità (spesso una quantità extra viene aggiunta dal produttore)
- Alcohol: indica la concentrazione di alcol presente nel vino
- Quality: rappresenta il voto da 0 a 10 assegnato a ciascun vino (vengono utilizzati esclusivamente valori interi, non ci sono informazioni sulla natura del voto: non è possibile stabilire se questo sia una media arrotondata oppure un voto singolo)

Per una descrizione più accurata di tutte le misure e delle variabili coinvolte nel dataset si consiglia l'approfondimento presso la piattaforma Kaggle [2].
La nostra domanda di ricerca si propone di determinare in base alle proprietà organolettiche del vino:

1. il tipo di vino (precisamente se si tratta di un vino rosso)
2. la qualità del vino

Nel primo caso la variabile dipendente sarà *Type*, nel secondo invece sarà *Quality*.
Per quanto riguarda la classificazione di un tipo di vino si è reso necessario immedesimarsi nei panni di un gruppo di Data Scientist che, su richiesta di un venditore, avrebbero dovuto determinare (in funzione dei dati) il tipo di vino di una serie di esemplari. Nel secondo caso abbiamo supposto che, lo stesso venditore, avesse la necessità di definire un procedimento per la valutazione della qualità del vino, che avrebbe dovuto essere di supporto a sommelier e produttori al fine di stabilire un ipotetico prezzo di vendita (l'approccio è limitato alla classificazione di un vino in base alla sua qualità e non alla determinazione del prezzo).
Determinare il tipo di vino in funzione del tipo di uva che viene utilizzata nel processo di vinificazione è banale, tuttavia non avendo a disposizione informazioni relative al tipo di uva utilizzato ed al suo processo di fermentazione (che avviene in maniera diversa per entrambi i tipi di vini) è necessario effettuare una classificazione basata unicamente sulle proprietà chimiche del vino.
Ugualmente, anche determinare la qualità di un vino risulta complesso: non solo allo scopo di effettuare una previsione ma anche solo considerando che il giudizio in termini di gusto è puramente soggettivo. Però le caratteristiche intrinseche dei vini considerati sono strettamente legate alle proprietà chimiche della bevanda, ed è quindi possibile (in linea teorica) utilizzarle per determinare la bontà di un vino.
Per poter effettuare una classificazione valida è necessario seguire un iter ben preciso in cui si deve, ordinatamente, effettuare una serie di passaggi per ottenere dei risultati soddisfacenti: nella prima fase si effettua l'attività di *Preprocessing* al fine di ottenere dei dati puliti ed adeguati per ogni domanda di ricerca, successivamente per ogni modello che è stato considerato sono state messe in luce le misure di *performance* relative al funzionamento sul dataset. In ultimo sono stati scelti i modelli maggiormente significativi per le nostre attività e sono state applicate delle tecniche volte a migliorare la qualità della classificazione, tra le quali *Cross Validation* e *Feature Selection*.
A seguito della procedura, che verrà svolta per entrambi gli obiettivi, saranno effettuate considerazioni oggettive sul funzionamento dei modelli ed i risultati ottenuti.

1. Preprocessing

La fase di *preprocessing* dei dati è unica per entrambi gli obiettivi prefissati: essa è volta alla pulizia del dataset, operazione che risulta fondamentale per garantire dei buoni risultati degli algoritmi; in primo luogo è stato deciso di utilizzare tutte le variabili, a disposizione (senza escluderne a priori), in quanto ognuna rappresenta un aspetto importante per ogni vino considerato.

1.1 Analisi dei Dati e Statistiche

Il dataset utilizzato è stato inizialmente analizzato tramite il software KNIME. Tramite un semplice nodo "Statistics" è stato possibile osservare il numero di vini bianchi e rossi presenti nel dataset.

red	white
1599	4898

Inoltre si è osservato che all'interno del dataset sono presenti 34 righe con almeno un missing data. Per quanto riguarda la qualità del vino si è voluto osservare visivamente la distribuzione dei vini tramite il grafico seguente [Figura 1].

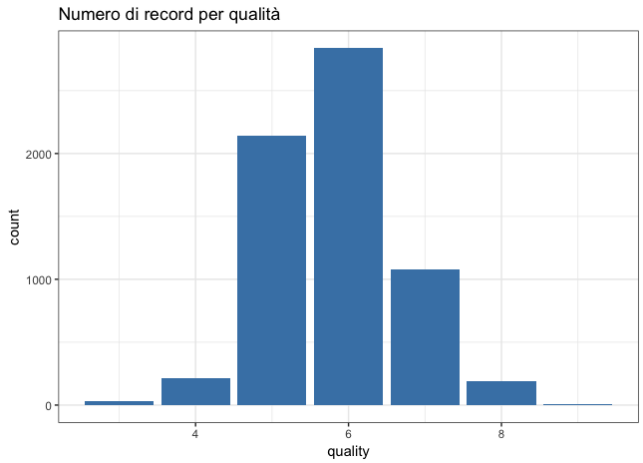

Figura 1. Grafico qualità

Come è possibile visualizzare, il maggior numero di vini sono stati classificati con una qualità pari a sei.

1.2 Missing Replacement

In questa fase è stato risolto il problema dei valori mancanti che si presentavano prevalentemente nelle variabili *Fixed Acidity, Residual Sugar, PH, Volatile Acidity, Chlorides* e *Sulphates*. La tecnica scelta per ovviare al problema è stata quella di eliminare i record corrispondenti: ogni singolo attributo che possiede dei missing values è influente sull'esito della previsione del tipo di vino e della sua qualità, procedere sostituendo il valore mancante con un valore proveniente da record simili sarebbe erroneo in quanto si considera ogni modello utilizzato per la classificazione estremamente sensibile ai cambiamenti dei valori; identicamente anche sostituire il missing value con il valore medio calcolato su tutti i valori assunti dalla variabile considerata risulterebbe sbagliato.
Il procedimento di eliminazione delle righe modifica il dataset riducendolo da 6497 osservazioni a 6463, eliminando 34 record.

1.3 Aggregazione

All'interno del dataset originale è possibile osservare come molteplici record si ripetano con gli stessi valori per tutte le 13 colonne: volendo considerare solo valori unici per ogni osservazione, è stato deciso di eliminare i duplicati procedendo a raggrupparli in record unici tramite il nodo di KNIME "Group by", di conseguenza c'è stata una diminuzione ulteriore delle osservazioni presenti nel dataset da 6463 a 5295.

1.4 Binning dell'attributo Quality

L'attributo "Quality" può assumere dieci valori numerici, in seguito all'eliminazione dei duplicati, si decide di effettuare un *binning* dei suoi valori:

- Low: valori da -inf a 5 (compreso)
- Medium: valori pari a 6
- High: valori da 6 (escluso) a +inf

È stata scelta questa tipologia di suddivisione perché i vini sono stati classificati secondo numeri naturali con evidente *bias* verso il valore "6", di conseguenza la variabile *Quality* è stata trasformata passando da un tipo Integer ad un tipo String, rendendola compatibile con il tipo di variabile richiesto da molteplici modelli di classificatori.

1.5 Normalizzazione delle variabili

Il processo di normalizzazione è stato effettuato per garantire performance superiori per tutti i modelli a disposizione: è stato scelto di normalizzare i valori delle variabili *Residual Sugar, Free Sulfur Dioxide* e *Total Sulfur Dioxide* tramite la tecnica di Normalizzazione Gaussiana, in quanto assumevano valori troppo sparsi

nel loro dominio e, di conseguenza, si sarebbe corso il rischio di ”saturare” i neuroni delle reti neurali. I rimanenti attributi non necessitavano di alcuna modifica, a riprova di questo l'applicare la normalizzazione ad essi non ha portato ad alcun miglioramento di performance.

2. Wine Type - Modelli, Misure di Performance

2.1 Modelli

Per svolgere la prima domanda di ricerca, riguardante la classificazione del tipo di vino, è stato scelto di utilizzare tutte le variabili coinvolte nel dataset poichè si considerano tutti gli attributi fondamentali per riconoscere le caratteristiche che rendono un vino unico, ad esclusione della variabile *Quality* che a parer nostro non risulta utile per determinare il tipo di vino; non avendo effettuato alcuna operazione complessa di *Feature Selection*, la prima operazione effettuata per definire un algoritmo di classificazione è stata confrontare i principali modelli a disposizione dividendoli in tre categorie in funzione della tecnica di classificazione impiegata:

- Separativi (SVM e Reti Neurali)
- Probabilistici (Bayesiani)
- Euristici (Decision Tree Learner, J48 e Random Forest)

2.2 Misure di Performance

Il nostro interesse è stato quello di determinare la tipologia di vino, in particolare se questo fosse di tipo ”Red”: la classe considerata risulta però sbilanciata rispetto alla classe ”White” in quanto possiedono relativamente 1599 e 4898 osservazioni ciascuna: si evidenzia una forte disparità a svantaggio della classe ”Red” che in proporzione è circa un terzo dell'altra.
Considerando la classe rara come classe positiva è possibile tracciare la seguente matrice di confusione:

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

Gli elementi che risultano fondamentali per la valutazione delle performance sono le seguenti quantità:

- Con **TN** indichiamo i True Negative, cioè il numero di osservazioni che il modello ha classificato correttamente come non-rossi (i vini che ha correttamente classificato come ”White”);
- Con **TP** indichiamo invece i True Positive, cioè il numero di osservazioni che il modello ha correttamente classificato come ”Red”;
- Con **FP** indichiamo False Positive, cioè quella porzione di osservazioni che il modello erroneamente considera ”Red” quando in realtà sono di tipo ”White”;
- Con **FN** indichiamo invece i False Negative, cioè quelle osservazioni che il modello erroneamente classifica come ”White” quando in realtà risultano ”Red”.

Definite queste quantità è possibile determinare il valore delle quattro seguenti grandezze derivate:

- *Accuracy*: $\frac{TP+TN}{TP+FP+TN+FN}$
  Rappresenta la percentuale delle osservazioni positive e negative che sono correttamente predette;
- *Precision*: $\frac{TP}{TP+FP}$
  Sono le osservazioni effettivamente positive su tutte le osservazioni che il modello prevede come positive;
- *Recall*: $\frac{TP}{TP+FN}$
  Rappresenta tutte le osservazioni positive che vengono correttamente predette;
- *F-Measure*: $\frac{2*R*P}{R+P}$
  Media armonica tra Recall e Precision;

3. Wine Type - Holdout, Feature Selection, Cross Validation & Validation

3.1 Holdout

L'attività di classificazione del tipo di vino (*implementata nel workflow nella componente ”PREVISIONE DELL'ATTRIBUTO TYPE”*) necessita fondamentalmente della ripartizione del dataset, definita ***Holdout***, in due importanti componenti: *Training Set* e *Test Set*.
Il *Training Set* rappresenta la porzione di dataset che viene utilizzata per allenare il modello a riconoscere le osservazioni, in modo tale che l'algoritmo possa apprendere gli elementi discriminanti che permettono di distinguere i record appartenenti a classi diverse.
Il *Test Set* è, invece, la porzione di dataset che viene utilizzata per verificare lo stato di apprendimento del

modello, ovvero le osservazioni a cui il modello non ha accesso durante la fase di apprendimento e su cui deve effettuare una previsione.
L'approccio comunemente utilizzato prevede di separare le due componenti assegnando al Training Set 2/3 delle osservazioni e al Test Set il rimanente 1/3; la soluzione appena descritta rappresenta una consuetudine ma non una regola: può infatti essere conveniente cambiare le percentuali in gioco per questioni legate alla dimensione del dataset o alla performance del modello di classificazione. Questa tecnica è stata utilizzata per allenare i modelli e valutarne le performance tramite misure appropriate sia nel caso della classificazione del tipo di vino che per la sua qualità. L'applicazione della tecnica di Holdout può essere meglio compresa attraverso l'osservazione della componente "CONFRONTO DEI CLASSIFICATORI" del workflow colorata in verde chiaro.
Sono stati conseguentemente selezionati nove modelli che sfruttano diverse tecniche di classificazione ed ognuno di essi è stato coinvolto allo scopo di eseguire una comparativa sui risultati di accuracy ottenuti nella fase di holdout:

Accuracy	
Modello	**%**
MLP	0.994
SMO	0.989
Spegasos	0.989
NaiveBayes	0.976
NBTree	0.989
NaiveBayes def	0.973
BayesNet	0.985
Random Forest	0.989
Decision Tree	0.98
J48	0.982

Tabella 1. Nodo "0:42 - Interactive Table (Accuracy)"

Alla luce di questi risultati non si è in grado di scegliere univocamente un classificatore ideale, pertanto si è scelto di selezionare quei classificatori che, a parità di approccio, abbiano ottenuto i migliori valori di *Accuracy* rispetto ai concorrenti: MLP, NBTree, BayesNet, RandomForest e SMO.
Considerando il modello BayesNet come esempio, si possono osservare le seguenti performance a seguito dell'applicazione della tecnica di holdout:

- **Accuracy: 0.985**
- **Recall: 0.978**
- **Precision: 0.981**
- **F-Measure: 0.98**

Pur non avendole riportate per evitare di appesantire la struttura del report, tali misure sono state calcolate per ogni classificatore (consultabili presso i nodi KNIME "0:41", "0:44", "0:46").

3.2 Analisi delle Curve ROC

Una porzione importante del confronto tra modelli, che consente di visualizzare come un modello apprenda progressivamente i dati elaborati, è l'analisi delle curve ROC. Una curva ROC è ottenuta mettendo in relazione True Positive Rate e False Positive Rate man mano che i dati vengono processati dal classificatore.
I risultati che sono ottenuti non verranno mostrati graficamente in quanto poco distinguibili tra loro, ma è possibile verificare che tutti i modelli presi in esame si comportano in modo eccellente fornendo questi valori:

AUC	
Modello	**%**
MLP	0.996
SMO	0.986
NBTree	0.986
BayesNet	0.997
Random Forest	0.996

Tabella 2. Nodo "0:60 - Interactive Table (Area Under Curve)"

3.3 Feature Selection

Un'importantissima componente del'iter che è stato seguito per l'implementazione di un classificatore è la fase di ***Feature Selection***, ossia un procedimento di selezione delle feature (variabili o attributi) che risultano più utili per poter prevedere il valore corretto di una variabile dipendente, in modo da evitare di utilizzare troppe o troppe poche variabili che peggiorano le performance del modello: si sceglie di ottenere il minor numero di feature utili possibili che permettono di ottenere la miglior valutazione complessiva.
Attraverso questa procedura è possibile ottenere un miglioramento nei tempi di esecuzione degli algoritmi, poichè con meno dati su cui effettuare l'attività di training si ottiene una minore attività computazionale che deve essere svolta anche per classificare le osservazioni, inoltre migliora la accuracy del modello con il giusto

sottoinsieme di feature selezionato, riduce la complessità di interpretazione del modello ed evita il fenomeno dell'overfitting.
Il metodo utilizzato per effettuare un procedimento di Feature Selection è la *Forward Feature Selection*: partendo da un'unica variabile si procede ad aggiungere una feature per ogni iterazione, col fine di valutare le performance del modello con ogni insieme di variabili considerate; una volta conclusa la procedura è possibile selezionare quelle variabili che garantiscono le performance migliori. È anche possibile utilizzare la tecnica di *Backward Feature Selection* ma i risultati ottenuti risultano molto simili quindi, a parità di Accuracy, è stata scelta la combinazione più leggera in termini di risorse. Per ogni modello gli attributi che sono stati selezionati risultano consultabili presso il relativo nodo KNIME "Feature Selection Filter".
L'applicazione della Feature Selection può essere osservata nel riquadro "FEATURE SELECTION / FILTER (su classificatori selezionati)" di colore verde scuro posizionato a sinistra e contenuto nel workflow knime.

3.4 Cross Validation

L'attività di ***Cross Validation*** consiste nella separazione del dataset in *n* diverse porzioni di uguale dimensione al fine di utilizzare, per *n* diverse iterazioni per ciascun modello, *n-1* porzioni come Training Set e una partizione *n-esima* come Test Set; tramite questo procedimento è possibile scambiare la partizione utilizzata come Test Set ad ogni interazione con una non ancora utilizzata facente parte del Training Set, valutando il modello con delle performance calcolate sulle computazioni di modelli basate sulle partizioni diverse per ogni iterazione; le performance complessive del modello sui dati vengono calcolate attraverso la media aritmetica dei risultati ottenuti per ciascuna iterazione.
La complessità computazionale di ogni modello viene moltiplicata per il numero corrispondente di iterazioni che vengono effettuate, di conseguenza, si tratta di una pratica sconsigliata per i modelli molto onerosi; tuttavia la domanda di ricerca considerata non fa emergere problematiche in tal senso, in quanto la dimensione del dataset e la complessità degli attributi risulta limitata. Il procedimento, *svolto nel riquardo "FEATURE SELECTION / FILTER (su classificatori selezionati)" posizionato nella parte sinistra inferiore del workflow*, ha permesso di ottenere i seguenti risultati:

Accuracy	
Modello	**%**
MLP	0.99
SMO	0.988
NBTree	0.98
BayesNet	0.988
Random Forest	0.99

Tabella 3. Nodo "0:193 - Box Plot (Confronto Accuracy)"

Come è possibile evincere dai dati contenuti in [Tabella 3], la misura di Accuracy risulta pressoché analoga. Attraverso un test statistico [Figura 2] si verifica tale osservazione:

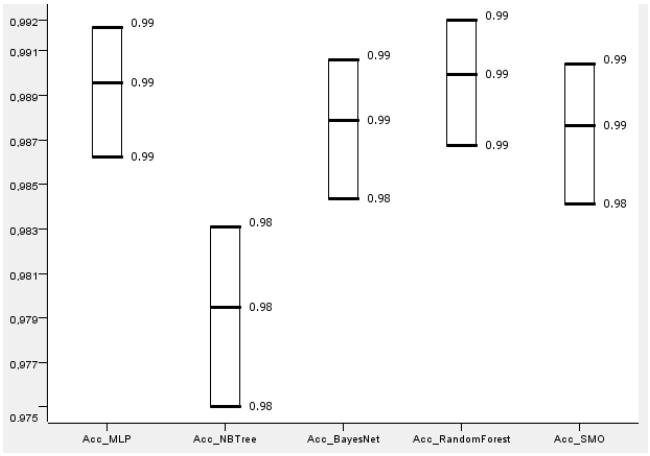

Figura 2. Test Accuracy, nodo "0:193 - Box Plot (Confronto Accuracy)"

Il grafico [Figura 2] mostra che l'unica differenza statisticamente significativa è relativa all'accuracy del modello NBTree che risulta più scarsa, gli altri modelli non presentano differenze statisticamente significative in quanto i boxplot si sovrappongono. Per questo motivo si decide di procedere confrontando i valori di F-Measure.

F-Measure	
Modello	**%**
MLP	0.98
SMO	0.976
NBTree	0.96
BayesNet	0.976
Random Forest	0.981

Tabella 4. Nodo "0:192 - Box Plot (Confronto F-Measure)"

Come è possibile evincere dai dati ottenuti [Tabella 4], anche la misura di F-Measure risulta simile per ogni modello. Si verifica attraverso un test statistico [Figura 3] la validità della nostra affermazione:

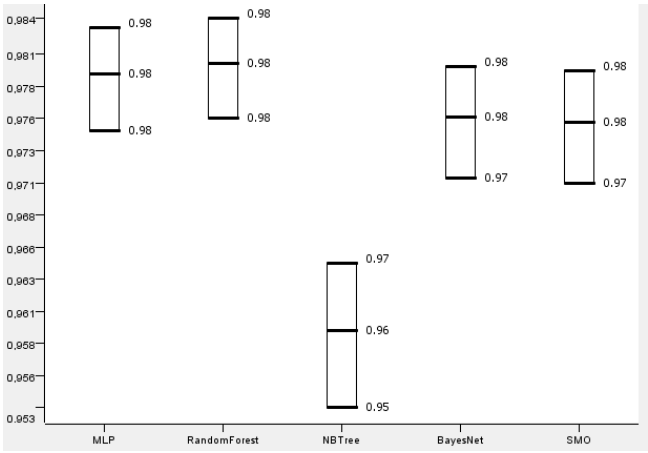

Figura 3. Test F-Measure, nodo ”0:192 - Box Plot (Confronto F-Measure)”

I seguenti risultati permettono di affermare che, sia per i valori di Accuracy che per i valori di F-Measure, i modelli non presentano differenze statisticamente significative ad eccezione del modello NBTree che risulta sensibilmente il peggiore in entrambe le situazioni. Al contrario, uno qualsiasi degli altri modelli presi in considerazione risulta ugualmente valido. Per poter verificare la validità del lavoro svolto, si decide di applicare anche la tecnica di Validation.

3.5 Validation

La procedura di ***Validation*** consiste nel separare il dataset in due porzioni, nel nostro caso saranno 90% e 10%, per poter valutare il funzionamento dei modelli di classificazione su dati che non ha mai considerato in precedenza e sono stati volutamente esclusi dal precedente procedimento di valutazione dei modelli; possiamo considerare la porzione più piccola (10%) un ”nuovo dataset” indipendente dai record che sono stati utilizzati fino al momento considerato (90% delle osservazioni) per allenare i modelli, selezionare le feature interessanti e per effettuare la cross validation; il fine di questa attività è determinare se l’algoritmo di classificazione si sia specializzato erroneamente sui dati a disposizione e non sia in grado di classificare dati mai considerati, ed inoltre valutare la reale performance dei modelli.
La Validation che è stata svolta separa il dataset nelle due porzioni precedentemente stabilite appena dopo l’attività di Preprocessing dei dati, poi -tenendo conto dei risultati ottenuti per Cross validation (con cui sono stati scelti i modelli in base alle valutazioni) e Feature Selection (con cui si aggiornano le variabili utili per ogni modello considerato)- effettua una valutazione dei modelli scelti sulla porzione di dataset mai utilizzata; a seguito dell’osservazione dei risultati è stato deciso di considerare come validi per effettuare un’attività di classificazione del tipo di vino i seguenti modelli:

Accuracy	
Modello	**%**
MLP	0.991
SMO	0.992
NBTree	0.981
BayesNet	0.989
Random Forest	0.992

Tabella 5. Nodo ”0:121 - Box Plot (Accuracy)”

L’analisi dei risultati mostra che i modelli possiedono dei valori di Accuracy prossimi a quelli ottenuti nella fase di Cross Validation durante la quale non sono emerse differenze significative. Si decide di sfruttare i valori di F-Measure per stabilire il modello di riferimento:

F-Measure	
Modello	**%**
MLP	0.981
SMO	0.985
NBTree	0.963
BayesNet	0.977
Random Forest	0.985

Tabella 6. Nodo ”0:115 - Interactive Table (Misure Modelli)”

Alla luce dei risultati ottenuti la selezione di uno qualsiasi tra questi modelli (ad eccezione di NBTree che risulta il peggiore anche in quest’ultimo test) non può essere ritenuta errata.

4. Wine Quality - Holdout, Feature Selection, Cross Validation & Validation

L’implementazione della seconda domanda di ricerca per la classificazione della qualità del vino (attributo ”Quality”) ha previsto sin dall’inizio l’utilizzo dei medesimi modelli sfruttati per prevedere l’attributo ”Type”. Anche le misure utilizzate per valutare la bontà del modello sono le stesse. Per ottenere una valutazione delle performance dei classificatori scelti, si è deciso di applicare lo

stesso iter di valutazione dei modelli con l'applicazione delle fasi di Holdout, Feature Selection, Cross Validation e Validation, che può essere analizzato nel riquadro bianco situato a destra del workflow intitolato "PREVISIONE DELL'ATTRIBUTO QUALITY" (*ogni tecnica implementata per la classificazione dell'attributo "Type" ha la sua corrispondente in questa sezione per classificare l'attributo "Quality", perciò ogni volta che verrà fatto riferimento al workflow, saranno considerate solo le componenti dei riquadri situati sul lato destro*).
Giunti a questa fase dell'analisi si è deciso di utilizzare l'algoritmo per poter classificare correttamente dei vini che saranno commercializzati presso supermercati ed altre attività di grande distribuzione (rivolgendosi quindi ad un pubblico che non risulta essere troppo pretenzioso nei confronti della qualità del vino). Si stabilisce di considerare il modello che meglio classifica i vini di qualità "Medium", quella ideale per un mercato non di nicchia.

4.1 Holdout

L'applicazione del metodo di Holdout sull'intero gruppo di modelli, *nel workflow knime osservabile nel riquadro verde chiaro*, permette di ottenere i seguenti risultati di Accuracy:

Accuracy	
Modello	**%**
MLP	0.599
SMO	0.582
NaiveBayes	0.484
NBTree	0.578
NaiveBayes def	0.533
BayesNet	0.526
Random Forest	0.580
Decision Tree	0.516
J48	0.523

Tabella 7. Nodo "0:144 - Interactive Table (Accuracy)"

È stato selezionato, per ogni approccio, il modello che risultasse il migliore in termini di accuracy. L'insieme risultante è composto da: MLP, SMO, NBTree, BayesNet, RandomForest.

4.2 Analisi delle Curve ROC

Contrariamente a quanto avvenuto per l'attributo Type, per l'attributo Quality è possibile osservare il comportamento dei modelli attraverso la rappresentazione grafica delle curve ROC [Figura 4]:

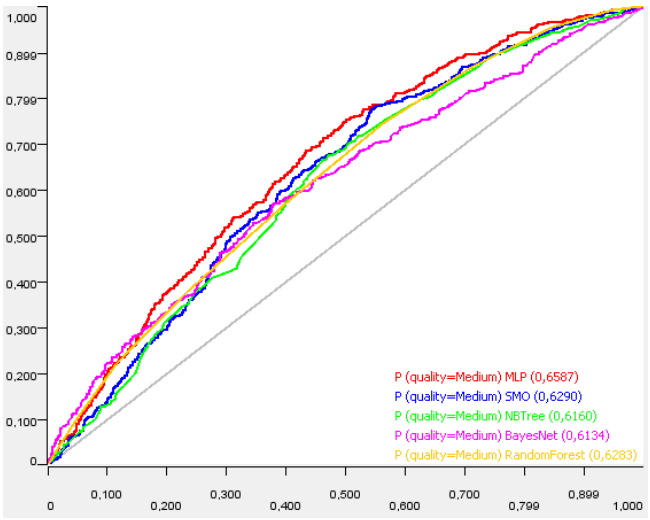

Figura 4. ROC Curve, nodo "0:158 - ROC Curve (MEDIUM)"

La porzione di workflow dedicato all'analisi delle curve ROC è di color salmone, qui vi possono essere trovati i risultati dei valori legati alle curve ROC per i classificatori selezionati nello step precedente

4.3 Feature Selection

Per ogni modello è stato scelto di applicare una tecnica di Forward Feature Selection, che ha portato alla selezione di diversi attributi per ognuno di essi.
Questi possono essere consultati al nodo "Feature Selection Filter" del riquadro verde scuro "FEATURE SELECTION / FILTER (su classificatori selezionati)" del workflow Knime.

4.4 Cross Validation

Anche per questa classificazione è stata applicata la tecnica di Cross Validation, producendo i seguenti risultati per la misura Accuracy:

Accuracy	
Modello	**%**
MLP	0.561
SMO	0.554
NBTree	0.558
BayesNet	0.558
Random Forest	0.542

Tabella 8. Nodo "0:207 - Box Plot (Accuracy)"

Come possiamo evincere dai dati ottenuti, la misura di Accuracy per i modelli considerati risulta pressochè analoga, sebbene non sia eccellente. Attraverso un test statistico osservabile in figura [5] confrontiamo i modelli.

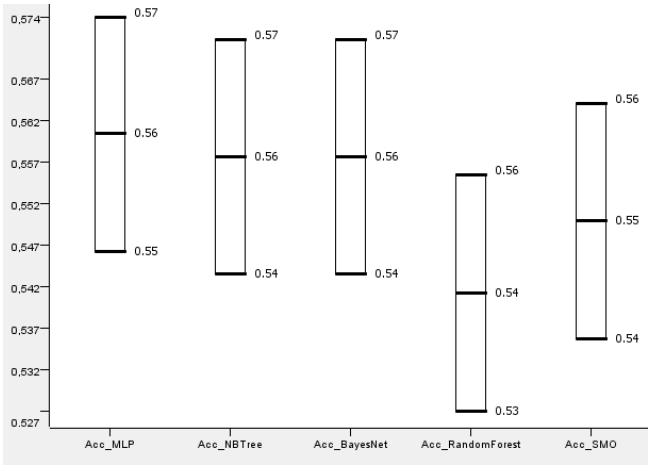

Figura 5. Test Accuracy - Cross Validation, nodo ”0:185 - Box Plot (Accuracy)”

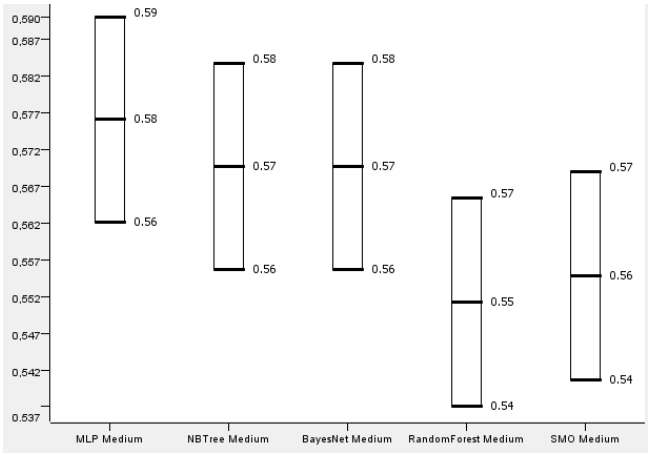

Figura 6. Test F-Measure ”Medium” - Cross Validation, nodo ”0:223 - Box Plot (F-Measure(medium))”

Il grafico mostra che non vi è alcuna differenza statisticamente significativa tra i modelli analizzati in quanto i boxplot si sovrappongono. Per questo motivo si decide di confrontare i valori di F-Measure per la qualità ”Medium”:

F-Measure ”Medium”	
Modello	**%**
MLP	0.576
SMO	0.555
NBTree	0.570
BayesNet	0.570
Random Forest	0.552

Tabella 9. Nodo ”0:223 - Box Plot (F-Measure (medium))”

Come possiamo evincere dai dati ottenuti, anche la misura di F-Measure (Medium) risulta simile per ogni modello. Si verifica attraverso un test statistico [Figura 6] la validità della nostra affermazione:

È possibile stabilire, a seguito del test, che non esiste una differenza statisticamente significativa tra i modelli. In accordo con lo scopo precedentemente descritto, si considerano validi i modelli che possiedono Accuracy e F-Measure più alti: MLP, BayesNet e NBTree.
Per poter verificare la validità del lavoro svolto, si decide di applicare la tecnica di Validation.

4.5 Validation

L’applicazione della tecnica di Validation permette di ottenere i seguenti valori di Accuracy sulla percentuale di dataset (10%) finora non considerata:

Accuracy	
Modello	**%**
MLP	0.557
SMO	0.530
NBTree	0.536
BayesNet	0.536
Random Forest	0.543

Tabella 10. Nodo ”0:216 - Box Plot (Accuracy)”

Attraverso l’analisi dei risultati si evince che i modelli possiedono dei valori di Accuracy prossimi a quelli ottenuti nella fase di Cross Validation.
Attraverso un test statistico [Figura 7] sulla Accuracy dei modelli si nota che non vi è una differenza significativa. Si considerano i modelli selezionati al termine del processo di Cross Validation.

Si stabilisce di considerare i valori di F-Measure ed effettuare su di essi un test statistico [8] per determinare quale dei tre modelli risulta essere il più valido:

F-Measure ”Medium”	
Modello	**%**
MLP	0.566
SMO	0.552
NBTree	0.526
BayesNet	0.526
Random Forest	0.546

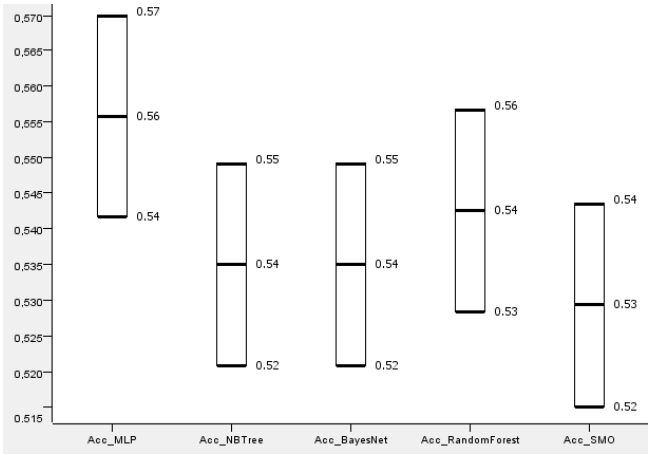

Figura 7. Test Accuracy - Validation, nodo ”0:216 - Box Plot (Accuracy)”

Tabella 11. Nodo ”0:227 - Box Plot (F-Measure (medium))”

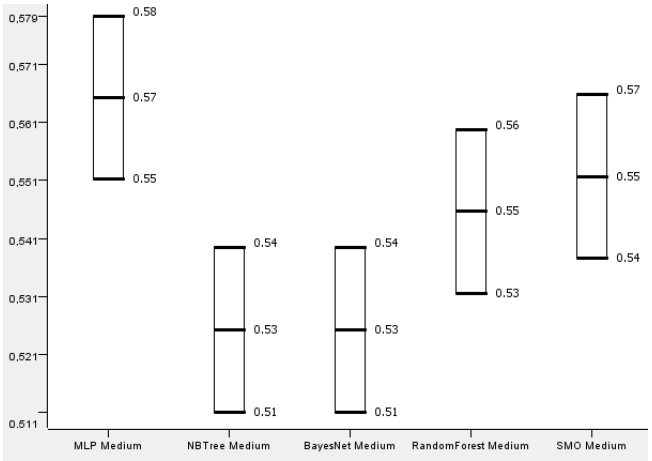

Figura 8. Test F-Measure ”Medium” - Validation, nodo ”0:227 - Box Plot (F-Measure (medium))”

Si può osservare facilmente che una differenza significativa è stata riscontrata, a favore di MLP rispetto agli altri due. A partire da questo test, tenendo in considerazione i risultati ottenuti dal modello per tutte le misure di Accuracy e F-Measure, si decide di utilizzare MLP per effettuare una classificazione della qualità del vino.

Si stabilisce di utilizzare un modello di riferimento per effettuare la classificazione dei vini in funzione della loro qualità: a quanto risulta dall’analisi effettuata sui modelli considerati, si ottengono comportamenti similari per quanto l’accuratezza, ma non per la F-Measure che viene quindi utilizzata per discriminare i modelli NBTree e BayesNet. Il modello MLP risulta essere la scelta più equilibrata, anche nell’ottica di dover utilizzare il modello per previsioni di valori diversi dell’attributo ”Quality”, in quanto risulta essere più flessibile anche rispetto ai classificatori SMO e RandomForest: le prestazioni di questi due modelli nella previsione del valore ”high” [Figura 9] evidenziano come sia SMO che RandomForest siano statisticamente più scarsi per tale compito.

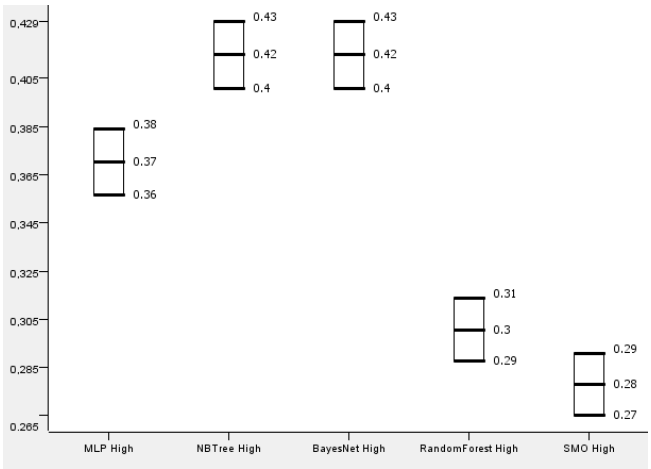

Figura 9. Test Accuracy - Quality ”High”, nodo ”0:224 - Box Plot (F-Measure (high))”

5. Conclusioni

Lo svolgimento di questo studio ha permesso di creare un sistema formato da due componenti atte a riconoscere il tipo di un vino e la sua qualità. L’analisi effettuata per riconoscere la tipologia del vino evidenzia come tutti i modelli considerati abbiano delle prestazioni eccellenti, ciò dovuto al fatto che l’attributo considerato sia binario e gli attributi restanti siano estremamente esplicativi ai fini della sua classificazione.
L’analisi effettuata sulla seconda componente, la quale si occupa di classificare la qualità del vino, mette in luce la maggiore difficoltà nella previsione del valore di un attributo non binario: la maggior parte degli attributi, associati alle osservazioni, non risultano esplicativi. Inoltre, il voto assegnato a ciascun vino può risentire della soggettività nell’assegnazione del voto da parte del critico, ciò potrebbe influenzare la prestazione del sistema; questa problematica non sussiste nel caso della tipologia del vino che risulta essere oggettiva e definita.

Si potrebbe implementare, al fine di migliorare le prestazioni dei modelli, un meccanismo di penalizzazione delle classificazioni errate. Ad esempio, etichettare un vino di qualità bassa come vino di qualità superiore può creare, nei confronti di un cliente, una reazione negativa che può trasformarsi in perdita economica. Classificare, invece, un vino di qualità elevata come un vino di qualità inferiore è un errore che va comunque penalizzato al fine di migliorare l'efficacia della classificazione che però non inficia sulla fiducia del cliente.

Ringraziamenti

Si ringrazia il Professor Stella per aver reso disponibile agli studenti il materiale e gli esempi Knime che sono risultati particolarmente utili per la comprensione del funzionamento della piattaforma; si ringraziano inoltre i creatori del dataset *Wine Quality Dataset*" [3] e del paper "*A probabilistic interpretation of precision, recall and F-score, with implication for evaluation*" [4] che ha permesso di comprendere l'importanza delle misure di performance ai fini dello studio.

Riferimenti bibliografici

[1] UCIML Kaggle, Raj Parmar. Wine quality dataset, uci machine learning repository.

[2] Marcelo Marques. Wines type and quality classification exercises - kernel.

[3] Paulo Cortez, A Cerdeira, F Almeida, T Matos, and J Reis. Wine quality data set. *UC Irvine Machine Learning Repository*, 2009.

[4] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. 2005.