**Features Extraction & Red-Wine Quality Predictions**

Università della Calabria

Dipartimento di Ingegneria Informatica , Modellistica , Elettronica e Sistemistica (DIMES)

Nicola Corea

|  |  |
| --- | --- |
| **Abstract.** Obiettivo del seguente lavoro è quello di fornire una soluzione efficiente al problema della classificazione della qualità dei vini. Molti articoli e lavori forniscono soluzioni singole , o ensemble , garantendo prestazioni al di sotto del 70% su dati mai visti. Tramite opportune preelaborazioni dei dati , vedremo come sia possibile ottenere , soluzioni con prestazioni superiori del 20-30% di quelle esistenti. L’applicazione di tecniche di riduzione della dimensionalità del campione (selezione , estrazione delle caratteristiche) ci consentirà , a scapito di una leggerissima perdita in prestazione , di ottenere modelli più semplici e meno costosi da un punto di vista computazionale. Questo ci consentirà l’addestramento efficiente di ensemble di classificatori. L’articolo si concluderà con l’addestramento di una rete neurale multi-layer (MLP) per scopi di classificazione ed, ovviamente , ad un confronto tra le prestazioni ottenute con quelle di tecniche ensemble.  **I. Intoduzione**  Il seguente articolo punta alla definizione di classificatori con l’obiettivo di predire la qualità del vino sulla base delle sue caratteristiche. Per addestrare i nostri modelli faremo riferimento al dataset reperibile dal repository UCI. Il dataset , grazie al lavoro di P. Cortez , A. Cerdeira , F. Almedia , contiene 1599 varianti del vino portoghese ‘Vinho Verde’ classificate in base a diverse caratteristiche del vino stesso , quali ad esempio: ‘fixed acidity’ , ‘volatile acidity’ , ‘citric acid’ etc… . Sulla base di tali caratteristiche la qualità del vino viene rappresentata da un intero variante sull’intervallo [3,8]. L’intero ‘3’ sarà a riferimento di una bassa qualità del vino , mentre ‘8’ , sarà rappresentativo di una eccellenza. Per quanto riguarda la valutazione delle prestazioni del classificatore , si farà riferimento alla accuratezza del classificatore stesso. Ricordiamo | brevemente , con riferimento alla matrice di confusione riportata in **Figura 1.1** , detto  l’errore nelle predizioni , l’accuratezza (ACC) del classificatore sarà allora data da    **Figura 1.1** Matrice di Confusione  Inizieremo la nostra discussione sul problema proposto , partendo da una piccola fase di preelaborazione dei dati , con l’obiettivo di addestrare in maniera efficiente diversi classificatori (singoli) per la risoluzione (predizione) del problema proposto. Vedremo in accordo a [1] come le prestazioni dei singoli classificatori vanno non oltre il 70%. L’obiettivo sarà allora quello di cercare delle possibili soluzioni con l’intenzione di migliore l’accuratezza del predittore del 20 – 30 %. Fatto ciò con l’intenzione di utilizzare tecniche ensemble , presenteremo e utilizzeremo rispettivamente tecniche di selezione e riduzione delle caratteristiche , per la riduzione della dimensionalità (caratteristiche) cercando di perdere il meno possibile in prestazioni. La trattazione di concluderà con l’addestramento di |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| un ‘Multi – Layer Perceptron’ (MLP) per scopi di classificazione. Vediamo cioè quanto una rete neurale multilivello riesce a ottenere predizioni superiori alle tecniche ensemble.  **II. Analisi Del Problema**  Con lo scopo di valutare le prestazioni del modello ottenuto prima di lasciarlo libero di operare nel mondo reale, suddividiamo il dataset in un 70% per l’addestramento del modello ed il restante 30% per scopi di validazione. Fatto ciò il passo successivo , e di notevole importanza è la riduzione in scala delle caratteristiche. Alcuni algoritmi di apprendimento , utilizzati nel proseguio , quale ad esempio un KNN (K-Nearest-Neighbors) o SVM (Support – Vector -Machine) a differenza di un Albero Decisionale , non è invariante rispetto alla scala delle caratteristiche. Si ricordi a tal proposito che l’individuazione dei k – vicini si traduce nella individuazione dei primi k esempi a distanza minima dal dato da classificare. Se scegliessimo come metrica la norma 2 , o norma Euclidea  appare evidente come le distanze saranno maggiormente influenzate dalle caratteristiche più elevate. Nella **Tabella 1.1** si riporta la dispersione sulla retta reale dei valori assunti da alcune delle caratteristiche. Si nota appunto , come i valori assunti da alcuni parametri quali : ‘Fixed Acidity’ , ‘Residual Sugar’ , sono notevolmente superiori rispetto alle altre. Dalle considerazioni fatte in precedenza risulta evidente la necessità di una prima fase di riduzione in scala delle caratteristiche. In particolare faremo uso della Standardizzazione. Determinata media e varianza per ciascuna caratteristica , le singole realizzazioni verranno trasformate in accordo alla seguente procedura  , intepretabili cioè come realizzazioni di una semplice normale standard. Finita questa prima fase di preelaborazione possiamo passare al compito successivo , quello di valutare le prestazioni di alcuni fra i tanti classificatori. | **Tabella 1.1** Dispersione Caratteristiche  possano essere facilmente estesi a situazioni multi-classe tramite la tecnica OvR. Procederemo con l’addestrare e valutare le prestazioni dei seguenti classificatori  1) KNN (K-Nearest-Neighbors)  2) SVM Kernel , con funzione kernel RBF  3) DecisionTreeClassifier , criterio Entropia  Si ricordi che come criterio di valutazione per i nostri classificatori abbiamo scelto l’accuratezza. Nella tabella qui di sotto si riportano i risultati ottenuti sia in fase di addestramento che in fase di convalida.   |  |  |  | | --- | --- | --- | | Classificatore | Addestramento | Test | | KNN | 0.94 | 0.69 | | SVM | 0.67 | 0.59 | | DTC | 0.66 | 0.58 |   **Tabella 1.2** Risultati Singoli Classificatori  Come possiamo notare dalla tabella risultate , e come confermato in [1] , il miglior classificatore per questo tipo di problema sono di certo le macchine a vettori di supporto (SVM). Possiamo in prima battuta provare a migliorare le prestazioni attraverso tecniche ensemble. Tuttavia non avendo ancora fatto una riduzione della dimensionalità (PCA etc..) , il numero dei classificatori , per via dello elevato numero delle caratteristiche sarà limitato a 25. Si riportano di seguito in formato tabellare i risultati ottenuti con tecniche ensemble , quali: ‘Majority Voting (MV)’ , ‘RandomForest (RF)’ , ‘AdaBoosting (AB)’   |  |  |  | | --- | --- | --- | | Ensemble | Addestramento | Test | | MV | 0.79 | 0.63 | | RF | 0.99 | 0.68 | | AB | 0.56 | 0.54 |   **Tabella 1.3** Ensemble Classificatori  Per quanto riguarda il ‘Majority Voting’ , è stata semplicemente implementata una votazione a |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| maggioranza tra i precedenti singoli classificatori. Come possiamo ben osservare dalla precedente tabella , anche sfruttando tecniche ensemble , non si riescono ad ottenere buone prestazioni; l’applicazione di una RandomForest (Bagging con DecisionTree) di fatto non migliore le prestazioni rispetto al singolo SVM.  Il motivo della scarsa qualità nella fase di generalizzazione ( ma anche in fase di addestramento ) potrebbe di fatto essere dovuta nella non uniforme distribuzione dei campioni per classe. Come infatti riporta la **Figura 1.2** , i dati risultano maggiormente distribuiti sulle classi [5,6]. Ma di fatto una scarsa capacità di predizione , potrebbe essere dovuta alla presenza di rumore nei dati.    **Figura 1.2** Distribuzione Esempi Per Classe  Vedremo nella prossima sezione come migliorare le prestazioni dei nostri classificatori. L’idea alla base della soluzione proposta , con l’obiettivo di massimizzare l’accuratezza , consiste , di fatto , in una seconda fase di preelaborazione dei dati , in cui si vanno a ridefinire le etichette delle classi.  **III. Soluzione Proposta**  Come già discusso nella precedente sezione , l’idea alla base della soluzione proposta , sta in una ridefinizione delle etichette delle classi. In particolare dalle etichette [3,4,5,6,7,8] , si passa alle etichette [0,1,2] corrispondenti a tre diversi livelli di qualità del vino , in particolare , saranno così definite | |  |  | | --- | --- | | Label | Wine Quality | | 0 | Low | | 1 | Medium | | 2 | High |   **Tabella 1.3** Associazione Label Qualità  Rimane , ovviamente , da stabilire come ridefinire l’etichetta della classe per ogni esempio del dataset. La trasformazione della etichetta per ciascun esempio viene fatta in accordo alla strategia presentata nella **Figura 1.3.**  Low Medium High  3 4 5 6 7 8  **Figura 1.3** Ridefinizione Label Classi  Quindi fino a 4 la qualità del vino sarà considerata bassa . Da 5 fino a 7 la qualità sarà considera media , superiore a 7 la qualità del vino sarà giudicata alta. Il perché si è scelta questa ridefinizione delle etichette è semplice . Si è cercato in qualche modo di aumentare il numero di campioni di addestramento per ogni singola classe.  Andiamo quindi a riaddestrare i nostri singoli classificatori in base a questa ridefinizione dei label delle classi , e andiamo infine a valutarne le prestazioni.   |  |  |  | | --- | --- | --- | | Classificatore | Addestramento | Test | | KNN | 0.99 | 0.95 | | SVM | 0.96 | 0.94 | | DTC | 0.95 | 0.95 |   **Tabella 1.4** Risultati Classificatori  Notiamo dunque con questa ridefinizione delle classi del nostro dataset un aumento sulla accuratezza del 20-30%. Possiamo considerarci soddisfatti dei risultati ottenuti. Ancora una volta i classificatori SVM sono quelli che danno i migliori risultati sia in fase di addestramento che sui dati di convalida. |

|  |  |
| --- | --- |
| Ricordiamo che i classificatori precedenti sono stati ottenuti considerando tutte e 11 le caratteristiche del dataset originale. Quello che vogliamo fare a questo punto è cercare di ridurre il numero delle caratteristiche con l’obiettivo di ottenere modelli più semplici , e quindi diminuire il costo computazionale. A tal proposito faremo uso sia di tecniche di selezione che di estrazione delle caratteristiche. |  |