POLITECNICO DI TORINO

III Facoltà di Ingegneria
Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea Magistrale

Visualizzazione e Analisi Interattiva di Dati di Qualità Video

Relatore
prof. Enrico Masala

Candidato
Alessandro Catto

Dicembre 2019

Ringraziamenti

Desidero per prima cosa ringraziare il prof. Enrico Masala per la grande disponibilità riservatami in ogni momento del mio lavoro, nonostante le tempistiche allungate rispetto alle aspettative iniziali di entrambi.

Ringrazio inoltre la mia famiglia, in particolar modo i miei genitori Lorenzo e Daniela, che mi hanno supportato durante il mio intero percorso accademico, sostenendomi nei momenti di sconforto e festeggiando insieme a me i successi ottenuti.

Ringrazio la mia fidanzata Lodovica che mi ha accompagnato fin dal primo anno durante tutto il mio percorso al Politecnico di Torino, ed ha sempre creduto in me indipendentemente dai risultati ottenuti durante gli anni.

Ringrazio inoltre tutti i miei compagni di corso, in particolare quelli con cui ho condiviso i progetti dei corsi che ho frequentato al Politecnico, grazie ai quali ho potuto confrontarmi sul campo con persone che hanno i miei stessi interessi professionali.

Un ultimo ringraziamento lo dedico, infine, a tutti i miei amici, che oltre a sostenermi durante il mio cammino universitario, lo hanno reso anche molto più divertente.

Indice

1 Introduzione

1.1 Motivazione del lavoro svolto

Lo scopo di questo progetto di tesi risiede nella necessità di possedere uno strumento che permetta di analizzare in modo completo ed esaustivo le relazioni presenti tra gli attributi propri di un determinato set di dati, con l'obiettivo di permettere uno studio adeguato, in particolare, di quelle presenti in alcuni dataset contenenti dati di misure di qualità video.

Per fare ciò ci si è domandati se esistesse già un tool in grado di fornire tali funzionalità; parecchi strumenti erano già disponibili ma, come vedremo nel seguito di quest'opera, ciascuno presentava delle limitazioni. Il software che più si avvicinava a quanto desiderato è *Weka,* un software estremamente potente che viene utilizzato per l'elaborazione di grandi quantità di dati mediante l'utilizzo di semplici file simili ai csv[1]*,* permettendone una visualizzazione grafica con un buon livello di dettaglio; tuttavia, utilizzando quest'ultimo per analizzare alcuni dataset, sono emerse alcune debolezze del software, a cui si è cercato di porre rimedio nel corso di quest'opera. Possiamo raggruppare questi aspetti negativi in due macrocategorie:

- La prima riguarda la visualizzazione e soprattutto la facilità di interpretazione dei grafici mostrati, che hanno creato la necessità di modificare alcuni componenti grafici già esistenti. Gli sviluppi volti in questa direzione non hanno portato modifiche particolarmente impattanti a livello di funzionalità, ma hanno avuto l'obiettivo, di certo non trascurabile, di migliorare la user experience del software.
- La seconda tipologia, invece, coinvolge l'implementazione di nuove funzionalità al fine di arricchire ed estendere *Weka,* introducendo il filtraggio dei dati in tempo reale e altre funzioni che si avrà modo di evidenziare nel proseguo.

[1] I file csv (Comma-Separated Values) permettono di definire all'interno della loro prima linea i nomi delle colonne descritte nel proseguo del file, separati dal carattere ",", mentre nelle righe successive vengono scritti i valori corrispondenti alle colonne definite, anch'essi separati da virgole. *Weka* utilizza un formato più evoluto di nome arff (Attribute Relationship File Format), che assomiglia al csv ma permette di definire alcuni vincoli sul modello dei dati rappresentato.

Si ha avuto quindi come obiettivo quello di ottenere un software che fosse estremamente malleabile, ovvero che fosse in grado di adattarsi con facilità a qualunque set di dati ricevuto in input, senza aver bisogno di modifiche o configurazioni manuali; infatti, l'unico requisito per poter utilizzare *Weka* è quello del formato dei dati forniti all'applicazione, ovvero il formato arff (Attribute Relationship File Format)[2]. Questa particolare estensione si può considerare come un file csv (Comma Separated Values) a cui sono state attribuite delle regole di configurazione proprie di ciascuna colonna, risultando di fatto una versione decisamente più evoluta di quest'ultimo. Nel dettaglio, i file utilizzati da *Weka* si possono idealmente suddividere in due parti:

- la prima rappresenta una parte di configurazione, che racchiude la parte semanticamente più potente del formato, e prevede, oltre alla definizione della tabella che si sta scrivendo, la definizione mediante l'utilizzo di particolari annotazioni (identificati con il carattere @) degli attributi raccolti nel dataset, attribuendo a ciascuno:
    - il *nome*, che identifica univocamente la colonna;
    - il *tipo di dato*, che può essere di varia natura (nominal e numeric sono i più utilizzati);
    - eventualmente una lista di *valori accettabili*[3], nel caso in cui l'attributo in questione possa contenere solamente un limitato set di valori.
- la seconda parte contiene i dati concreti del dataset, che sono rappresentati nel tradizionale formato csv[4], ovvero ciascun valore è separato dal successivo da un carattere di delimitazione, nel nostro caso la virgola (,).

In Figura 1 possiamo vedere un esempio di file di input utilizzato. Analizziamo nel dettaglio le varie parti che compongono il file:

- Con l'annotazione *@relation* si indica il nome degli oggetti che si sta andando a definire;

[2] In alternativa è possibile utilizzare *Weka* per compiere delle elaborazioni su set di dati presi direttamente da un database per mezzo di uno dei connettori supportati dal software.
[3] Questo è il caso degli attributi *nominal*, che prevedono un elenco di valori che possono assumere.
[4] Dal momento che gli attributi, vale a dire i nomi delle colonne, sono definiti nella precedente parte di configurazione all'interno del file arff, in questa sezione la prima riga, che in un csv classico rappresenta l'header della tabella rappresentata, non è presente.

- Con l'annotazione *@attribute* vengono definiti gli attributi che verranno scritti nei record del file: nell'esempio proposto l'attributo "outlook" corrisponderà alla prima colonna di ogni record inserito, l'attributo "temperature" alla seconda e così via. Si può notare come per alcuni attributi vengano indicati tra parentesi graffe i soli valori possibili per quell'attributo, mentre per altri sia presente la keyword *real*[5], che viene utilizzata per indicare i valori numerici; possiamo affermare, per esempio, che per l'attributo "play" sono accettati solamente i valori "yes" o "no", mentre per l'attributo "temperature" sono validi tutti i valori rappresentabili da un numero reale.
- A seguito dell'annotazione *@data* sono presenti i dati veri e propri: ogni colonna, che fa riferimento a uno degli attributi definiti in precedenza, viene separata dalla successiva da una virgola, e ogni record viene scritto su una riga separata.

```
@relation weather
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
```

Figura 1 - Esempio di file in formato arff

[5] La keyword *real* può essere sostituita con *numeric,* come nel caso dei dataset utilizzati nel corso di quest'opera. Le due notazioni possiedono il medesimo significato, indicando entrambe un attributo che può assumere valori numerici.

1.2 Metriche di qualità video

Prima di esaminare nel dettaglio la soluzione proposta all'interno di questo progetto di tesi, si ritiene importante, almeno ad alto livello, presentare il concetto di analisi della qualità video. La stima della qualità di un prodotto, nel nostro caso un contenuto grafico, è ricorrente in ogni ambito e nel corso degli anni si sono sviluppate tecniche di previsione sempre più precise ed affidabili. Nell'ambito dell'analisi video, in particolare, sono presenti due tipologie di modelli di stima: la prima prevede l'analisi delle caratteristiche misurabili in modo oggettivo, con l'obiettivo di costruire una scala di qualità basata su di esse. La seconda, invece, si basa sulle opinioni soggettive delle persone, che attribuiscono, secondo vari criteri, un punteggio al video preso in considerazione.

Il motivo per il quale risulta interessante l'analisi della qualità video, e la conseguente adozione di un metodo affidabile per la sua determinazione, è legato al fatto che essa può essere influenzata dalle numerose operazioni, come per esempio la compressione, che un video subisce nelle fasi precedenti alla sua visualizzazione.

Per approssimare la qualità di un video, come già accennato, si può ricorrere all'utilizzo di due diverse tipologie di metriche:

- *Metriche soggettive:* le metriche soggettive prevedono che un gruppo di persone, che non devono necessariamente essere esperte di qualità video, dal momento che risulta interessante solamente la percezione soggettiva dell'individuo, osservino delle sequenze video e ne esprimano un giudizio sulla qualità percepita. Solitamente questi esperimenti vengono condotti in ambienti controllati, che rispecchiano precise caratteristiche volte a non alterare la percezione del video visualizzato, al fine di garantire la bontà dei giudizi forniti dagli utenti. Una volta raccolti i risultati di questi test, essi vengono utilizzati per calcolare il MOS[6] (Mean Opinion Score), che servirà come riferimento con il quale confrontare i risultati ottenuti dalle metriche oggettive, che potranno in questo modo essere valutate.

[6] Il MOS è una metrica che viene utilizzata con le stesse modalità in numerosi altri contesti che non sono in relazione con l'analisi di qualità video.

- *Metriche oggettive:* le metriche oggettive, a differenza delle precedenti, si basano su misurazioni scientifiche di grandezze oggettive, e hanno l'obiettivo di predire con il miglior grado di approssimazione possibile il MOS. La necessità di ricercare una metrica oggettiva affidabile nasce dal fatto che, rispetto alle precedenti, esse sono notevolmente meno costose e più veloci da utilizzare, basandosi fondamentalmente sulla raccolta e sull'analisi di dati fisici. All'interno di questa tipologia di metriche si possono distinguere tre classi:
    - *Full-Reference (FR):* le metriche full-reference prevedono il confronto tra l'immagine o il video originale e quello distorto preso in esame. Le più famose che appartengono a questa categoria sono PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity) e VMAF (Video Multimethod Assessment Fusion).
    - *Reduced-Reference (RR):* le metriche reduced-reference prevedono di avere a disposizione solamente alcune caratteristiche dell'immagine o del video originale, con le quali vengono messe a confronto quelle del contenuto analizzato.
    - *No-Reference (NR)*: queste metriche non richiedono la conoscenza di alcuna informazione sul contenuto originale, ma basano la loro stima della qualità esclusivamente sulle caratteristiche intrinseche dell'immagine o del video.

A titolo di completezza prendiamo in analisi alcune delle metriche più diffuse.

1.2.1 MOS

Il MOS (Mean Opinion Score) è un metodo soggettivo che viene utilizzato per stimare la qualità di un determinato servizio, risultando dunque applicabile non solo nel campo dell'analisi di qualità video. Per calcolarlo numerosi soggetti vengono sottoposti ad un test durante il quale viene chiesto di attribuire un punteggio alla qualità del contenuto visualizzato su una scala da 1 a 5. A test terminato verranno raccolti tutti i dati forniti dai tester e verrà calcolata la relativa media che andrà a determinare il valore del MOS che rappresenterà la stima della qualità percepita da un individuo. Esistono, inoltre, altri tipi di MOS, in particolare DMOS (Degradation

Mean Opinion Score) prevede una valutazione del grado di distorsione dell'immagine percepito, mentre CMOS (Comparison Mean Opinion Score) prevede l'attribuzione di un punteggio in relazione al confronto di uno dei due contenuti con quello ad esso associato.

1.2.2 PSNR

Il PSNR (Peak Signal to Noise Ratio) è un metodo oggettivo che ricade nella classe dei Full-Reference ed esprime il degrado dell'immagine o del video preso in esame mediante il rapporto tra la potenza massima di un segnale e il segnale degradato dal rumore. Il PSNR, data la natura dei dati analizzati, è solitamente espresso in decibel su scala logaritmica. Il PSNR, data un'immagine I senza rumore e un'immagine K con distorsione, può essere stimato mediante l'MSE (Mean Squared Error o errore quadratico medio):

$$MSE = \frac{1}{n * m} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Dove n e m sono le dimensioni dell'immagine, i e j sono gli indici necessari ad effettuare il ciclo su ogni pixel, $I(i,j)$ rappresenta il valore dell'intensità luminosa del pixel nella posizione indicata dagli indici i,j e $K(i,j)$ rappresenta il valore del pixel nell'immagine distorta K. Il PSNR, dunque, viene definito, in decibel, come:

$$PSNR = 10 * log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

Dove MAX_I è il valore massimo che il pixel può avere.

1.2.3 SSIM

Il SSIM (Structural Similarity) è un metodo oggettivo utilizzato per misurare il grado di somiglianza tra due immagini e per questo motivo ricade anch'esso, come PSNR, nella categoria delle metriche Full-Reference. SSIM è nato per fornire un'approssimazione migliore di algoritmi precedenti quali il PSNR ed è stato scientificamente provato che tale metrica risulta particolarmente affidabile se accostata ai risultati delle metriche soggettive. Una delle caratteristiche principali di SSIM, che lo differenziano dalla maggior parte delle altre metriche, risiede nel

fatto che esso tenta di stimare l'errore percepito dall'immagine distorta. SSIM viene calcolato su varie finestre dell'immagine, in particolare se consideriamo due finestre x e y di grandezza uguale NxN possiamo scrivere:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Dove:

- μ_x e μ_y rappresentano la media dei valori di x e y;
- σ_x^2 e σ_y^2 rappresentano la varianza di x e y;
- σ_{xy} rappresenta la covarianza di x e y;
- $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ dove L rappresenta il range dinamico dei valori dei pixel e $k_1 = 0.01$ e $k_2 = 0.03$.

1.2.4 VMAF

VMAF (Video Multimethod Assessment Fusion) è una metrica oggettiva che si colloca nella classe dei Full-Reference. VMAF è stato sviluppato da Netflix e attualmente rappresenta lo stato dell'arte per quanto riguarda i metodi di calcolo della qualità video. Come suggerisce il nome, VMAF trae beneficio dall'utilizzo congiunto di molteplici metriche di qualità dell'immagine, tra le quali figurano:

- *VIF* (Visual Information Fidelity): prende in considerazione la perdita di fedeltà dell'informazione;
- *DLM* (Detail Loss Metric): misura la perdita di dettaglio dell'immagine;
- *MCPD* (Mean Co-Located Pixel Difference): misura la differenza temporale della componente luminosa su fotogrammi differenti;
- AN-SNR (Anti-Noise Signal-to-Noise Ratio)

Il risultato ottenuto dalla combinazione delle metriche precedenti viene rappresentato da un punteggio su una scala da 0 a 100 per fotogramma analizzato, dove 100 rappresenta una qualità identica a quella dell'immagine originale di riferimento. Vengono infine raccolti i punteggi di tutti i fotogrammi che compongono il video in esame e su di essi viene effettuata una media che determina la valutazione complessiva del contenuto.

1.3 Progetto JEG

Il dataset che si ha avuto modo di analizzare nel corso di questo lavoro di tesi, relativo alle misure di qualità video, è stato reso disponibile dal JEG[1] (Joint Effort Group). Tramite l'utilizzo del software originale *Weka,* si è potuta compiere una prima analisi dei dati messi a disposizione dal dataset, in seguito alla quale si è deciso di intraprendere un processo di perfezionamento delle funzionalità offerte dal software. Gli sviluppi, inoltre, hanno previsto l'introduzione di alcune implementazioni assenti nella versione originale di *Weka*, al fine di ottenere una presentazione esaustiva dei dati analizzati all'interno del software. I dataset utilizzati nel corso dell'opera sono stati ottenuti grazie ad un'attività seguita anche dal Politecnico di Torino, in particolare dal prof. Enrico Masala.

All'interno di JEG sono presenti numerosi progetti relativi all'analisi qualitativa di dati ottenuti a partire dalla visualizzazione di alcune sequenze video, in particolare *Hybrid database for HEVC* ha il compito di realizzare un database di larga scala contenente i risultati di misure oggettive di numerose sequenze video. Grazie all'utilizzo di un software quale *Weka,* tale database permetterà di evidenziare le relazioni tra i valori dei dati analizzati.

Il database in questione ha una dimensione notevole, in quanto supera i 300 Gigabytes di spazio utilizzato, ed è suddiviso in varie cartelle, che contengono i video analizzati in formato esteso (all'interno della cartella SRC), e compresso (nella cartella ENC), mentre all'interno della cartella FR sono presenti i file csv relativi alle misure qualitative dei video già analizzati. L'obiettivo di tale progetto è quello di studiare, e possibilmente di affinare, i modelli di misure oggettive e soggettive dei dati di qualità video, evidenziandone eventuali differenze a parità di sequenza video analizzata.

Nonostante la volontà di sviluppare e quindi ottenere uno strumento di analisi universale, che quindi sia in grado di gestire un qualsiasi file arff fornito in input al software, durante lo svolgimento della tesi ci si è focalizzati sui dataset riguardanti la raccolta di dati relativi alla qualità del video, in particolare su quelli derivati da HEVC. I dati in esso contenuti fanno riferimento all'analisi di 13 sequenze video e sono stati resi disponibili dal sopra citato JEG, come si può anche notare dal nome della relazione definita nel file arff.

Gli attributi presenti in questo dataset sono numerosi e il dettaglio di ciascuno di essi esula dallo scopo di questa tesi, tuttavia si ritiene interessante presentare gli attributi più rilevanti ai fini delle analisi effettuate con il software realizzato durante questo progetto. Si possono dunque segnalare, tra quelli presenti all'interno del dataset in questione, i seguenti attributi:

- *seq*: è un valore compreso tra 01 e 13, che indica a quale delle 13 sequenze video analizzate si riferisce il record in questione.
- *w*: rappresenta la larghezza del fotogramma analizzato, può assumere solamente tre valori distinti, *1920*, *1280*, *960*.
- *h*: rappresenta l'altezza del fotogramma analizzato, può assumere solamente tre valori distinti, *540*, *720*, *1080*.
- *psnr000*: PSNR (Peak Signal-to-Noise Ratio) è una misura che permette di valutare la qualità di un'immagine compressa rispetto a quella originale. Viene definita come il rapporto tra la massima potenza di un segnale e la potenza di rumore che può invalidare la fedeltà della sua rappresentazione compressa. Viene rappresentato su una scala logaritmica di Decibel
- *ssim000:* l'indice SSIM (Structural Similarity) è un metodo di predizione della qualità percepita a riguardo di un video. Il modello venne sviluppato all'Università del Texas ad Austin e successivamente migliorato all'Università di New York e all'Università di Waterloo.
- *msssim000:* l'indice MSSSIM (Multi Scale Structural Similarity) è una variante del precedente metodo che ne migliora le performance.
- *vqm000:* VQM (Video Quality Metric) è un modello che tiene conto anche della degradazione della qualità dell'immagine nel corso del tempo.
- *vmaf0.6.0:* VMAF (Video Multimethod Assessment Fusion) è la metrica oggettiva che attualmente rappresenta lo stato dell'arte nella valutazione della qualità video; è stata sviluppata da Netflix in collaborazione con l'Università della California del Sud e l'Università del Texas ad Austin.

```
@RELATION HEVC_JEG_59520seq_metrics

@ATTRIBUTE seq  {01,02,03,04,05,06,07,08,09,10,11,12,13}
@ATTRIBUTE w  {1920,1280,960}
@ATTRIBUTE h  {544,720,1080}
@ATTRIBUTE reqrate
{-1,500000,1000000,2000000,4000000,8000000,16000000}
@ATTRIBUTE qp  {-1,26,32,38,46}
@ATTRIBUTE rc  {CBR,FIXEDQP}
@ATTRIBUTE bitratetype  {FRAME,LCU,NA}
@ATTRIBUTE gop  {2,4,8}
@ATTRIBUTE goptype  {NORMAL,LOWDELAY}
@ATTRIBUTE refresh  {OPENGOP,CLOSEDGOP}
@ATTRIBUTE intraperiod  {8,16,32,64}
@ATTRIBUTE searchrange  {64}
@ATTRIBUTE bitdepth  {8}
@ATTRIBUTE slicearg  {0,2,4,1500}
@ATTRIBUTE hrc  STRING
@ATTRIBUTE effqp  NUMERIC
@ATTRIBUTE effbitrate  NUMERIC
@ATTRIBUTE psnr000  NUMERIC
@ATTRIBUTE ssim000  NUMERIC
@ATTRIBUTE msssim000  NUMERIC
@ATTRIBUTE vifp000  NUMERIC
@ATTRIBUTE vqm000  NUMERIC
@ATTRIBUTE pvqm000  NUMERIC
@ATTRIBUTE vmaf0.6.1  NUMERIC
@ATTRIBUTE vmaf0.6.0  NUMERIC

@DATA
01,1920,1080,1000000,-1,CBR,FRAME,2,NORMAL,OPENGOP,16,64,8,0,
GOP2_1000000_1_16_64_8_
0,39.312,1043379.2,38.123456,0.969835,0.990072,0.648672,0.203
4329,11.368428,87.958373,92.370611
```

Figura 2 - Struttura del dataset utilizzato

1.4 Strumenti esistenti

Come accennato in precedenza, erano già presenti in rete numerosi strumenti che permettevano l'analisi di dataset per mezzo di ausili grafici come rappresentazioni di piani cartesiani o di diagrammi e ciascuno di essi presentava pregi e difetti; per decidere quale fosse il candidato ideale per gli scopi di quest'opera si è dovuta effettuare un'analisi preventiva di quanto ciascuno di essi fosse in grado di offrire nella sua versione distribuita, in modo da avere un software di partenza che fosse da un lato solido e dall'altro facilmente estendibile. Per questo motivo, prima di ricadere sulla scelta di *Weka,* si sono messi a confronto gli strumenti più diffusi in questo settore, di cui si riassumono brevemente le principali caratteristiche.

1.4.1 Rapidminer

Rapidminer[2] rappresenta una delle alternative più valide a *Weka*, di cui ne condivide gli scopi essendo anch'esso un software utilizzato nell'ambito del machine learning. YALE (Yet Another Learning Environment), così era chiamato originariamente il software che sarebbe poi diventato nel 2007 Rapidminer, nasce nel 2001 nella Technical University di Dortmund in Germania dall'idea di Ralf Klinkerberg, Ingo Mierswa e Simon Fischer. Come *Weka* anche Rapidminer è scritto in Java e comprende un'ampia scelta di operatori, più di cinquecento. A partire dal 2006 lo sviluppo di Rapidminer è stato affidato alla Rapid-I, società fondata dai primi due ideatori del software; dal nome della società deriverà il conseguente cambio di nome del prodotto nell'anno successivo. Nel 2013, infine, a fronte del successo ottenuto su scala globale da Rapidminer, viene rinominata la stessa azienda con tale nome. La versione base del software è gratuita, distribuita sotto licenza AGPL (Affero General Public License). Nella sua versione gratuita, però, Rapidminer presenta una serie di limitazioni che ne riducono la possibilità di utilizzo per gli scopi di questa tesi; infatti, tra le feature mancanti rispetto alle versioni a pagamento (Professional ed Enterprise Edition), si può notare che la versione base di Rapidminer prevede un massimo di diecimila righe analizzate, non sufficienti per i dati a disposizione, e di poter ricorrere ad un solo processore logico. Un altro punto a svantaggio di Rapidminer se confrontato con *Weka* è la presenza di un'interfaccia grafica non particolarmente ricca, soprattutto se si prende in considerazione la versione gratuita e open source del prodotto, motivo per il quale si è preferita l'adozione del software neozelandese.

1.4.2 Google Data Studio

Google Data Studio[3] è uno strumento che permette all'utente di creare svariate tipologie di report e grafici, tra i quali figurano gli scatter plot, a partire dai dataset forniti in input, che possono provenire da altri servizi Google, per esempio da file di Google Sheets con i quali Google Data Studio e perfettamente integrato, o anche da fonti esterne. Nasce inizialmente come parte della suite enterprise di Google Analytics 360 e nel Maggio 2016 Google annuncia il rilascio di una versione gratuita rivolta a piccoli team e a singoli utenti. Tale versione diventerà

interamente gratuita in seguito al rilascio di Febbraio del 2017. Come la maggior parte dei prodotti Google, anche Data Studio è un software cloud-based, con un'interfaccia familiare a chi già conosce l'ambiente dell'ormai famoso Google Drive, da cui si ispira. Il principale svantaggio di questo strumento, in relazione a quanto ricercato per gli scopi di questa tesi, risiede nel fatto che Google Data Studio non è un progetto open source, e per questo motivo non risulta possibile andare a modificare i componenti presenti al suo interno. Google Data Studio, in ogni caso, risulta un tool di analisi dati decisamente performante e completo.

1.4.3 Rawgraphs

Rawgraphs[4] è un framework di visualizzazione dei dati che ha come obiettivo quello di rendere la rappresentazione grafica e la conseguente analisi dei dati mostrati intuitiva anche per un utente poco esperto. Il progetto, che è sviluppato e mantenuto dal DensityDesign Research Lab del Politecnico di Milano, è stato rilasciato al pubblico nel 2013 e rappresenta uno degli strumenti più rilevanti nell'ambito della visualizzazione dei dati. Rawgraphs è un soluzione client-based, quindi utilizzabile direttamente da un comune browser, scritta in javascript. Nonostante Rawgraphs presenti numerosi aspetti favorevoli per la sua adozione per gli scopi di questa tesi, dal momento che è distribuito gratuitamente, il codice è facilmente reperibile trattandosi di un progetto open source e la sua interfaccia è estremamente intuitiva, esso presenta una limitazione non trascurabile se si deve analizzare una grossa mole di dati; Rawgraphs, infatti, sebbene lavori molto bene su dataset di dimensioni contenute, soffre a livello di prestazioni se utilizzato per analizzare set di dati contenenti migliaia di record, come nel caso delle misure di qualità video oggetto di quest'opera.

A valle di quanto visto rapidamente per gli strumenti analizzati, si possono comprendere le motivazioni che hanno portato alla scelta di *Weka* per compiere le analisi ricercate: esso permette di essere studiato, e, ancora più importante, facilmente modificato, dal momento che il software è open source; offre inoltre una buona interfaccia grafica, che rappresenta il punto di partenza ideale per gli sviluppi effettuati, e consente l'analisi di dataset anche di grandi dimensioni, che ha

rappresentato un elemento fondamentale nella scelta di *Weka* rispetto agli strumenti sopra citati.

2 Weka

Weka (Waikato Environment for Knowledge Analysis) è un software nato nell'università di Waikato[5] in Nuova Zelanda con lo scopo di implementare funzionalità di apprendimento automatico. L'intero progetto è open source[7] e viene distribuito con licenza GNU General Public License. Il nome del software, anche se come già accennato rappresenta un acronimo, richiama il nome di un comune uccello caratteristico della Nuova Zelanda, motivo per il quale esso è stato adottato come logo dell'applicazione.

Figura 3 - Esemplare di Weka

2.1 Storia

La prima versione di *Weka* fu sviluppata nell'università di Waikato nel 1993; essa era molto diversa da come è diventato il software al giorno d'oggi. Fu originariamente scritto utilizzando C, Tcl/TK, e Makefiles. Successivamente, nel 1997, gli sviluppatori di *Weka* decisero di riscrivere l'intero software utilizzando un altro linguaggio di programmazione che stava facendo il suo ingresso sulla scena mondiale in quel periodo, e che sarebbe diventato uno dei linguaggi di programmazione più utilizzati in tutto il mondo negli anni successivi, ossia Java[8]. Gli sviluppatori non si limitarono a riscrivere la parte di interfaccia del software, bensì reimplementarono anche gli algoritmi di modeling presenti al suo interno. Negli anni successivi, quelli a cavallo dell'anno 2000, *Weka* conobbe un'enorme diffusione su scala globale, fino a meritare, nel 2005, il *SIGKDD Data Mining and*

[7] Il codice di Weka è accessibile gratuitamente dal repository Git presente all'indirizzo https://github.com/Waikato/weka-trunk[6]. La versione da cui è stato clonato il progetto è quella presente sul branch *master* in data 4 Aprile 2018.

[8] Java è un linguaggio di programmazione ad oggetti ampiamente diffuso. Durante il corso del progetto si sono consultate ripetutamente le pagine di documentazione[7] relative agli oggetti utilizzati al fine di comprendere a fondo i dettagli implementativi.

Knowledge Discovery Service Award, il più alto riconoscimento nel campo del Data Mining. Nel 2006 la Pentaho Corporation acquisì una licenza esclusiva che gli permetteva di utilizzare *Weka* per la business intelligence, andandosi ad integrare nella Pentaho[8] business intelligence suite. Al momento della stesura di quest'opera, *Weka* rimane ancora uno dei software di machine learning più diffusi ed utilizzati al mondo.

2.2 Guida all'installazione di Weka in Eclipse

Avvertenza: la seguente guida è stata testata utilizzando Eclipse Oxygen.
L'intero software è stato sviluppato utilizzando Eclipse Oxygen[9] come ambiente di sviluppo (IDE) unitamente a Git come strumento di version control, dunque per aver accesso al codice è sufficiente importarlo dal repository su cui risiede seguendo questa procedura[10]:

- Aprire Eclipse;
- Per importare il progetto contente il codice sorgente di *Weka* da Git selezionare File > Import…;
- Dalla finestra che si apre selezionare Git > Projects from Git e quindi cliccare su Next >;
- Nella schermata successiva selezionare Clone URI e quindi Next >;
- Nel campo URI inserire il seguente indirizzo: https://gitlab.com/alessandro.catto/Weka.git[11]; esso rappresenta l'indirizzo della versione di *Weka* sviluppata nel corso di questa tesi. Una volta inserito, i restanti campi verranno automaticamente popolati (ad eccezione di quelli relativi all'autenticazione se non precedentemente impostate). Cliccare ancora su Next >;

[9] Si è mantenuto l'utilizzo dello stesso IDE (Integrated Development Environment) durante tutto il corso del progetto, mantenendo sempre la versione da cui si sono cominciati gli sviluppi.
[10] Tale procedura è stata testata utilizzando la versione di Eclipse indicata su una macchina con installato Windows 10.
[11] In questo repository sono presenti due branch, *master* contiene la versione stabile più recente, *develop* è stato utilizzato per effettuare gli sviluppi (in un primo momento era presente un solo branch).

- Comparirà la schermata di selezione del branch; selezionare il branch desiderato (se non si è sicuri su cosa scegliere selezionare *master*) e premere Next >;
- Selezionare la cartella dove verrà salvato il progetto importato da Git;
- Dopo aver cliccato su Next > inizierà il processo di clonazione del repository Git. Una volta terminato controllare che sia selezionato il campo *Import existing Eclipse projects* e quindi cliccare su Next >;
- Nell'ultima schermata lasciare i valori di default e cliccare su Finish.

Terminata la procedura per l'importazione del progetto, lo stesso comparirà nel Project Explorer di Eclipse (se il Project Explorer non è visibile, dal menù di Eclipse selezionare Window > Show view > Project Explorer).

A questo punto si ha a disposizione il codice sorgente della versione di *Weka* proposta all'interno di quest'opera; è possibile dunque analizzarlo ed eventualmente modificarlo continuando a contribuire al progetto mediante l'utilizzo di Git.

Per lanciare l'applicazione all'interno di Eclipse, fare clic con il pulsante destro del mouse sul progetto nel Project Explorer e selezionare Run as > Java Application

Dalla finestra che comparirà selezionare la classe Java di nome GUIChooser e quindi premere OK.

L'applicazione verrà lanciata e si potrà iniziare ad utilizzarla nella sua interezza.

Nota bene: la versione di *Weka* presente sul repository precedentemente indicato include già al suo interno il plug-in che permette di visualizzare il tab *Visualize3D*, non presente nella versione originale del software, utile per visualizzare gli scatter plot in uno spazio tridimensionale.

3 Interfaccia grafica

3.1 Versione originale

Weka, essendo un progetto di ampia portata e largamente diffuso, si compone di moltissimi moduli dipendenti tra loro; esso nasce come software per il machine learning offrendo numerosi strumenti ad esso collegati.

Figura 4 - Schermata iniziale di Weka

All'avvio dell'applicazione viene proposta un'interfaccia grafica di benvenuto che offre una barra dei menu da cui è possibile modificare le impostazioni ed accedere alle funzionalità offerte dal software. Da questa schermata è possibile scegliere quale delle modalità (Applications) di *Weka* si desidera utilizzare:

- *Explorer*: è l'ambiente su cui si sono concentrati gli sviluppi effettuati. Permette di esplorare i dati mediante l'utilizzo dei comandi *Weka*; al suo interno sono presenti sei tab che consentono l'accesso alle varie funzionalità disponibili:

    - *Preprocess*: è il tab in cui vengono caricati i dati da analizzare da una base di dati o da file.
    - *Classify*: al suo interno possono essere valutate le prestazioni di diversi algoritmi di machine learning applicati sui dati immessi.
    - *Cluster*: al suo interno possono essere valutate le prestazioni di diversi algoritmi di clustering applicati sui dati immessi.
    - *Associate*: permette di estrarre delle regole di associazione tra gli attributi presenti nel dataset inserito.
    - *Select attributes*: attraverso appositi algoritmi vengono valutati gli attributi in base alla loro utilità per la costruzione di un modello predittivo.
    - *Visualize*: rappresenta il tab in cui vengono visualizzati i grafici relativi ai dati analizzati.
- *Experimenter*: al suo interno possono essere effettuati test statistici tra i diversi algoritmi di data mining.
- *KnowledgeFlow*: è lo strumento che permette di costruire, mediante l'utilizzo di un'interfaccia grafica, il flusso definito per il machine learning. Una volta definito esso può essere eseguito e testato all'interno dello stesso tool.
- *Workbench*: è un ambiente in cui vengono raccolte tutte le funzionalità offerte dalle interfacce grafiche presentate in precedenza.
- *Simple CLI*: offre all'utente la possibilità di interagire in modo diretto con l'applicazione da riga di comando.

Per quanto riguarda il lavoro svolto per questa tesi ci si è focalizzati esclusivamente sulla parte di visualizzazione grafica dei dati analizzati, che viene trattata nella sezione *Explorer* e in particolare all'interno del tab Visualize. Dopo aver caricato il dataset contenente i dati da analizzare dalla scheda nominata *Preprocess*, cliccando sul tab *Visualize* viene mostrata la cosiddetta *plot matrix,* che fornisce all'utente una panoramica dei grafici disponibili per la visualizzazione dettagliata. Tale pannello contiene tutti gli scatter plot che si possono ottenere combinando gli attributi messi a disposizione dal dataset inserito dall'utente. Selezionandone uno viene aperta una nuova finestra che contiene lo scatter plot vero e proprio, che rappresenta la parte centrale degli sviluppi effettuati. Il pannello contenente la *plot matrix* risulta particolarmente interessante in quanto permette di avere una preview generale dei grafici che è possibile andare ad analizzare offrendo già un primo riscontro visivo.

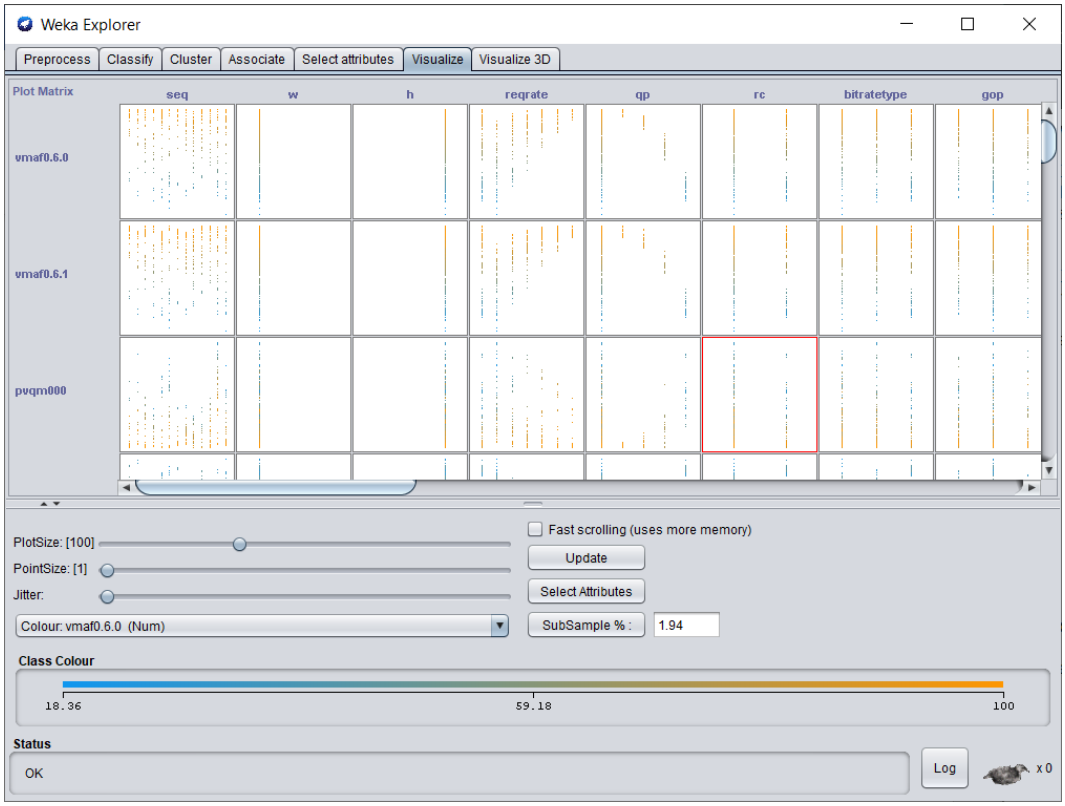

Figura 5 - Plot matrix

All'interno della finestra contenente il grafico, già nella versione originale è possibile impostare e modificare numerose opzioni di analisi e di visualizzazione. Tra le varie operazioni che l'utente può effettuare possiamo citare la possibilità di

selezionare, mediante l'utilizzo di tre pulsanti a selezione multipla (dette *combobox*[12] in Java), gli attributi visualizzati in ciascuna delle tre dimensioni presenti sullo scatter plot (asse orizzontale, asse verticale e colore del punto). Infatti, in ciascuno di questi campi di selezione sono presenti tutti gli attributi definiti nel file analizzato, a cui viene affiancato il tipo di dato rappresentato dallo stesso (*nominal, numeric,* o *string*), permettendo di ricreare ogni singolo grafico già presente sotto forma di preview nella finestra precedente. L'informazione riguardante la tipologia del dato selezionato è fondamentale per quanto riguarda l'attributo relativo al colore (e come vedremo più avanti anche per poter arricchire ulteriormente la qualità dei dati visualizzati) dal momento che esso viene gestito in maniera differente a seconda del tipo dell'attributo selezionato; infatti, come mostra la legenda posta nella parte inferiore della finestra, per gli attributi di tipo *nominal* viene associato un determinato colore per ciascun valore assegnabile a tale attributo (eventualmente personalizzabile), mentre per quelli di tipo *numeric* viene visualizzata una scala di colori che presenta come estremo inferiore il colore blu (valore più basso all'interno del dataset) e come estremo superiore il colore arancione (valore più alto all'interno del dataset).

Un'altra funzionalità particolarmente interessante è quella a cui è dedicata la quarta combobox presente nella parte superiore della finestra; essa permette la selezione di una porzione di grafico utilizzando una delle forme disponibili (*Rectangle*, *Polygon*, o *Polyline*) disegnabili con l'utilizzo di un *mouse* o di un *touchpad*, in modo da focalizzare l'attenzione dell'utente su una parte specifica del piano cartesiano, consentendone un'analisi più dettagliata. Dopo aver effettuato la selezione di interesse, cliccando sul pulsante *Submit* lo stesso viene ricalcolato in modo da visualizzare solamente la porzione di grafico collocata all'interno dell'area selezionata, ricalcolando in maniera contestuale gli estremi di tutte e tre le dimensioni rappresentate (X, Y e colore). A discrezione dell'utente è possibile tornare alla visualizzazione originaria cliccando sul pulsante *Reset* o eventualmente effettuare ulteriori operazioni sul nuovo grafico ridimensionato.

[12] Nel corso degli sviluppi si sono utilizzati gli oggetti messi a disposizione dal framework Swing di Java, che viene utilizzato per la costruzione di interfacce grafiche. Per quanto riguarda le combobox gli oggetti utilizzati sono le *JComboBox*.

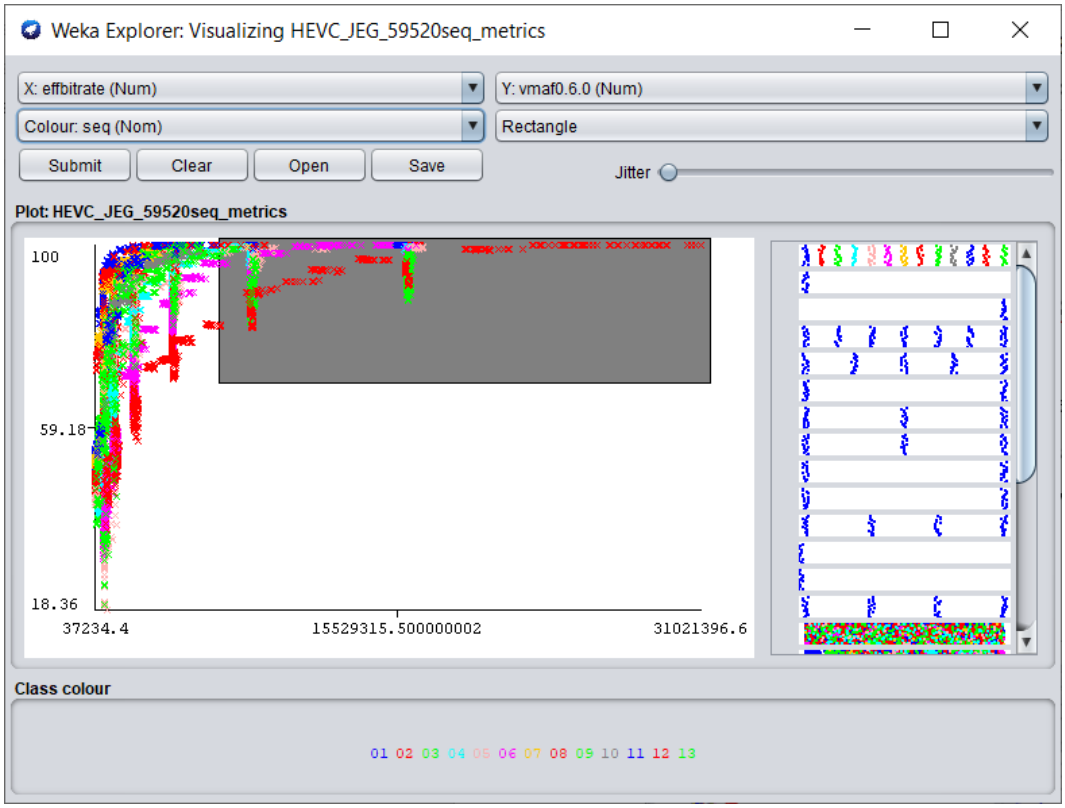

Figura 6 - Selezione di una porzione del grafico

3.2 Miglioramenti ricercati

In seguito ad un primo utilizzo di *Weka* per analizzare i dati contenuti nei dataset a disposizione, si è potuto osservare come esso potesse essere esteso in modo da raffinare, sia visivamente che analiticamente, i risultati ottenuti dall'analisi; si sono potute individuare due direzioni verso cui porre l'attenzione durante gli sviluppi: in primo luogo si è visto come fosse necessario migliorare alcune funzionalità già presenti, in modo particolare in relazione alla semplicità di lettura dei dati visualizzati. Si è ritenuto, infatti, che il software originale potesse essere migliorato grazie ad alcuni accorgimenti a livello grafico che hanno avuto come scopo quello di fornire una visualizzazione ottimale dei dati e soprattutto di facilitarne la loro comprensione all'utente, che deve essere in grado di evidenziare in maniera immediata i dati effettivamente di interesse. Questa necessità risulta evidente se, per esempio, ci si sofferma sulla legenda dei colori visualizzata nella parte inferiore della finestra che contiene il grafico: se l'utente seleziona come attributo associato alla dimensione del colore un attributo di tipo nominal, la legenda conterrà

l'associazione di ciascun valore possibile per l'attributo selezionato con il colore visualizzato sul grafico. La modalità con cui viene costruita e presentata quest'ultima risulta talvolta scomoda, in quanto i nomi dei valori possibili vengono scritti utilizzando il colore di riferimento. Talvolta tale colore non presenta un contrasto sufficientemente elevato con lo sfondo del pannello, soprattutto se si ipotizza una eventuale presentazione della finestra in questione utilizzando un proiettore, che a seconda dell'illuminazione presente nell'ambiente può far sì che il contrasto con il colore di sfondo si riduca ulteriormente. Quest'ultimo esempio rappresenta solo una delle problematiche di visualizzazione rilevate, le altre verranno analizzate nel dettaglio nel proseguo di quest'opera.

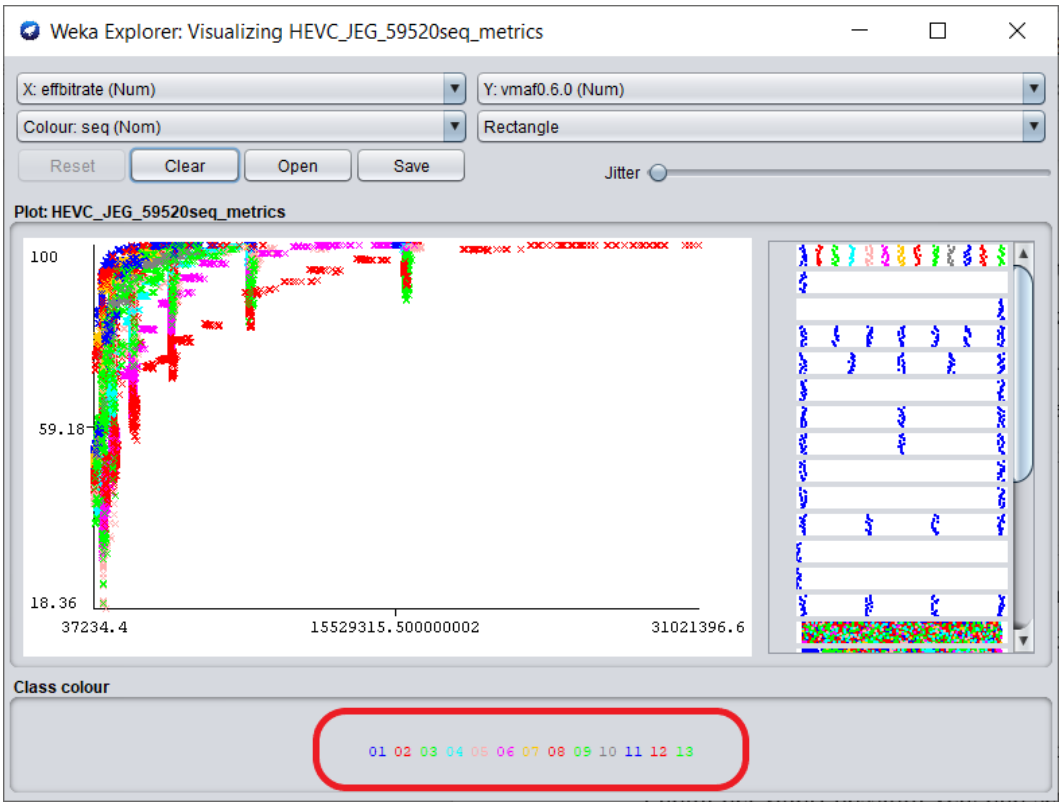

Figura 7 - La legenda di Weka originale

La seconda direzione verso cui ci si è mossi nel corso del progetto di tesi è quella che ha portato all'introduzione di nuove funzionalità, in modo da estendere ed arricchire il software originario. Le implementazioni più rilevanti, solo a titolo esemplificativo dal momento che verranno presentate in dettaglio nel seguito, sono

rappresentate dall'aggiunta di una quarta dimensione (definita *shape*[13], ovvero la forma del punto disegnato sul grafico), che permette di arricchire ulteriormente il dettaglio dei dati visualizzati, e dalla possibilità di filtrare i dati in tempo reale, in accordo a vincoli definiti dall'utente mediante l'utilizzo di apposite interfacce grafiche di semplice comprensione. Si tratta di aggiunte che accrescono ulteriormente la duttilità di un software già molto potente come *Weka*, che, come vedremo, permetteranno di compiere operazioni di visualizzazione anche molto complesse sui dati consentendo di raggiungere il livello di dettaglio necessario ai fini dell'analisi del dataset in questione.

[13] Si è utilizzata la definizione di *shape,* ricorrente all'interno di quest'opera, al fine di essere coerenti con la notazione utilizzata per l'attributo *colour*, che prevede l'uso del termine inglese; inoltre tale nomenclatura risulta più puntuale se paragonata con la relativa traduzione in lingua italiana ("forma del punto").

4 Soluzione proposta

La soluzione pensata e realizzata permette al software *Weka* di svolgere numerose funzionalità aggiuntive, che non sono presenti nella versione originale dello stesso; molte di esse sono facilmente fruibili grazie all'utilizzo di un menu a comparsa che viene visualizzato in seguito alla pressione del tasto destro del mouse sul pannello contenente lo scatter plot visualizzato, all'interno del tab *Visualize*. Il menu in questione fornisce l'accesso a buona parte delle opzioni di visualizzazione introdotte nel corso di quest'opera, le quali verranno prese singolarmente in esame all'interno di questo capitolo.

4.1 Colore dello sfondo

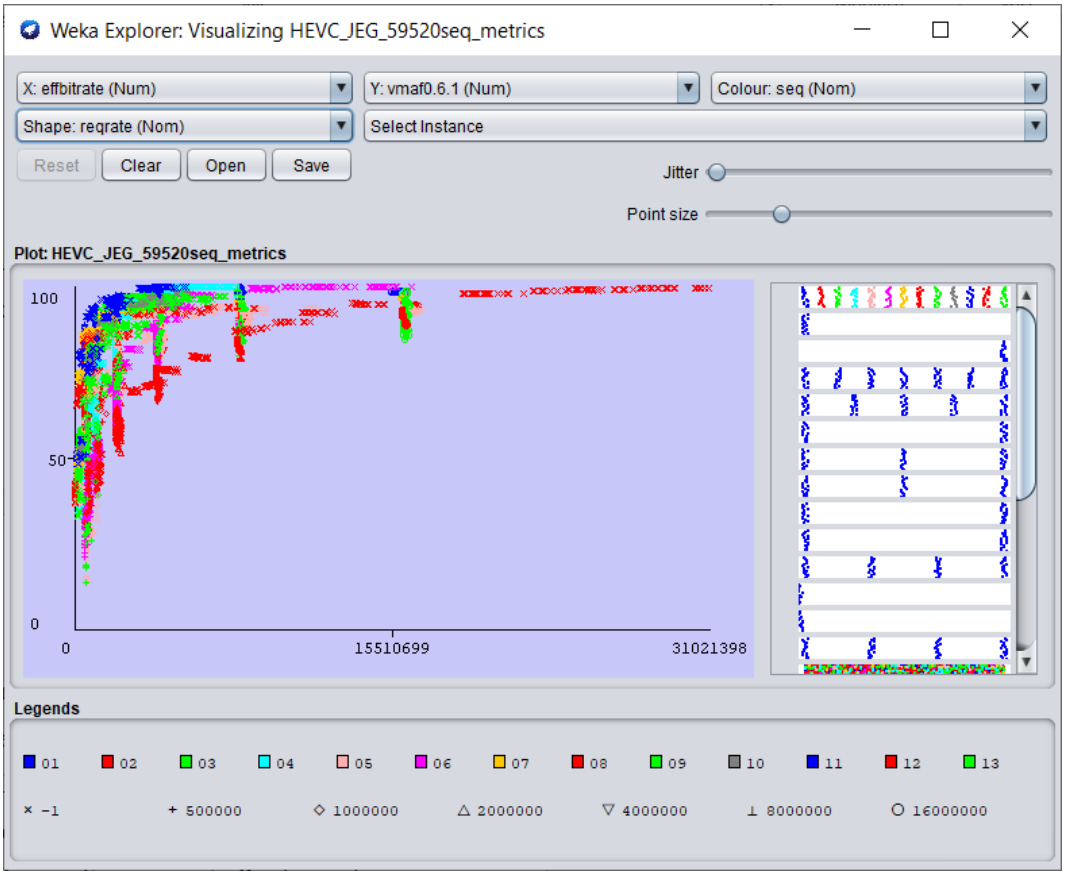

Figura 8 - Colore di sfondo modificato

La prima impostazione che è possibile modificare all'interno del menu è quella riguardante il colore dello sfondo dello scatter plot. Questa modifica si è resa necessaria in quanto già nella versione originale di *Weka* era possibile cambiare i colori predefiniti dei colori utilizzati per stampare i punti sul grafico, ossia quelli relativi all'attributo selezionato nella combobox colour; dal momento che l'utente era in grado di modificare i colori utilizzati per visualizzare i punti tracciati, si è ritenuto necessario dare la possibilità di poter cambiare anche il colore dello sfondo, in modo che quest'ultimo si potesse adattare ai colori scelti per i punti. Per fare ciò si è riutilizzata la stessa libreria, *JColorChooser*[14] per la precisione, che veniva già utilizzata per la selezione dei colori dei punti; essa permette di selezionare, tramite un'apposita interfaccia, il colore da utilizzare, che risulta interamente personalizzabile dall'utente secondo varie modalità, che spaziano dalla selezione di colori campione al classico sistema RGB[15] (Red, Green, Blue).

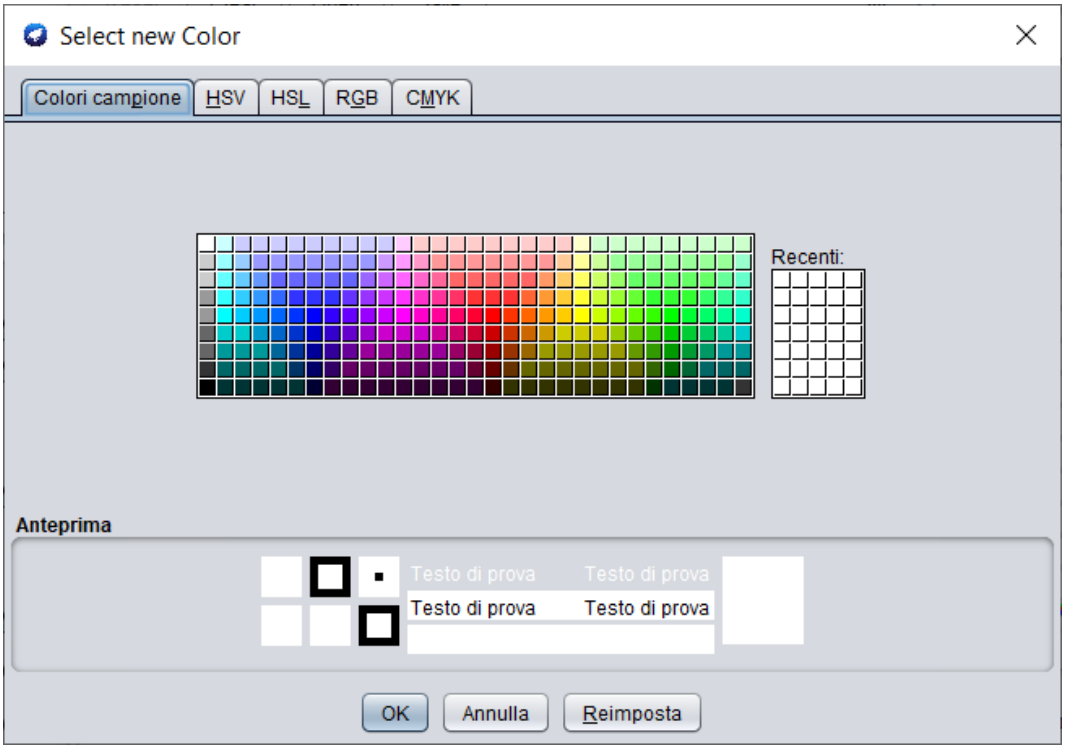

Figura 9 - Pannello di selezione del colore

[14] Fa parte del framework *Java Swing*.
[15] Il sistema *RGB* prevede l'identificazione di un preciso colore mediante la rappresentazione delle sue tre componenti principali (rosso, verde e blu) utilizzando tre numeri compresi tra 0 e 255, dove il colore (0,0,0) rappresenta il nero e il colore (255,255,255) rappresenta il bianco.

4.2 Colore degli assi

Figura 10 - Colore degli assi modificati

Avendo scelto di poter cambiare il colore di sfondo del grafico, non si è potuto fare a meno di fornire all'utilizzatore l'opportunità di poter cambiare anche il colore degli assi dello stesso, in modo da poter selezionare il miglior contrasto possibile tra le due componenti. L'utente è in questo modo in grado di scegliere, secondo le sue preferenze e necessità del momento, la giusta combinazione dei colori utilizzati dall'intero grafico, requisito fondamentale, per esempio, per effettuare una qualsiasi presentazione. Come nel caso precedente, anche per questa modifica è stata riutilizzata la libreria *JColorChooser* per raggiungere lo scopo, che costruisce un'interfaccia grafica analoga a quella presentata in precedenza.

4.3 Visualizzazione della griglia

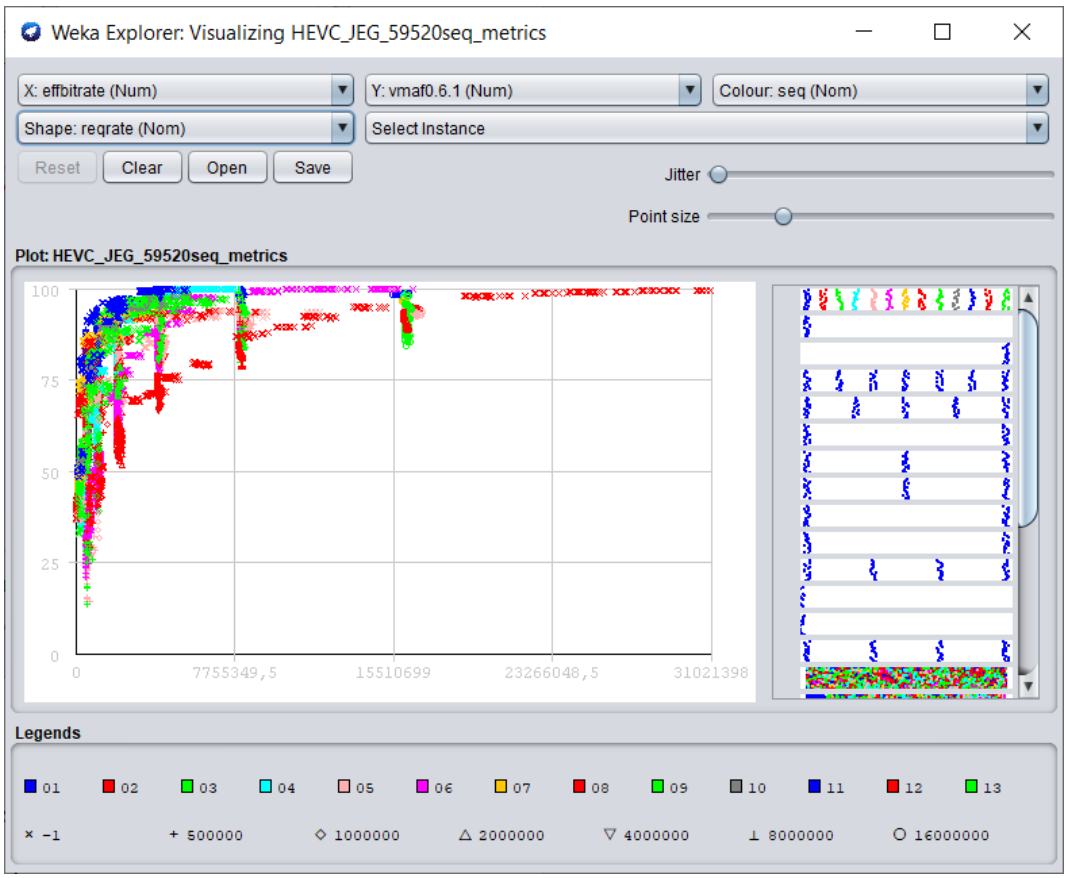

Figura 11 – Visualizzazione della griglia

Uno dei principali motivi per i quali si è scelto di utilizzare *Weka* è stato quello della sua capacità di gestire in modo agevole grandi quantità di dati in ingresso. A fronte di questa premessa si è pensato di creare uno strumento che semplificasse la lettura di tali dati, in qualsiasi situazione e indipendentemente dalla loro natura. Per soddisfare questo requisito si è adottata una soluzione semplice e potente allo stesso tempo; si è deciso di ricorrere all'utilizzo di una griglia, disegnabile su richiesta dell'utente, che guidi quest'ultimo nella lettura dei valori rappresentati. La potenza di questa funzionalità risiede nella possibilità di personalizzare in modo completo la struttura della stessa: per gli attributi nominal, infatti, le linee verranno tracciate seguendo i valori discreti rappresentati sull'asse corrispondente, motivo per il quale non è del tutto apprezzabile l'efficacia di tale funzionalità per questo tipo di dati; quest'ultima risulta invece evidente se si considerano gli attributi di tipo numeric, che rappresentano la tipologia per la quale si è scelto di introdurre questa nuova funzionalità.

Nella versione originale del software, infatti, erano presenti solamente due intervalli rappresentati sugli assi cartesiani, insufficienti a comprendere in modo semplice il valore dei dati visualizzati; grazie a questo sviluppo, invece, la costruzione della griglia diventa interamente personalizzabile, mediante l'utilizzo di uno strumento strettamente correlato, che oltre a ciò permette il ridimensionamento del grafico seguendo le direttive dettate dall'utente. Si rimanda dunque la presentazione di suddette modalità nell'apposita sezione riguardante il ridimensionamento del grafico.

4.4 Colore della griglia

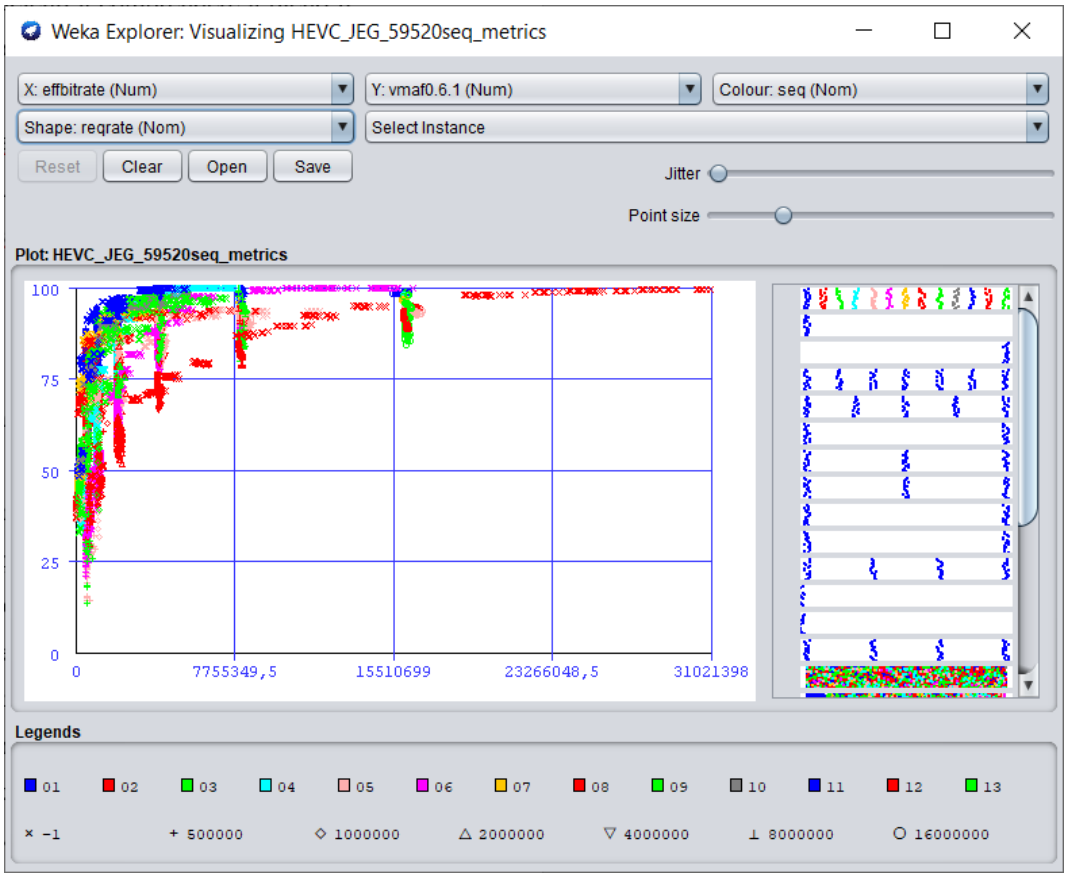

Figura 12 - Colore della griglia modificato

Seguendo la direzione intrapresa per i componenti presi in analisi fino a questo punto, appare evidente come anche il colore della griglia dovesse essere personalizzabile dall'utente, in modo da avere un buon contrasto a seconda del

colore utilizzato come sfondo e di quelli utilizzati per visualizzare gli altri elementi del grafico. Se ciò non fosse possibile potrebbe accadere che il colore della griglia risulti molto simile, se non addirittura uguale, a uno dei colori utilizzati dall'interfaccia per disegnare i punti relativi ai dati, rendendo difficile l'individuazione degli stessi da parte dell'utente. In un primo momento si è pensato di mantenere una relazione tra il colore degli assi e quello della griglia, dal momento che sono strettamente correlati; la scelta finale, invece, è stata quella di mantenerli svincolati, al fine di fornire all'utente la completa personalizzazione di tutti gli elementi presenti nel pannello.

Ancora una volta per ottenere il risultato desiderato si è ricorso all'utilizzo della ormai ben nota libreria *JColorChooser*, che ben si adatta allo scopo e mantiene la continuità di sviluppo intrapresa nelle precedenti implementazioni. L'utilizzo della medesima libreria all'interno di tutti i componenti del pannello volti a personalizzare il colore di uno dei componenti dello scatter plot, inoltre, rende il software maggiormente intuitivo, facilitandone l'apprendimento per ogni tipo di utente.

4.5 Ridimensionamento del grafico

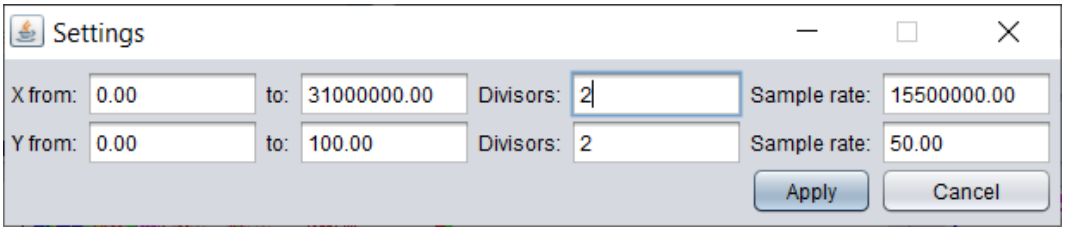

Figura 13 - Finestra di ridimensionamento

Il software originario presentava già la possibilità di selezionare graficamente una porzione dello spazio visualizzato in modo da analizzare solamente i punti compresi all'interno di tale selezione; questo strumento, tuttavia, presentava alcune limitazioni intrinseche, tra cui la più evidente è quella di non poter determinare con precisione, trattandosi di una selezione in termini visivi e quindi non associata ad alcuna grandezza, gli estremi numerici dei range individuati.

Si è dunque scelto di introdurre un secondo metodo di ridimensionamento del grafico, che permette la definizione puntuale degli intervalli di interesse per l'utente[16]; grazie a questo sviluppo è dunque diventato possibile personalizzare con precisione gli estremi dello scatter plot visualizzato, funzionalità che risulta di grande utilità nel caso in cui si voglia limitare l'analisi dei dati in ingresso ad un ben preciso intervallo, già individuato a priori; tale scenario risulta comune nel caso di attributi che possono essere raggruppati in sotto intervalli definiti a partire dalla loro stessa natura.

Da questo stesso pannello è possibile, inoltre, definire il numero di intervalli visualizzato su ciascun asse, informazione che viene utilizzata, come già accennato in una sezione precedente, anche per il disegno della griglia. L'utente può dunque scegliere, a seconda dei propri scopi, tra due differenti modalità di utilizzo dello strumento, in base alle quali verranno suddivisi gli assi cartesiani e, di conseguenza, le porzioni di piano individuate dalle maglie della griglia:

- *Numero di divisioni*: questa prima modalità consente di dividere il relativo asse (ciascun campo presente all'interno di questo pannello si riferisce a una singola dimensione) in parti uguali secondo il valore immesso dall'utente; per esempio se si vuole suddividere l'asse delle ascisse in quattro parti di ugual dimensione sarà sufficiente digitare il numero 4 nell'apposita casella relativa all'asse X; tale operazione andrà contestualmente ad aggiornare il relativo campo di selezione del passo.
- *Selezione del passo:* la seconda modalità permette di suddividere ciascun asse in segmenti di lunghezza fissa a scelta dell'utente; il valore digitato dall'utente, definito come *passo*, viene utilizzato per calcolare gli intervalli da disegnare. Nel dettaglio, a partire dall'estremo inferiore dei dati attualmente visualizzati, viene disegnata una linea ad ogni raggiungimento del passo; appare dunque evidente come l'ultimo intervallo possa essere di lunghezza diversa dai precedenti nel caso in cui l'intervallo presente sull'asse non sia divisibile esattamente per il passo selezionato. Questa

[16] Questo pannello mostra i campi abilitati solamente per gli attributi di tipo numeric. Questo è dovuto al fatto che nel caso degli attributi nominal non è possibile definire degli intervalli, né tantomeno prevedere una suddivisione dell'asse di riferimento secondo un criterio diverso da quello dettato dai valori stessi dell'attributo.

modalità può essere particolarmente utile per alcuni tipi di dati in cui è necessario individuare degli intervalli regolari.

Con l'introduzione di queste diverse tipologie di visualizzazione della griglia l'utente è in grado di tracciare la stessa secondo le proprie necessità.

4.6 Filtraggio dei dati

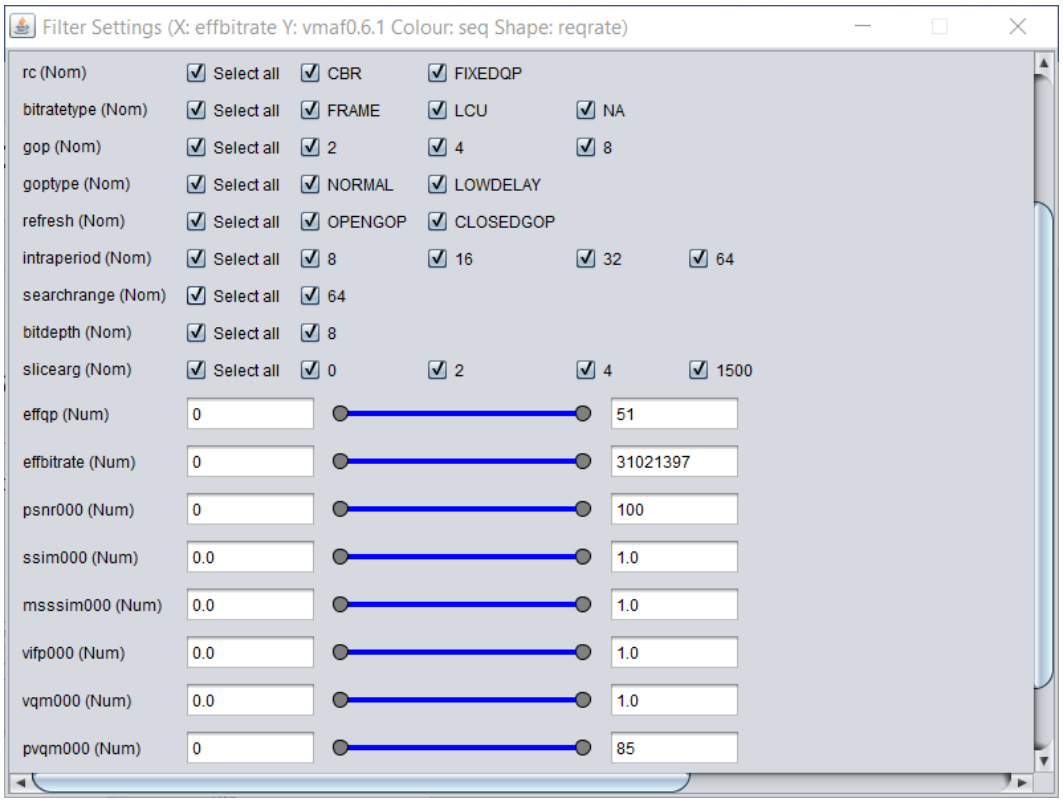

Figura 14 - Finestra di filtraggio dei dati

Dopo aver analizzato le modifiche effettuate con lo scopo di migliorare la semplicità di lettura e la corretta visualizzazione dei dati, in questa sezione verrà introdotto uno sviluppo volto ad aggiungere al software una funzionalità analitica di notevole importanza; quest'ultimo rappresenta una delle implementazioni più rilevanti che sono state effettuate, e fornisce la possibilità, a discrezione dell'utente, di filtrare i dati visualizzati nel grafico secondo uno o più degli attributi presenti nel dataset; questa operazione viene effettuata in tempo reale, ovvero contestualmente alla visualizzazione dei dati stessi, senza bisogno di dover premere alcun pulsante di aggiornamento.

La ragione di questo sviluppo deve essere cercata nella necessità di analizzare non solo gli attributi presenti sugli assi cartesiani, personalizzabili mediante l'utilizzo delle apposite combobox, ma anche di comprendere il rapporto presente tra i dati presenti nel dataset analizzato in funzione degli attributi non presenti tra le grandezze visualizzate esplicitamente sul grafico. Questo strumento, infatti, permette di analizzare i dati inseriti nella loro interezza, permettendone una visualizzazione intuitiva e soprattutto una personalizzazione estremamente semplificata. Analizziamo nel dettaglio il funzionamento di questo potente strumento.

Il pannello di filtraggio[17], così come avviene per quanto già presentato fino a questo punto dell'opera, può essere invocato mediante la sua selezione dal menu a tendina che viene visualizzato alla pressione del tasto destro del mouse sul grafico, selezionando la voce *Filter attributes.* Così facendo si aprirà nella parte alta e a destra dello schermo, un nuovo pannello accanto a quello attualmente visualizzato; al suo interno possiamo osservare tutti gli attributi presenti nel dataset attualmente visualizzato, che possono essere suddivisi, sia dal punto di vista grafico che da quello funzionale, in due categorie:

- Nella parte superiore del pannello vengono presentati gli attributi di tipo nominal*;* questo tipo di attributi rappresenta valori discreti delle grandezze rappresentate, pertanto è possibile filtrare tali valori mediante l'utilizzo di una lista di checkbox[18]. In particolare, ogni attributo nominal è visualizzato su una riga distinta, sulla quale si può distinguere una prima colonna che contiene il nome e il tipo di attributo mostrato, seguito da una lista di checkbox che hanno lo scopo di gestire la logica di filtraggio per tale attributo; tale lista è formata in primo luogo da due checkbox che permettono rispettivamente di selezionare/deselezionare tutti i valori per quella riga, funzionalità che risulta particolarmente utile nel caso di attributi con numerosi valori possibili, e di invertire le selezioni effettuate in precedenza, utile per esempio nel caso in cui si voglia dividere in due gruppi complementari i valori da rappresentare. A seguito di queste prime due,

[17] Verrà chiamato pannello di filtraggio la finestra contenente le opzioni disponibili per effettuare le operazioni di filtraggio sui dati.

[18] Le checkbox vengono rappresentate utilizzando l'oggetto *JCheckBox* del framework *Java Swing.*

viene visualizzata la lista delle checkbox che rappresentano ciascun valore disponibile per quel determinato attributo. L'utente può dunque cliccare su una qualsiasi casella associata ad un determinato valore dell'attributo di interesse, ottenendo al click il ricalcolo e la contestuale presentazione dei dati visualizzati sul grafico, nascondendo, per esempio, i punti che rappresentano uno dei valori deselezionati.

- In coda ai sopra citati attributi, vengono mostrati quelli di tipo numeric*,* ovvero quegli attributi che possono assumere un qualunque valore rappresentabile da una quantità numerica; così come per i precedenti, anche per gli attributi numeric viene mostrata una prima colonna contenente l'attributo e il proprio tipo, a cui segue però un'interfaccia che permette la definizione dell'intervallo da prendere in considerazione per l'attributo in questione. Quest'ultima struttura è formata da due aree di testo[19], una sulla parte sinistra e l'altra su quella destra, e da uno slider a due cursori[20]; le prime hanno il compito di visualizzare i valori numerici degli estremi dell'intervallo selezionato, il secondo, invece, ne permette una quantificazione grafica a partire dal range iniziale. L'utente può filtrare i dati utilizzando una di queste due modalità, strettamente correlate dal momento che l'utilizzo di una ne causa l'implicito e contestuale aggiornamento dell'altra. Se si vuole inserire, in particolare, un intervallo specifico, ovvero un intervallo i cui estremi sono valori individuati a priori dall'utente, è possibile digitare tali valori numerici all'interno delle aree testuali, al fine di definire puntualmente l'intervallo ricercato; viceversa, se si vuole utilizzare un approccio maggiormente empirico, è possibile scorrere i cursori dello slider per osservare, in tempo reale, l'evoluzione della rappresentazione dei dati disegnati sullo scatter plot al variare degli estremi. Per rendere possibile tale comportamento si è ricorso all'implementazione dell'oggetto *RangeSlider* (un particolare tipo di slider che a differenza di quello standard possiede due cursori, rendendolo perfetto per la

[19] Le aree di testo vengono rappresentate utilizzando l'oggetto *JTextArea* del framework *Java Swing.*
[20] Lo slider a due cursori è stato implementato con la classe *RangeSlider,* che estende la classe *JSlider* del framework *Java Swing,* e fornisce i metodi necessari alla gestione dei due cursori e dei relativi valori di minimo e massimo associati.

rappresentazione di intervalli di valori), che definisce gli estremi stessi del range.

L'aggiunta di questa finestra di filtraggio conferisce a *Weka* un livello di personalizzazione enormemente migliorato; infatti, se all'apparenza può sembrare che questo strumento fosse già presente nella versione originale del software, che effettivamente permetteva già di "filtrare" graficamente i dati utilizzando lo strumento di selezione del grafico, in realtà questa nuovo sviluppo consente una completa personalizzazione dei dati visualizzati. Ora l'utente è in grado di filtrare i dati visualizzati operando non solo sugli attributi disponibili sugli assi cartesiani, bensì sull'intero set di attributi, funzionalità che risulta fondamentale se si vogliono analizzare i dati a disposizione senza essere vincolati in alcun modo dalle dimensioni fornite dal grafico. Inoltre, le elaborazioni richieste vengono eseguite in tempo reale, contribuendo al miglioramento ricercato della user experience.

Per realizzare il pannello di filtraggio ci si è appoggiati alle librerie fornite dalla documentazione del progetto *Weka*[9], che ora andremo ad analizzare nel dettaglio.

4.6.1 Librerie utilizzate

Al fine di costruire il pannello di filtraggio dei dati, si è deciso di riutilizzare alcune classi già utilizzate in altri contesti dal software originale; in questo modo è stato possibile fare affidamento su delle funzioni già largamente testate ed utilizzate, senza il bisogno di introdurre alcuna logica ulteriore rispetto agli algoritmi già presenti, che avrebbe portato con sé il rischio di introdurre nuovi bug; inoltre si è mantenuta la continuità con altre funzionalità del software relative alla parte di machine learning, che tuttavia non sono presentate all'interno di quest'opera dal momento che esulano dallo scopo del lavoro di tesi.

Le classi utilizzate in questo ambito sono contenute all'interno del package *weka.filters* a cui è possibile accedere in rete dal momento che il software *Weka*, come già detto in precedenza, è open source[6]. Analizziamole nel dettaglio:

- *Filter.java* è la classe che fornisce i metodi di filtraggio sul set di dati analizzato. Essa è una classe astratta da cui estendono le relative sottoclassi;

queste ultime, mediante la riscrittura[21] dei metodi definiti da questa classe astratta, effettueranno le operazioni vere e proprie sui dati. Questa classe contiene il metodo statico *Instances useFilter(Instances data, Filter)* che consente il filtraggio dei dati nel seguente modo: il primo parametro contiene i dati di partenza, che quindi inizialmente corrispondono al dataset in input, mentre il secondo definisce secondo quali criteri tali dati devono essere filtrati; il metodo ritorna un oggetto *Instances* che contiene il risultato delle operazioni effettuate. Nel punto successivo analizzeremo le modalità con cui vengono definiti questi criteri.

- *RemoveWithValues.java* è la classe principalmente utilizzata nel corso di quest'opera per compiere i differenti tipi di filtraggio messi a disposizione dell'utente. Per utilizzare correttamente questo oggetto è necessario definire una serie di opzioni, che vengono salvate in un apposito array, e successivamente associarle all'oggetto di tipo *RemoveWithValues;* grazie a questo meccanismo si possono comporre le richieste di filtraggio desiderate. Le opzioni di interesse disponibili sono:
    - *-C <num>*: permette di selezionare l'attributo corrispondente all'indice *num* facendo riferimento all'ordine in cui vengono definiti gli attributi all'interno del dataset in ingresso[22].
    - *-S <num>*: permette di selezionare, tra i dati presenti, quelli che hanno come valore dell'attributo definito dall'opzione -C un valore strettamente inferiore a quello indicato dal parametro -S. Questa modalità di filtraggio può essere utilizzata per gli attributi di tipo *numeric.*
    - *-L <index, index2-index4…>:* permette di selezionare, tra i dati presenti, quelli che hanno come valore dell'attributo definito dall'opzione -C il valore corrispondente a quello alla posizione index-esima (eventualmente si può specificare anche più di un

[21] Nei linguaggi di programmazione questa operazione viene detta *override*. Per tale motivo in Java viene riportata l'annotazione *@Override* al di sopra della firma del metodo in questione.
[22] Una particolarità di questa opzione è rappresentata dal fatto che la numerazione degli indici degli attributi presenti nel dataset parte da 1, discostandosi dal più usuale valore 0 usato spesso in informatica per indicare il primo indice di una lista.

valore o anche un intervallo di valori), facendo riferimento, anche in questo caso, all'ordine in cui tali valori vengono definiti all'interno del file arff. Questa modalità di filtraggio può essere utilizzata per gli attributi di tipo nominal.

È inoltre possibile eseguire più operazioni in cascata, avendo la cura di utilizzare come Instances di partenza i dati ottenuti dall'operazione precedente: in questo modo è possibile realizzare algoritmi di filtraggio complessi.

Altre modalità di filtraggio sono messe a disposizione dalla libreria; tuttavia esse esulano dallo scopo di questa tesi, e per tale ragione si è omessa la loro descrizione che si può in ogni caso facilmente recuperare dalla documentazione ufficiale *Weka*[9] presente in rete.

4.7 La dimensione shape

Weka, già nella sua versione originale, era in grado di offrire una buona qualità in termine di visualizzazione dei dati su scatter plot. Il tab *Visualize,* come si è già avuto modo di osservare, permetteva di presentare un grafico su due dimensioni, grazie al quale era già possibile selezionare tre diversi attributi secondo i quali visualizzare i dati del dataset preso in esame; questo significa che, a fronte di due dimensioni dello spazio, lo scatter plot permetteva una rappresentazione dei dati secondo una terza dimensione aggiuntiva. Riassumendo il grafico veniva tracciato utilizzando gli attributi selezionati come grandezze di riferimento per ciascuna delle dimensioni rappresentate a video, ovvero:

- Asse delle ascisse (X)
- Asse delle ordinate (Y)
- Colore del punto tracciato

Tutte e tre le dimensioni citate potevano essere rappresentate da un differente attributo del dataset, indipendentemente dal tipo ad esso associato (per esempio nominal o numeric).

Dal momento che i dataset relativi alle misure di qualità video analizzati nel corso di quest'opera presentano un numero elevato di attributi, e che in generale *Weka* costituisce un software utilizzato per l'analisi di dataset anche molto complessi, si

è ritenuto che potesse essere di particolare interesse la visualizzazione di un grafico che prevedesse la rappresentazione di un'ulteriore dimensione, associabile ad un quarto attributo del dataset. L'implementazione di questo sviluppo ha permesso al software di tracciare il punto relativo al dato in questione non più come un semplice punto, bensì mediante l'utilizzo di una *shape,* ovvero una forma del punto disegnato, diversa a seconda del valore associato a tale attributo. Per la sua natura si può facilmente intuire come tale attributo possa essere scelto solamente tra gli attributi nominal presenti nei dati in ingresso: infatti, essendo la shape rappresentabile solamente in relazione ad un preciso valore di un attributo, non potrà essere utilizzata per quegli attributi che, invece, possono ricoprire un intero range di valori, ossia gli attributi di tipo numeric. Al fine di evitare che un utente selezioni erroneamente un attributo non rappresentabile secondo questa notazione, causando una visualizzazione impredicibile dei punti associati alla grandezza selezionata, la casella di selezione dell'attributo da associare alla shape permette la selezione dei soli attributi di tipo nominal, escludendo a priori la visualizzazione, e di conseguenza l'utilizzo, di attributi di tipo differente.

L'introduzione di questa nuova dimensione, che di fatto non è associata a una dimensione spaziale, ma soltanto qualitativa, all'interno dello scatter plot, ha reso necessaria l'introduzione di una mappatura tra le forme visualizzate e i valori ai quali si riferiscono. Infatti, così come accade per l'attributo associato al colore del punto visualizzato, anche per quello associato alla shape è stata aggiunta una legenda, che mostra l'associazione della forma del punto disegnato con il relativo valore dell'attributo selezionato. Al fine di preservare il maggior spazio possibile a disposizione dello scatter plot*,* questa nuova legenda è stata aggiunta al già presente pannello in cui era visualizzata quella relativa all'attributo legato al colore del punto disegnato; in particolare vengono visualizzati tutti i valori possibili per l'attributo nominal*,* selezionato nell'apposita combobox*,* con accanto la forma del punto ad essi associati, utilizzando il colore nero in modo da mantenerla neutra nei confronti di quella relativa all'attributo colour. In questo modo si è costruito un pannello che permette di individuare con semplicità le due dimensioni non rappresentate sugli assi, che vengono in questo modo combinate tra loro; mediante un grafico in due dimensioni, così, l'utente è in grado di identificare quattro grandezze differenti.

L’introduzione di questa nuova dimensione potrebbe non essere particolarmente evidente nel caso in cui l’utente stia prendendo in esame un dataset che presenta dei valori in ingresso molto densi, vale a dire con dati che rappresentano valori molto simili tra loro, o anche nel caso in cui essi vengano disegnati all’interno di un grafico avente degli estremi molto dilatati sugli assi, concentrando in questo modo la maggior parte dei dati in aree ravvicinate del piano. Questa nuova funzionalità, viceversa, risulta di particolare interesse nel momento in cui i dati analizzati appaiono più rarefatti; questa situazione si verifica principalmente in due situazioni distinte:

- *Natura dei dati selezionati:* la prima situazione, che è anche la più intuitiva, si verifica quando i dati analizzati sono per loro natura più radi; questo è il caso tipico che si verifica quando sugli assi cartesiani vengono rappresentati degli attributi nominal, che per loro conformazione sono graficamente molto distaccati tra di loro, potendo assumere solamente dei valori discreti e, per quanto riguarda i dataset presi in esame nel corso di quest’opera, accettando una quantità di valori predefiniti molto bassa, che supera la decina solo nel caso di pochi attributi.
- *Selezione di un intervallo ristretto:* il secondo caso che si può verificare è quello in cui, a fronte di un ridimensionamento dello scatter plot, i dati visualizzati risultino dispersi; tale comportamento si può osservare nel caso in cui l’utente decida di porre la sua attenzione su un intervallo ristretto del grafico inizialmente disegnato: restringendo i range da rendere graficamente sugli assi, i dati mostrati sul piano saranno di conseguenza di numero ridotto, risultando inevitabilmente più distanti tra di loro.

Entrambe le situazioni prese in esame fanno in modo che le forme dei punti disegnati, in accordo all’attributo selezionato nell’apposita casella, risultino maggiormente evidenti all’utente, che può in questo modo beneficiare a pieno di questa nuova funzionalità.

4.8 Dimensione del punto

In seguito all'introduzione della nuova dimensione della shape*,* ci si è domandati se esso fosse sempre facilmente leggibile da parte dell'utente; si è già analizzato nella sezione precedente come in determinate situazioni esso risulti difficilmente "interpretabile", ma tale difficoltà risiede nella natura degli stessi dati analizzati. È possibile, tuttavia, individuare un ulteriore eventualità in cui l'utente potrebbe avere difficoltà a leggere tali dati; infatti, se si ipotizza, come già fatto più volte nel corso di quest'opera, di mostrare lo scatter plot mediante l'utilizzo di un proiettore, l'utente potrebbe osservare lo stesso da una distanza potenzialmente anche di decine di metri. Nonostante la superficie di visualizzazione notevolmente aumentata rispetto a quella di un comune personal computer, l'eccessiva distanza potrebbe in questo caso rappresentare un problema da affrontare per fare in modo che la forma del punto tracciato sia effettivamente riconoscibile da qualsiasi persona. Per fare ciò si è deciso di fornire all'utilizzatore del software un ulteriore strumento, che renda modificabile, impostando un'apposita grandezza, la dimensione del punto disegnato.

Questa funzionalità risulta particolarmente interessante, inoltre, nel caso in cui il grafico venga ridimensionato in modo da visualizzare un intervallo molto ristretto dei valori presenti sugli assi cartesiani; facendo ciò, può accadere che rimangano pochi punti disegnati sul grafico, risultando di conseguenza meno densi e talvolta addirittura isolati; in questi casi appare conveniente aumentare la dimensione dei punti tracciati sullo scatter plot, in modo da rendere la comprensione dell'informazione trasportata dai punti più immediata.

A livello implementativo, per raggiungere l'obiettivo prefissato si è ricorso all'utilizzo di uno slider[23] nella parte immediatamente superiore al pannello contenente il grafico che, in maniera molto intuitiva, vale a dire grazie allo scorrimento del proprio cursore, permette il ridimensionamento di tutti i punti presenti sul grafico in tempo reale, mantenendo inalterate tutte le altre proprietà ad esso associate. Il valore di default è impostato a un quarto della lunghezza dello slider*;* in questo modo l'utente è in grado sia di aumentare la dimensione dei punti

[23] Lo slider viene rappresentato utilizzando l'oggetto *JSlider* del framework *Java Swing.*

disegnati, per far fronte alla problematica già introdotta all'interno di questa sezione, sia di diminuirla, nel caso in cui, per esempio, i dati visualizzati fossero troppo densi. L'utilizzatore potrebbe ritenere utile, al fine di evitare il ridimensionamento del grafico che ne permetterebbe una migliore visualizzazione in particolar modo delle forme dei punti, diminuire la dimensione dei punti disegnati, in modo da non dover ridimensionare gli estremi degli assi cartesiani, continuando a visualizzare l'intero range di valori. La scelta effettuata per quanto riguarda il valore di default dello slider ha l'obiettivo di non permettere l'eccessivo ridimensionamento dei punti verso il basso, in quanto si rischierebbe di "perdere" l'informazione contenuta nella shape.

A differenza di quanto discusso in relazione alla nuova dimensione rappresentata dalla shape del punto disegnato, si è scelto di dare la possibilità all'utente di modificare la dimensione del punto a proprio piacimento, con il solo fine di migliorare la leggibilità dei dati visualizzati dal grafico. Un'altra strada che si sarebbe potuta intraprendere sarebbe stata quella di associare la dimensione del punto ad un'ulteriore dimensione, arrivando, in questa eventualità, ad un numero di cinque dimensioni visualizzate. Se da un lato questa opzione avrebbe permesso di incrementare ancora una volta la bontà dei dati mostrati, nel corso degli sviluppi si è deciso di non intraprendere questa strada; le ragioni di questa scelta sono principalmente due:

- *Difficoltà di lettura:* la prima motivazione è rappresentata dal modo in cui si sarebbe presentato il grafico in seguito a questa modifica: in primo luogo la rappresentazione di punti più grossi avrebbe reso difficilmente distinguibili, sia per posizione che soprattutto per forma, punti rappresentanti attributi con valori simili; avrebbe inoltre richiesto la contestuale aggiunta della sua legenda, che sarebbe stata difficilmente esprimibile graficamente e che avrebbe richiesto dello spazio aggiuntivo a discapito del pannello utilizzato dallo scatter plot*;*
- *Complessità dei dati:* la seconda ragione risiede nel fatto che si è ritenuto troppo difficoltosa, e di non particolare interesse, la visualizzazione grafica di così tanti attributi sullo stesso piano. L'utente è già in grado di visualizzare quattro grandezze differenti su un piano bidimensionale,

un’ulteriore aggiunta, che sarebbe stata la terza dimensione in aggiunta a quelle rappresentate sugli assi che sono le più intuitive a livello visivo, avrebbe portato anche un grado di confusione aggiuntivo.

L’aggiunta della quinta dimensione è stata demandata al plug-in 3D, che verrà analizzato nel seguito di quest’opera; nel suddetto pannello si è ritenuto accettabile, e anzi, è stato uno degli sviluppi ricercati, avere cinque dimensioni rappresentate graficamente. Questo ha richiesto, così come accaduto per il grafico sul piano cartesiano, l’introduzione della dimensione aggiuntiva legata alla forma del punto tracciato, la shape. Si è deciso, in sintesi, di fornire per ciascuna tipologia di grafico due dimensioni aggiuntive, colore e forma del punto, rispetto a quelle fornite dalle dimensioni geometriche dello scatter plot.

4.9 Plug-in 3D

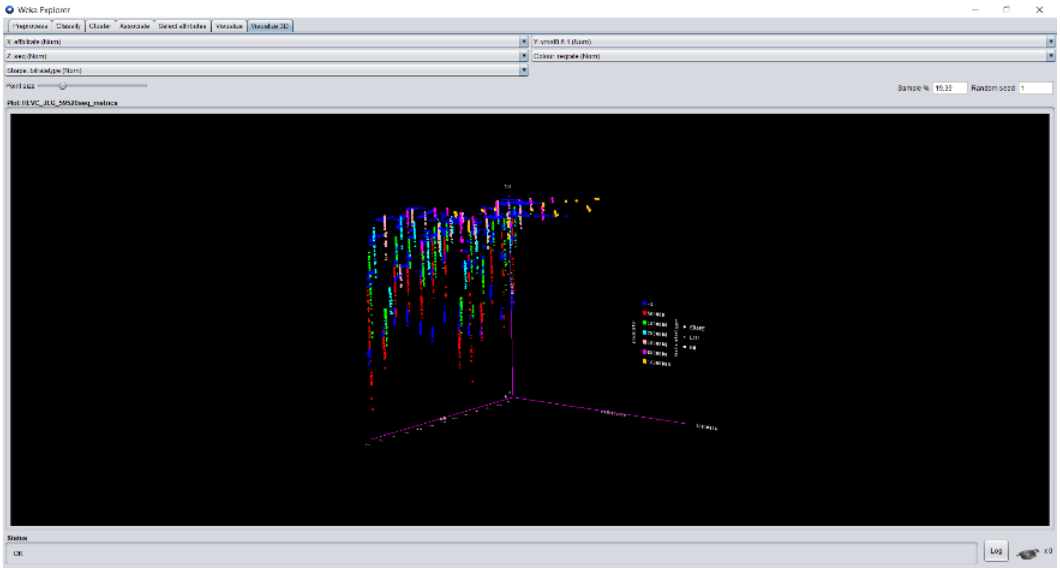

Figura 15 - Pannello Visualize 3D

Nel capitolo precedente ci si è focalizzati sulle modifiche apportate al tab *Visualize,* che rappresenta la parte di visualizzazione bidimensionale dei dati analizzati da *Weka.* Come già accennato nella parte introduttiva di quest'opera, lo scopo della stessa prevede anche l'analisi e, in modo particolare, il miglioramento del *plug-in* con cui è possibile estendere le funzionalità del software permettendo l'introduzione di uno strumento di analisi che utilizza una tipologia di grafico maggiormente evoluta. Il tool in questione permette, infatti, la rappresentazione dei dati contenuti nel dataset analizzato, lo stesso utilizzato per la parte su due dimensioni, mediante l'utilizzo di un grafico disegnato in uno spazio tridimensionale. Questa funzione permette di avere un'ulteriore dimensione rappresentata sullo schermo, che può essere sfruttata per la visualizzazione dei dati; è possibile, inoltre, ricondursi alla rappresentazione su due dimensioni mediante la rotazione del grafico virtuale in modo da allineare il punto di vista dell'utente con una delle dimensioni spaziali. I dati in ingresso sono i medesimi già presentati in precedenza all'interno di questo elaborato e, in particolare, sono condivisi tra le due funzionalità del software; una volta caricato il dataset su *Weka,* esso ne consente la

visualizzazione analitica utilizzando uno dei due tab a disposizione, *Visualize* o *Visualize3D.*

Parallelamente allo sviluppo relativo alla visualizzazione bidimensionale, dunque, ci si è occupati anche del plug-in 3D*,* che ricalca per molti aspetti le linee guida già individuate nella presentazione della parte precedente. Il suddetto plug-in ha da un lato l'obiettivo di mantenere il più possibile invariata la struttura originaria dello stesso, andando a modificare in maniera marginale la disposizione dell'interfaccia già presente, dall'altro quello di introdurre le funzionalità già presentate per la parte di visualizzazione sul piano, in modo da conferire un'omogeneità funzionale all'intero software; appare infatti evidente come il disporre di interfacce simili, a tratti addirittura uguali, all'interno di pannelli differenti, permetta all'utilizzatore un più rapido apprendimento delle potenzialità offerte dal software, oltre ad un più facile utilizzo dello stesso.

A fronte di queste premesse, che hanno anche in questo caso l'obiettivo di migliorare l'efficacia dell'analisi del dataset di misure di qualità video, si è deciso di riproporre la maggior parte delle funzionalità già analizzate, con alcuni piccoli accorgimenti resi indispensabili dalla maggiore complessità della tipologia di rappresentazione grafica. Per questo motivo si ritiene utile ripercorrere le novità introdotte anche per la parte tridimensionale, evidenziandone i punti in comune e le eventuali differenze. In particolare:

- Viene riproposto il menu a tendina al click con il tasto destro del mouse sul pannello contenente il grafico per la visualizzazione delle opzioni a disposizione dell'utente; esso offre le medesime funzionalità già presenti nella rappresentazione sul piano, nel dettaglio:
    - *Modifica del colore dello sfondo:* a differenza di quanto accade nel tab *Visualize,* il colore di default è il nero, che meglio si adatta alla resa tridimensionale del grafico. Viene riutilizzata la stessa libreria già vista nel corso di quest'opera (*JColorChooser),* riproponendo la stessa interfaccia ormai nota all'utente. Tale libreria viene riutilizzata anche da tutti gli altri componenti presenti in questo pannello.

- *Modifica del colore degli assi:* il colore di default si avvicina al fucsia, in modo da enfatizzare il contrasto con lo sfondo nero visualizzato di default.
- *Visualizzazione della griglia:* è una delle caratteristiche che risalta maggiormente a livello visivo. Inizialmente si era pensato di visualizzare la griglia su tutte e sei le facce del cubo individuato dagli assi cartesiani, ma per evitare una rappresentazione che poteva diventare confusionaria soprattutto per intervalli ristretti sugli assi, si è deciso di mostrarla solamente su tre facce, ovvero quelle identificate e comprese tra gli assi cartesiani.
- *Modifica del colore della griglia:* il colore di partenza è il grigio. Questa scelta è stata fatta volutamente, pur non offrendo un elevato contrasto come accade nel caso degli assi, al fine di non appesantire eccessivamente il pannello con troppi colori, che renderebbero il grafico più confuso.
- *Ridimensionamento del grafico:* è il medesimo pannello utilizzato nel grafico sul piano, permette di modificare gli estremi degli assi, il numero di divisioni di ciascun asse e di conseguenza il passo, grandezze che vengono anche in questo caso utilizzate per disegnare la griglia.
- *Filtraggio dei dati:* presenta le stesse funzionalità dell'analogo pannello relativo allo scatter plot bidimensionale, con l'unica differenza caratterizzata dal fatto che l'aggiornamento del grafico avviene mediante l'utilizzo di un apposito pulsante di update posizionato all'interno del pannello principale, e quindi non in tempo reale. Vedremo a breve le motivazioni di questa scelta.

- Anche in questo pannello è stata introdotta la dimensione aggiuntiva della shape. Essa permette di raggiungere il numero di cinque attributi visualizzati graficamente (tre sugli assi cartesiani, uno legato al colore, e uno legato appunto alla shape), offrendo all'utente un elevatissimo livello di dettaglio. La differenza più evidente rispetto al grafico sul piano risiede inevitabilmente nella forma del punto disegnato. Vengono utilizzate delle

forme tridimensionali[24], numericamente più limitate rispetto a quelle in due dimensioni; ciò è dovuto al fatto che il grafico nello spazio è, come vedremo meglio nel seguito, ruotabile e traslabile secondo la volontà dell'utente, e per questo motivo forme di per sé diverse, come potrebbero essere, per esempio, due cilindri orientati secondo due assi diversi, risultano difficilmente distinguibili in seguito ad una rotazione del grafico. L'aggiunta di questa nuova dimensione rende indispensabile l'inserimento contestuale della relativa legenda, che si colloca a fianco di quella già presente per il colore e che verrà descritta nel dettaglio all'interno di questa sezione.

- È possibile, infine, variare la dimensione del punto disegnato nello spazio. Per le motivazioni già discusse relativamente alla parte bidimensionale, si è deciso di non legare questa dimensione ad un attributo specifico dei dati, ma si tratta solamente di una facilitazione visiva che può essere utile all'utilizzatore per far fronte a determinate tipologie di analisi.

L'introduzione della dimensione aggiuntiva della shape, come già analizzato per la parte bidimensionale, ha richiesto l'introduzione di una nuova legenda, con lo scopo di mappare i valori dei dati con le forme rappresentate a livello grafico. Anche in questo caso si è deciso di posizionare questa nuova legenda a fianco di quella già presente per gli attributi legati alla dimensione del colore. La nuova configurazione grafica delle legende prevede:

- *Legenda colour:* è posta, come già nella versione originale del plug-in, nella parte destra dello scatter plot; tuttavia è stata migliorata a livello grafico in modo da rendere più facilmente leggibili i valori, e soprattutto i colori, rappresentati al suo interno. Si è deciso, infatti, di riprendere le linee guida già seguite per lo scatter plot sul piano, che prevedono dunque una rivisitazione della legenda per quanto riguarda gli attributi nominal. Nella versione da cui si è partiti i valori erano rappresentati mediante la visualizzazione del valore in questione utilizzando il colore ad esso associato; questo tipo di visualizzazione rendeva talvolta la lettura difficoltosa, perciò si è deciso di stampare i valori testuali utilizzando

[24] Le forme tridimensionali utilizzati nel pannello di visualizzazione 3D vengono create utilizzando gli oggetti *Sphere*, *Cone, Cylinder* e *Box* contenuti nella libreria *Java3D.*

sempre un colore neutro, a cui viene affiancato un quadrato con il colore corrispondente a tale valore per identificare l'associazione. Si può notare, dunque, come venga riproposta la stessa tipologia di rappresentazione utilizzata nel pannello bidimensionale.

- *Legenda shape:* è stata posizionata a fianco della legenda precedente; anche in questo caso ci si è ispirati al pannello relativo allo scatter plot sul piano, dal momento che la legenda viene costruita visualizzando il nome del valore dell'attributo nominal con a fianco la forma tridimensionale ad esso associata, anch'essa visualizzata mediante l'utilizzo di un colore neutro.

Da sottolineare anche il fatto che entrambe le legende vengono ancorate sulla parte laterale del grafico, diventandone parte integrante; ciò fa in modo che quest'ultime subiscano le medesime trasformazioni geometriche che l'utente richiede di eseguire sul grafico visualizzato.

L'interfaccia fornita all'utente è stata costruita sulla base di quella fornita dal plug-in originale, alla quale sono state aggiunte la combobox relativa alla dimensione della shape e lo slider che permette la modifica della dimensione dei punti visualizzati.

La parte di rappresentazione grafica in tre dimensioni è senza dubbio la più onerosa dal punto di vista computazionale, motivo per il quale è presente, a differenza di quanto osservabile nel tab bidimensionale, il pulsante *Update display,* che permette l'aggiornamento del grafico su richiesta dell'utente. Gli sviluppi effettuati nel corso di quest'opera hanno tenuto conto anche della complessità che avrebbe comportato l'aggiornamento del grafico in real-time*,* in modo analogo a quanto avviene sul piano, e ci si è resi conto che esso sarebbe risultato troppo oneroso, soprattutto ipotizzando di utilizzare *Weka* e il suo plug-in su macchine più modeste e non dotate di una scheda grafica dedicata[25]. Si è dunque scelto di mantenere le modalità di aggiornamento già presenti nel plug in originale in modo da non rendere eccessivamente oneroso il dispendio computazionale, che per alcune operazioni, come per esempio quelle disponibili all'interno del pannello di filtraggio, potrebbe risultare insostenibile.

[25] La macchina utilizzata nel corso degli sviluppi e dei test è un *HP Probook 15"* con processore *Intel i7-8550U, 512GB* di *SSD, 16GB* di *RAM* e scheda grafica dedicata *NVIDIA GeForce 930MX*. Il sistema operativo utilizzato è *Windows 10 Home* a *64 bit.*

Un'altra differenza che può essere evidenziata rispetto al pannello di visualizzazione bidimensionale, in relazione al peso in termini di prestazioni di questo plug-in, risiede, come già accennato, nella possibilità di maneggiare il grafico secondo tutte le sue dimensioni geometriche. L'utilizzo di questa funzionalità, che risulta necessaria per la natura della visualizzazione tridimensionale e per gli scopi dichiarati in precedenza, presenta allo stesso tempo aspetti positivi e negativi. I primi si possono riassumere nei punti seguenti:

- *Adattabilità del grafico:* il grafico può essere in qualsiasi momento zoomato, ruotato o traslato all'interno dell'ambiente virtuale costruito all'interno del pannello di visualizzazione. Ciò permette allo stesso di adattarsi alle diverse situazioni di lettura dei dati, che talvolta potrebbero risultare difficoltose a causa della distanza dell'utente o della dimensione utilizzata, per esempio, per stampare le label degli assi, consentendo anche di focalizzarsi su una parte specifica degli intervalli selezionati su questi ultimi.
- *Visualizzazione esaustiva dei dati:* questa è sicuramente la caratteristica più importante conferita dalla maneggevolezza del grafico. In una rappresentazione tridimensionale, infatti, accade spesso che alcuni punti siano visivamente sovrapposti a causa della terza dimensione che per ovvie ragioni non può essere rappresentata in modo concreto su una superficie piatta come è quella dello schermo. La possibilità di ruotare il grafico permette all'utente di analizzare lo scatter plot da una qualsiasi angolazione visiva possibile, avendo una visuale dello stesso a 360 gradi.
- *Potenza del software:* l'implementazione di questa feature rende *Weka* uno strumento ancora più potente, sia dal punto di vista grafico, grazie al quale l'utente può analizzare visivamente il dataset in modo più accurato e puntuale, sia dal punto di vista computazionale, dal momento che l'elaborazione grafica tridimensionale è un'operazione estremamente avanzata.

A fronte dei sopracitati punti di forza introdotti, bisogna segnalare anche alcune note negative introdotte dal plug-in in questione:

- *Onerosità computazionale:* questo è indubbiamente il fattore negativo più evidente riguardante la visualizzazione 3D; infatti, l'utilizzo di tale strumento è estremamente oneroso per il calcolatore che utilizza tale software, a cui è richiesto già un notevole sforzo grafico per la rappresentazione nello spazio. Quest'ultimo viene notevolmente accentuato nel momento in cui si richiede di ruotare o traslare il grafico rappresentato. Per compiere queste azioni è consigliabile utilizzare il software su una macchina dotata di scheda grafica dedicata, senza la quale si avrebbe un considerevole calo delle prestazioni con un conseguente peggioramento della fluidità dell'applicazione.
- *Apprendimento di utilizzo:* per eseguire queste operazioni grafiche è necessario utilizzare i comandi di un dispositivo di input quali un mouse o un touchpad. Soprattutto durante i primi utilizzi può risultare poco intuitivo il funzionamento di tali azioni, richiedendo all'utente un periodo di tempo, seppur non particolarmente lungo, al fine di apprendere a pieno le funzionalità offerte da questo strumento grafico.

A riguardo di quest'ultimo aspetto, si ritiene necessario fornire una panoramica sui comandi disponibili per effettuare le operazioni sul grafico tridimensionale, che come detto in precedenza, vanno a coprire le trasformazioni geometriche[26] di una rappresentazione spaziale:

- *Traslazione:* è possibile traslare il grafico mediante il click del pulsante destro del mouse con contestuale movimento del cursore nella direzione desiderata.
- *Rotazione:* è possibile ruotare il grafico mediante il click del pulsante sinistro del mouse con contestuale movimento del cursore nella direzione di rotazione desiderata.
- *Zoom:* è possibile ingrandire o rimpicciolire il grafico mediante l'utilizzo della rotella di scorrimento del mouse (in alternativa, se si dispone di un touchpad è sufficiente utilizzare due dita su di esso).

[26] Con il termine *trasformazione geometrica* si indica in questo contesto un'alterazione della figura nello spazio tridimensionale che non ne varia le proporzioni.

Bisogna fornire, infine, un ultimo cenno riguardo all'architettura che si è utilizzata per strutturare il codice del plug-in*:* nonostante la simmetria rispetto alla parte bidimensionale pensata per quest'ultimo, a causa della natura del software (essendo un plug-in si è cercato di renderlo indipendente dal punto di vista architetturale dalle altre classi di Weka) e delle diverse librerie utilizzate per la presentazione grafica dei dati, si è potuto riutilizzare solamente in parte il codice già sviluppato in precedenza, adattandolo alle necessità richieste da questo tipo di visualizzazione e mantenendo in parte separate le due distinte implementazioni.

5 Risultati ottenuti

Per apprezzare al meglio i risultati ottenuti grazie agli sviluppi effettuati, che hanno reso possibile una migliore analisi dei dataset in possesso, è possibile fare un paragone tra le due versioni[27] del software *Weka*; nel seguito verranno messi a confronto gli screenshots effettuati a partire dal software originale e dalla versione sviluppata nel corso di quest'opera.

5.1 Scatter plot 2D

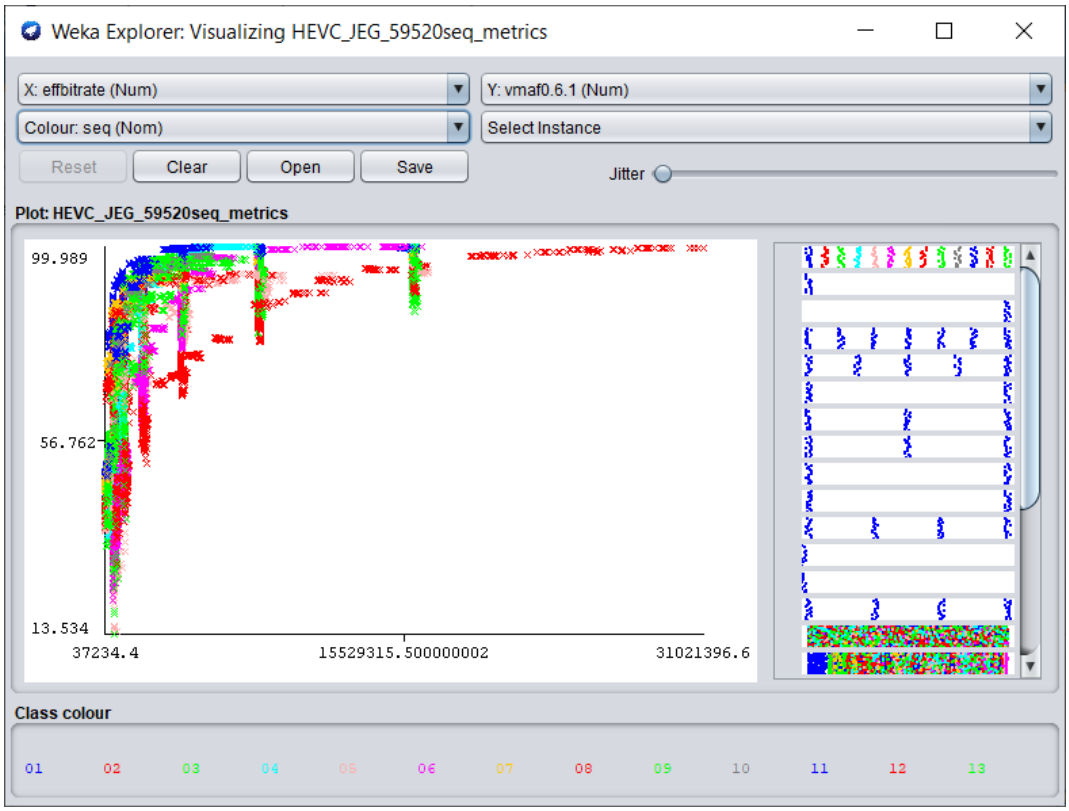

Figura 16 - Visualize tab: versione originale

[27] La versione di partenza è rappresentata dalla versione 3.8 di Weka, mentre la versione sviluppata è rappresentata dalla versione presente sul repository presente su Git all'indirizzo https://gitlab.com/alessandro.catto/Weka.git nel branch *master* al momento della stesura di quest'opera (Dicembre 2019).

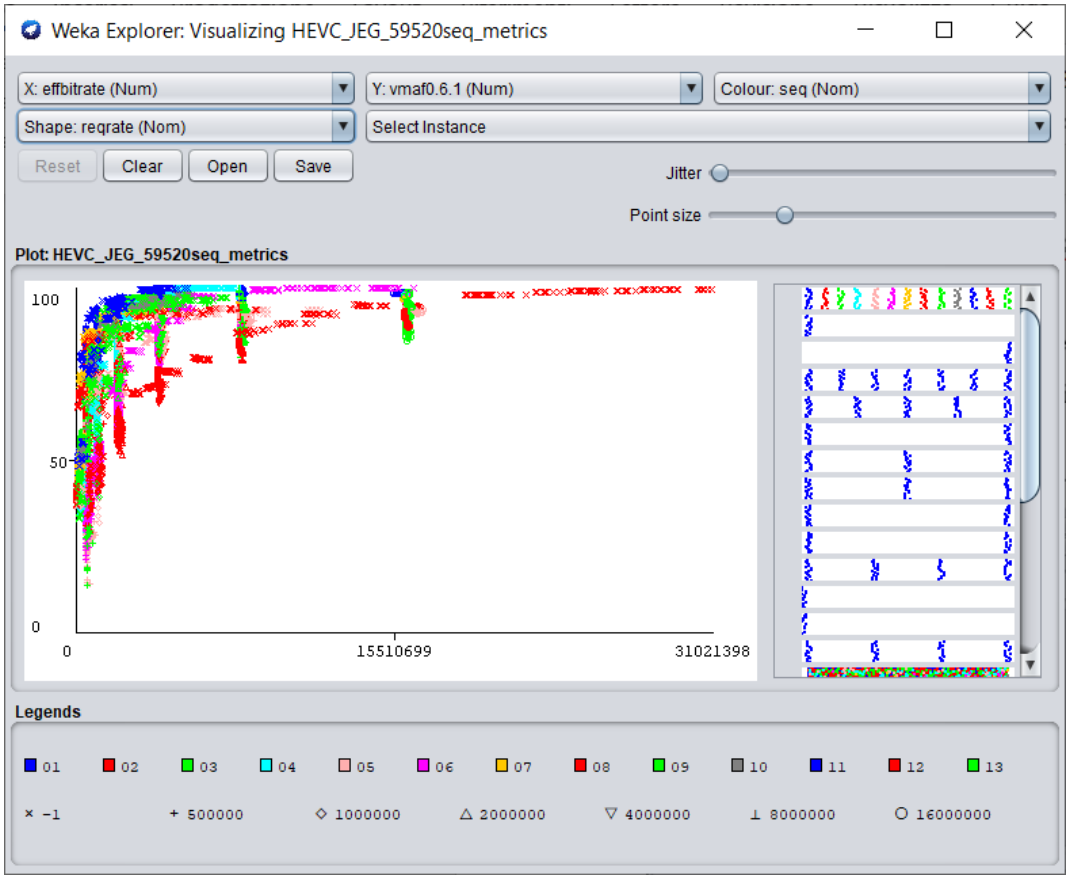

Figura 17 - Visualize tab: versione sviluppata

In questo primo confronto possiamo apprezzare due differenze principali; in primo luogo nella parte alta della finestra troviamo gli strumenti destinati all'utilizzo di parte delle nuove funzionalità introdotte, che hanno richiesto la ridistribuzione dello spazio a disposizione. Si è scelto, in particolare, di disporre i vari elementi su tre colonne, dal momento che il contenuto delle stesse può essere comodamente collocato in uno spazio più ristretto, a vantaggio dell'inserimento della nuova combobox relativa alla dimensione della shape. Nella parte bassa, invece, si può notare l'introduzione della legenda relativa alla quarta dimensione introdotta, ovvero quella della forma del punto disegnato sul grafico; essa viene collocata all'interno del medesimo pannello destinato alla legenda dell'attributo colour, in modo da limitare la richiesta di spazio destinato ad elementi differenti dal grafico vero e proprio. Sempre in relazione alla sopra citata legenda, si può notare come sia nettamente migliorata la leggibilità di quella relativa ai colori utilizzati nel grafico rispetto alla versione precedente, la quale risultava, soprattutto nel caso in cui la

schermata in questione fosse stata proiettata, di non facile lettura a causa dello scarso contrasto tra il colore del testo e il colore di sfondo. Un altro aspetto che vale la pena sottolineare è quello riguardante i valori visualizzati sugli assi: rispetto alla versione originale, la nuova esegue delle apposite approssimazioni che permettono di evitare la visualizzazione di valori con molte cifre decimali in casi, come quello mostrato nelle figure 16 e 17, che non lo necessitano.

5.2 Scatter plot 3D

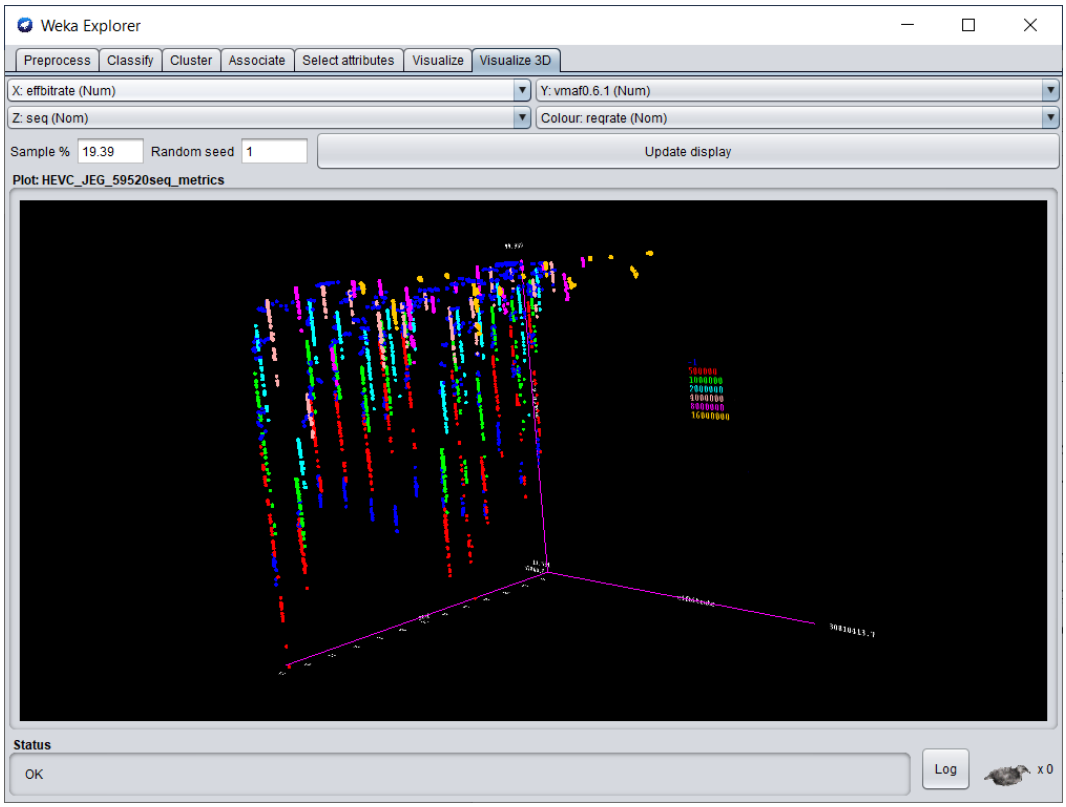

Figura 18 - Visualize 3D tab: versione originale

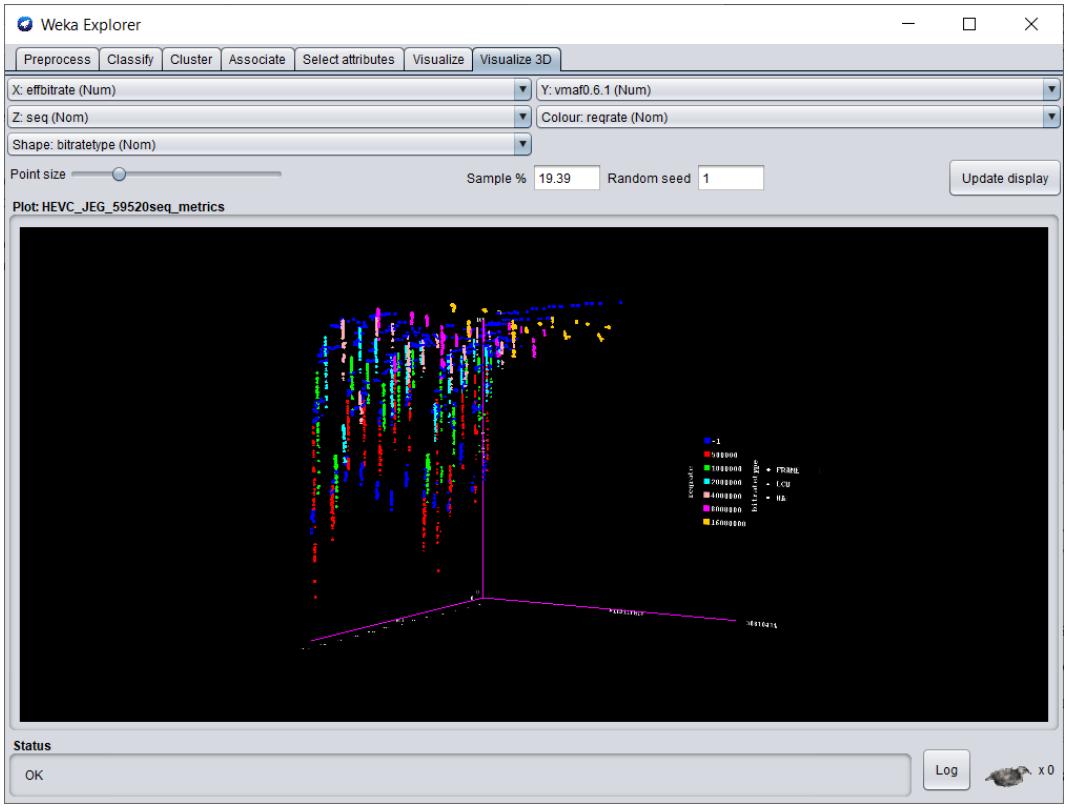

Figura 19 - Visualiaze 3D tab: versione sviluppata

In questa seconda coppia di screenshots possiamo mettere a confronto la visualizzazione del plug-in 3D. In primo luogo, come già evidenziato per il grafico in due dimensioni, è stata modificata l'interfaccia contenente i comandi di base di questo tab, in modo da aggiungere, anche per il plug-in, un'ulteriore dimensione rispetto a quelle già presenti. In questo caso, però, si è preferito mantenere il layout su due colonne, che meglio si adatta alle esigenze di questo pannello. La seconda differenza che l'utilizzatore può riscontrare con facilità è il rinnovo grafico della legenda; nella versione aggiornata essa risulta più ordinata grazie ad un uso più accorto dei riferimenti ai colori utilizzati, come già detto anche per la parte bidimensionale; inoltre essa contiene le informazioni del nuovo attributo aggiunto, ovvero quelle relative alla forma del punto disegnato, all'interno dello spazio dedicato allo scatter plot.

6 Conclusioni

L'obiettivo del lavoro svolto è stato quello di realizzare uno strumento che permettesse la rappresentazione visiva dei dati ottenuti dalle misure, oggettive e soggettive, di qualità video, in modo da evidenziare in modo chiaro le relazioni tra i modelli utilizzati. Per fare ciò, partendo da un software già disponibile in rete, si sono effettuati alcuni sviluppi al fine di eliminare alcune imperfezioni a livello di user experience, e di estendere alcune funzionalità del software, in modo da incrementare in modo consistente la personalizzazione di uno strumento già molto potente. La disponibilità del codice sorgente, essendo *Weka* un progetto open source, e della relativa documentazione, unitamente all'utilizzo di un linguaggio di programmazione largamente conosciuto e documentato quale *Java*, ha permesso un'integrazione relativamente agevole con la versione originale del software. Lo svolgimento di questo lavoro di tesi, a seguito dell'analisi dei dati presenti nei dataset a disposizione sulle misure di qualità video, si è articolato in diverse fasi: una prima fase, resa obbligatoria dalla complessità logica ed architetturale di *Weka*, è stata volta alla comprensione dell'architettura del software originale, essendo esso stesso molto corposo in termini di righe di codice; una volta terminato l'approccio al nuovo software, ci si è dedicati allo sviluppo delle funzionalità volte a migliorare alcuni dettagli soprattutto a livello grafico, per poi passare all'implementazione delle varie parti che sono state introdotte all'interno di *Weka* nel corso di quest'opera; le prime modifiche effettuate in questa direzione sono state quelle relative all'introduzione del menu a comparsa sul pannello di visualizzazione bidimensionale, che racchiude al suo interno buona parte delle novità introdotte. Una menzione particolare va fatta in relazione allo sviluppo del pannello di filtraggio, uno dei componenti più complessi e analiticamente potenti che si ha avuto modo di introdurre, e per cui ci si è dovuti documentare più nel dettaglio al fine di sfruttare a pieno i metodi già presenti nelle librerie del software originale.
Ci si è occupati, in seguito, della parte di visualizzazione tridimensionale, dove si è ripercorso quanto fatto sul piano in precedenza; tali operazioni sono risultate da un lato più semplici, in quanto hanno richiesto sviluppi molto simili a quelli precedenti,

ma dall'altro impegnative dovendosi approcciare alla programmazione per la realizzazione di un ambiente virtuale in tre dimensioni.

La fase conclusiva ha permesso di effettuare, in seguito a test effettuati mediante l'utilizzo empirico del software, il bug fixing di quanto sviluppato e la messa a punto di alcuni dettagli che si sono resi evidenti solo nel corso del testing a livello di interfaccia grafica.

Grazie al lavoro svolto si sono ottenuti, a parere dell'autore, risultati soddisfacenti in termini di usabilità, se confrontati con il software di partenza, che permettono un'analisi più accurata e, in particolar modo, più immediata dei dataset oggetto di questa tesi.

Gli sviluppi futuri possibili potrebbero riguardare il miglioramento del design delle interfacce grafiche, dal momento che esse risultano un po' lontane da quanto un'applicazione sviluppata a partire dagli ultimi anni è in grado di offrire, e il refactoring del codice già esistente. L'intero software, infatti, seppur ben strutturato mediante l'utilizzo di numerosi package, presenta classi di dimensioni notevoli, che superano spesso le migliaia di righe; la leggibilità di tali classi talvolta risulta molto impegnativa, per questo motivo *Weka* potrebbe trarre beneficio nel caso in cui venisse effettuato un refactoring del codice, che richiederebbe un'attenta analisi di quanto già presente in modo da organizzarlo al meglio, permettendo auspicabilmente un contestuale miglioramento complessivo delle prestazioni.

Bibliografia

[1] *Joint Effort Group,* http://vqegjeg.intec.ugent.be, consultato il 3 Novembre 2019.

[2] *Rapidminer,* https://rapidminer.com, consultato il 12 Novembre 2019.

[3] *Google Data Studio,* https://datastudio.google.com, consultato il 13 Novembre 2019.

[4] *Rawgraphs,* https://rawgraphs.io, consultato il 13 Novembre 2019.

[5] *Weka,* https://www.cs.waikato.ac.nz, consultato il 3 Agosto 2018.

[6] *Progetto Weka,* https://github.com/Waikato, consultato il 4 Aprile 2018.

[7] *Oracle Java,* https://docs.oracle.com, consultato il 10 Maggio 2018.

[8] *Pentaho,* https://wiki.pentaho.com, consultato il 3 Agosto 2018.

[9] *Documentazione Weka,* http://weka.sourceforge.net, consultato il 10 Agosto 2018.

[10] *Stackoverlow,* https://stackoverflow.com, consultato il 3 Agosto 2018.

[11] *Wikipedia,* https://en.wikipedia.org, consultato il 18 Settembre 2018.

[12] *Google Developers,* https://developers.google.com, consultato il 4 Novembre 2019.

[13] *Machine learning mastery,* https://machinelearningmastery.com, consultato il 6 Novembre 2019.