INTER-JUVENTUS: COME GLI UTENTI TWITTER INTERAGISCONO TRA LORO E NELLA PIATTAFORMA DURANTE IL MATCH

Lavoro a cura di Alessandro Fossati (819499) e Luca Pretini (864014)
Università degli studi Milano Bicocca, cdLM Data Science, SocialMediaAnalytics

Abstract Inter-Juventus, conosciuta come "Il Derby d'Italia", rappresenta una classica del calcio italiano, seguita con passione e chiacchierata da milioni di tifosi e non tifosi, anche per il fatto che spesso risulta essere una partita decisiva per l'assegnazione del titolo di Campione d'Italia. La partita, giocata il 17 gennaio 2021, ha visto l'Inter vincere per 2 a 0, con una prestazione esaltante, a discapito di una Juventus un po' spenta.
La partita ha offerto diversi spunti di analisi. All'interno di questo report si andrà dunque ad analizzare come gli utenti interagiscono tra loro attraverso la rete dei retweet e come si esprimono tramite la piattaforma Twitter, in particolare in relazione ad alcuni eventi che hanno segnato il corso della partita, ma anche del pre-partita e del post-partita.

Introduzione

Analizzare come gli utenti utilizzano Twitter durante la partita può essere utile per moltissimi scopi. Le domande di ricerca alle quali il report proverà a fornire una risposta sono le seguenti:
come è strutturata e quali sono le caratteristiche della rete degli utenti che effettuano retweet? Come gli utenti che twittano in lingua italiana reagiscono a particolari eventi all'interno della partita e come mutano opinione sull'evento analizzato? Come la presenza dei retweet può distorcere gli score di *sentiment analysis*?
La risposta a queste domande verrà fornita all'interno di questo report, la cui struttura prevede una prima sezione dedicata alle modalità inerenti alla raccolta dei dati Twitter, seguita da una sezione relativa alla analisi della rete sociale degli utenti che retwittano i post di altri utenti. All'interno di questa seconda sezione, verranno analizzate nel dettaglio le caratteristiche della rete sociale, tramite alcune metriche ed alcune visualizzazioni, e verranno effettuate operazioni di *community detection*, al fine di provare ad identificare gli utenti pro-Inter e pro-Juve all'interno della rete. Successivamente vi sarà una terza sezione dedicata alla *sentiment analysis* dei tweet, la

quale fornirà risposte alla maggior parte delle domande di ricerca, passando attraverso *sentiment score* ed interessanti visualizzazioni dei dati.
Si precisa che per questo lavoro si è deciso di trattare esclusivamente **tweet in lingua italiana**, pertanto sia la *social network analysis* che la *sentiment analysis* saranno effettuate solo con tweet pubblicati in italiano.

1. Raccolta dati

La raccolta dati è stata effettuata tramite **Tweepy**, libreria scritta in Python che permette di accedere alle API di *Twitter Developers*. La raccolta è stata effettuata in batch, nei giorni successivi all'evento.
Tweepy permette di interagire con l'endpoint *GET search/tweets*[1] fornito da Twitter, per ottenere tweet conformi ad una certa ricerca. Sono stati raccolti i tweet pubblicati il 17 gennaio in cui erano presenti gli hashtag relativi all'evento. Gli hashtag più usati sono stati verificati sia manualmente navigando nella piattaforma Twitter che tramite strumenti che indicano i trends estraendoli dalle API di Twitter.[2]
Le API *search/tweets* permettono di ottenere i tweet in oggetti JSON contenenti tutti i dati relativi ad esso, sono stati quindi selezionati a priori i campi di interesse relativi al lavoro che si va a svolgere successivamente, in particolare:

- *created_at*: indica quando il tweet e' stato pubblicato, riportando l'orario UTC
- *id* e *id_str*: entrambe riportano l'ID, un identificativo univoco del tweet raccolto, la prima in formato *int*, la seconda in *str*. Per evitare possibili casi di incompatibilità con una delle due variabili sono state raccolte entrambe
- *text*: una stringa contenente il testo del tweet
- *retweeted_status*: dato centrale ai fini del lavoro, presente solo se un tweet è un retweet, nel caso di tweet "originali" questo campo non è presente, nella raccolta si controlla infatti se è presente tramite un *try/except*, se non lo è, viene immagazzinata nel campo la stringa "None". Se l'oggetto raccolto è un retweet, presenterà in questo campo una rappresentazione JSON del retweet originale, permettendo di risalire a dati di interesse come ID del tweet originale o *username* dell'autore.
- *tweet.user.id_str, tweet.user.screen_name, tweet.user.location*: riportano informazioni relative all'autore del tweet, in particolare ID dell'utente, nome utente (visualizzabile nella piattaforma come @*username*), location dell'utente e lingua.
- *tweet.lang*: indica la lingua del tweet di riferimento
- *tweet.entities['hashtags'], tweet.entities['user_mentions']*: due liste che riportano

1. Twitter API search, howpublished = https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets, note = Accessed: 2021-02-01.
2. Daily trends twitter 17 gennaio, ore 23, howpublished = https://getdaytrends.com/it/italy/2021-01-17/23/, note = Accessed: 2021-02-01.

hashtag e menzioni contenute nel tweet.

La raccolta effettuata è salvata in un DataFrame Pandas. Si consideri che tramite Tweepy sono stati cercati tweet tramite data e non tramite un orario specifico. Si è stabilito che la fascia oraria di interesse per il progetto va dalle due ore antecedenti alla partita alle due successive, pertanto il DataFrame è stato filtrato seguendo questi orari. Si effettua un filtro anche sulla lingua dei tweet, dato che si decide di lavorare sulla lingua italiana. É interessante notare come questi siano più della meta del totale raccolto.
Il subset ottenuto verrà quindi esportato in formato *CSV*, si ottiene un file dalla dimensione di 160 MB che sarà il dataset di riferimento per i lavori di *Social Network Analysis* e *Sentiment Analyisis* effettuati successivamente.

2. Social Network Analysis

Questa sezione, come anticipato, è dedicata alla analisi della rete dei retweet in lingua italiana effettuati tra le 20:45, fischio d'inizio della partita, e le 23:00, fascia oraria che permette di comprendere anche i retweet del primo post-partita (22:40 – 23:00), il momento più caldo della serata calcistica, in cui gli appassionati, delusi od euforici per il risultato, condividono su Twitter le loro emozioni.
La sezione sarà suddivisa in due parti: la prima relativa alla *community detection*, la seconda relativa all'analisi della rete attraverso alcune metriche specifiche.

2.1 Community Detection

Grazie alle funzionalità fornite dalla libreria *Networkx* di Python, si implementa un grafo diretto rappresentante la rete dei retweet. I vertici del grafo non comprendono solamente gli utenti che hanno effettuato almeno un retweet, bensì, anche gli utenti che hanno effettuato i tweet originari, che sono stati retwittati.
Lo *user_id* di questi utenti viene ricavato dal campo **retweeted_status**,tramite un espressione regolare. Essendoci la possibilità che un utente retwitti più volte un secondo utente nel corso della serata, si è deciso di assegnare ad ogni edge del grafo un peso pari al numero di interazioni direzionate che vi sono state nell'arco della serata tra i due utenti considerati.
Come si evince dalle statistiche presentate nella Tabella 2.1.1 relative alla rete, essa presenta una quantità di vertici ed edges elevata. Questa tipologia di rete conduce le analisi di community detection verso un approccio basato sull'algoritmo di clustering delle K-medie, il cui funzionamento è riportato nella Figura 2.1.2.

L'obiettivo è quello di identificare le comunità degli utenti **Pro-Inter**, **Pro-Juve**, e **Neutri**. Questi ultimi rappresentano tutti gli appassionati di calcio non juventini od interisti che entrano nella rete dei retweet. A questo pro viene generata la **matrice di adiacenza** pesata del grafo diretto, ossia una matrice *nxn* (dove *n* è il numero di nodi della rete) che indica il numero di interazioni tra ogni coppia di nodi considerata, e mediante l'algoritmo *K-means*, al quale viene richiesto di ricercare le tre community

type	nodes	edges
DiGraph	4747	5887

Tabella 2.1.1

Algorithm 1 Basic K-means Algorithm.

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

Fig 2.1.2

precedentemente elencate, si cerca di ottenere un risultato soddisfacente per proseguire l'analisi.
Nella Figura 2.1.3 viene riportato il grafo relativo alla rete, in cui le tre community sono rappresentate in nero, rosso o blu. I nodi colorati di verde rappresentano gli utenti che non hanno effettuato retweet, ma che tuttavia vengono retwittati, alcuni di loro anche un elevato numero di volte. Questi ultimi sono stati categorizzati come Influencer, dal momento che, come si evincerà dalle metriche presentate nella sezione 2.2, sono coloro che hanno ricevuto il maggior numeri di retweet.
Per comprendere meglio il grafo, sono state effettuate alcune analisi empiriche, in cui si evidenzia come coloro che non hanno effettuato retweet, ma che sono stati solamente retwittati almeno una volta, si posizionano all'esterno della rete. Coloro che si trovano al centro della rete sono gli utenti che hanno effettuato più di un retweet e che hanno retwittato più influencer, ad esempio coloro che hanno retwittato il post di Inter e SerieA al fischio finale del match. L' *out_degree* dei nodi dunque aumenta spostandosi verso il centro della rete, dove con *out_degree* si intende il numero di archi uscenti da un determinato vertice del grafo. Nella Tabella 2.1.4 vengono conteggiati gli elementi appartenenti a ciascuna community secondo l'algoritmo *K-means*[3]:

comm_uno	comm_due	comm_tre
3452	7	1288

Tabella 2.1.4

3. K Mean Clustering algorithm, howpublished = https://www.learndatasci.com/tutorials/k-means-clustering-algorithms-python-intro/, note = Accessed: 2021-02-01.

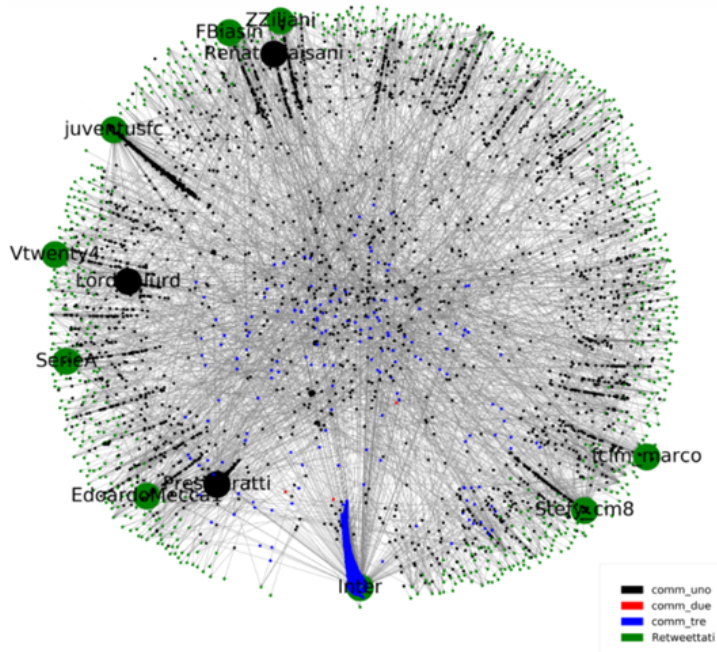


Fig 2.1.3

Si evince uno sbilanciamento numerico a favore della community numero uno, e a sfavore della community numero due. Tuttavia, l'algoritmo fornisce un importante indizio tramite la community numero tre, la quale potrebbe essere associata alla comunità Pro-Inter.
Basandosi su quanto appena appreso, al fine di migliorare l'analisi relativa alla *community detection*, si decide di creare il campo categoriale denominato **tifo** al dataframe di partenza, in modo da etichettare un retweet come proveniente da un utente Pro-Inter, Pro-Juve oppure Neutro.
Questa operazione è possibile andando a ricercare nel testo del retweet determinate parole, emoticons e determinati hashtag che possano identificare un tweet originario come proveniente da una delle tre fazioni in gioco.
Esempi di questi filtri possono essere *#FORZAINTER* per gli utenti Pro-Inter, oppure *#FORZAJUVE* per gli utenti Pro-Juve. La selezione della categoria Neutro avviene qualora non si identificasse un tweet né come Pro-Inter, né come Pro-Juve.
Nonostante i limiti che può comportare l'adozione di un metodo semplice come questo, i risultati sono buoni, anche migliori di quelli ottenuti con il *K-means* a livello grafico, come mostra l'Immagine 2.1.5.

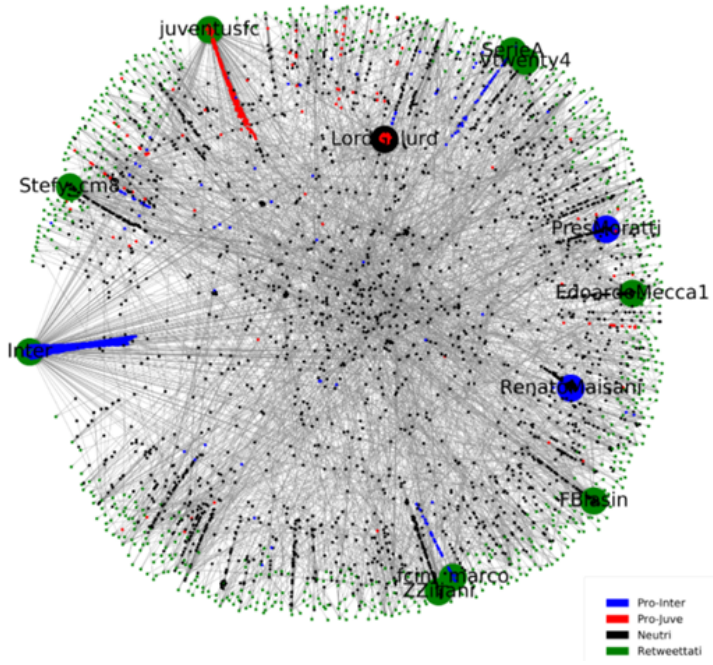


Fig 2.1.5

Con questo approccio la community Pro-Inter resta simile a quella dell'approccio K-means, mentre la community Pro-Juve mostra dei risultati ben più soddisfacenti, come si evince dall'Immagine 2.1.5. Anche per questo secondo approccio si riporta la Tabella 2.1.6, la quale presenta la ripartizione nelle communities a livello numerico, più equilibrata rispetto alla ripartizione di *K-means*:

Neutri	Pro-Juve	Pro-inter
2282	418	1234

Tabella 2.1.6

Nei grafici soprastanti i nodi relativi agli *Influencer* vengono ingranditi, per permetterne una migliore visibilità. Ma come si è determinato quale utente venga categorizzato come un Influencer?
Dalla Figura 2.1.7 si evince come ben pochi utenti abbiano ottenuto un numero maggiore di 60 retweet nell'arco della serata calcistica. Si decide dunque di adottare come valore soglia un numero maggiore di 60 retweet ottenuti per identificare un utente come Influencer:

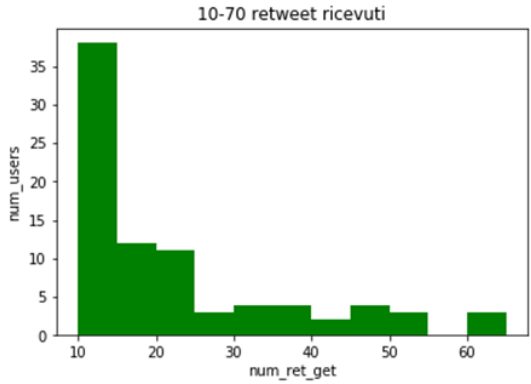


Fig 2.1.7

2.2 Metriche per la Social Network Analysis

In questa seconda parte della seconda sezione verranno presentate alcune metriche che permetteranno di esplorare in maniera più approfondita la rete dei retweet precedentemente presentata. Nella Tabella 2.2.1 e Tabella 2.2.2 vengono presentate una serie di statistiche rilevanti inerenti alla rete:

avg_indeg	avg_outdeg	assortativity	density
1.24	1.24	-0.3	0.0003

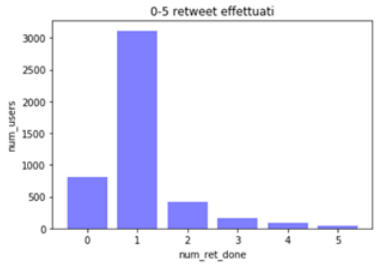
Tabella 2.2.1

node_connectivity	edge_connectivity	overall_reciprocity
0	0	0.002

Tabella 2.2.2

Dai valori di connettività si evince che si è in presenza di un grafo disconnesso. Infatti, le misure di connettività indicano il numero di elementi (nodi o edges) da eliminare dal grafo per renderlo disconnesso. Il valore di reciprocità del grafo è anch'esso molto vicino a zero, il che indica che gli utenti tendono a retwittarsi molto poco a vicenda. Il grado medio di entrata e di uscita dei nodi del grafo risulta essere uguale e di valore poco superiore ad 1, il che indica che la somma dei gradi dei nodi è molto simile al numero di nodi presenti nel grafo, e che quindi la maggior parte dei nodi abbia grado di entrata (*in_degree*) minore o uguale a 1 e grado di uscita (*out-degree*) pari ad 1. Tutte queste statistiche assumono senso di fronte alle

visualizzazioni delle Figure 2.2.3 e 2.2.4, le quali evidenziano il numero di utenti che effettuano (figura di sinistra) e ricevono (figura di destra) rispettivamente da 0 a 5 retweet all'interno della rete:



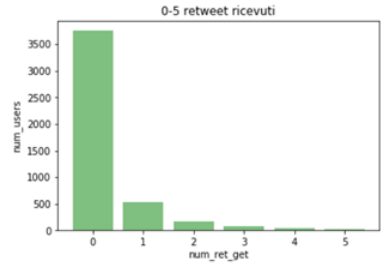
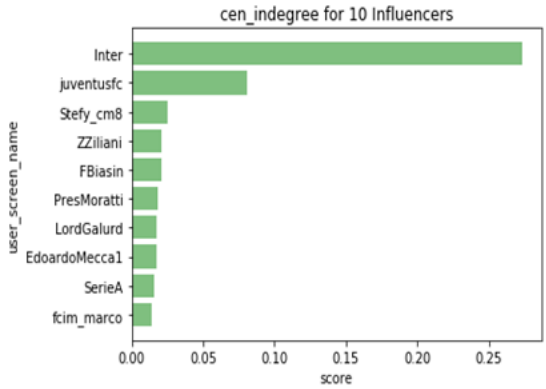


Fig 2.2.3 e Fig 2.2.4

Si evince chiaramente come la maggior parte degli utenti effettui un solo retweet nell'arco della serata, senza riceverne nemmeno uno. Il valore di densità della rete risulta essere molto vicino a zero di conseguenza, ed il valore di *assortativity* (omofilia della rete) è negativo (*disassortativity*) dal momento che gli *Influencer* non si legano tra loro all'interno della rete, bensì sono collegati a nodi di grado decisamente inferiore. In seguito all'analisi di queste statistiche si può affermare che la rete dei retweet analizzata in questa sezione sia una rete di tipo *ego-centrico*, dal momento che vi è la presenza di pochi *Influencer* che vengono retwittati da nodi di basso grado (*preferential attachment*).
Si passa ora ad analizzare le misure di centralità all'interno della rete. Nella Figura 2.2.5 si evidenzia il valore di *centrality degree* in entrata per gli *Influencer* della rete, ossia il valore indicante quanto un utente è stato retwittato. Il valore assoluto di questa misura (riportato in tabella) indica il numero di volte che l'account di riferimento è stato retwittato da altri utenti, mentre il valore normalizzato di centralità (riportato nel grafico) indica il valore assoluto calcolato precedentemente rapportato al numero di nodi meno uno (*n-1*) facenti parte della rete.

Viene anche riportata una tabella all'interno della quale viene indicato il numero di retweet ricevuti da ognuno degli influencer:

Si evince dunque che sia l'account dell'Inter quello più retwittato nell'arco della serata, come era lecito aspettarsi, in seguito alla vittoria, nonostante sia l'account della Juventus quello con il maggior numeri di follower tra gli influencer. Spiccano tra gli *Influencer* alcuni account di tifosi interisti (PresMoratti e FBiasin) e juventini (Stefy_cm8 e LordGalurd).
Oltre ai valori di *degree_centrality*, si presentano anche i valori di *closeness_centralit* dei primi 5 *Influencer* (Tabella 2.2.6), i quali logicamente non si discosteranno troppo dai valori di centralità di grado in entrata. Questa misura di centralità permette di misurare la vicinanza di un nodo a tutti gli altri nodi della rete e si ottiene computando



user_screen_name	node_indegree
Inter	386
juventusfc	121
Stefy_cm8	101
ZZiliani	99
FBiasin	84
PresMoratti	80
LordGalurd	79
EdoardoMecca1	77
SerieA	65
fcim_marco	62

Fig 2.2.5

il reciproco della somma delle distanze di un nodo da tutti gli altri nodi del grafo. I valori riportati in tabella sono normalizzati moltiplicando il valore di centrality per il numero di nodi meno uno (n-1).

user_screen_name	cen_close
Inter	0.0273
juventusfc	0.084
Stefy_cm8	0.024
EdoardoMecca1	0.023
ZZiliani	0.022

Tabella 2.2.6

Si valuta ora la *betweenneess_centrality* per i nodi della rete[4]. Questa misura di centralità misura l'importanza di un nodo nelle comunicazioni con gli altri nodi, e quindi quanto il nodo considerato possa essere un nodo di passaggio.

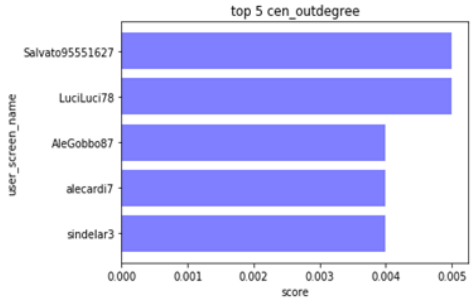
Come ci si aspetta e come si evince dalla Tabella 2.2.7, non vi sono nodi di passaggio rilevanti in questa rete, dal momento che l'informazione fluisce in maniera diretta tra *Influencer* e nodi di basso grado, senza passare attraverso nodi intermedi. Infatti, l'unico *Influencer* presente in questa tabella risulta essere LordGalurd:

4. Slide del corso, howpublished = https://elearning.unimib.it/course/view.php?id=31236, note = Accessed: 2021-02-01.

user_screen_name	cen_bet
juvemyheart	0.000024
GiAdUzZoLa90	0.000016
LordGalurd	0.000007
_grazy87	0.000008
Upupa234	0.000004

Tabella 2.2.7

Si conclude la sezione relativa alla SNA presentando un'analisi che si discosta dagli influencer. In particolare, si decide di andare alla ricerca degli utenti più attivi dal punto di vista del numero di retweet effettuati nel lasso di tempo indicato. Ci si aspetta che non vi sia la presenza di *Influencer* tra i 5 utenti che hanno effettuato il maggior numero di retweet. Nel Figura 2.2.8 vengono riportati i 5 migliori score di *centrality_out_degree* all'interno della rete con i rispettivi *screen_name* degli utenti coinvolti:



user_screen_name	node_outdegree
Salvato95551627	22
LuciLuci78	19
AleGobbo87	18
alecardi7	17
sindelar3	16

Fig 2.2.8

Il numero maggiore di retweet è pari a 22 ed è curioso notare come non siano stati effettuati da un tifoso interista, bensì da un tifoso del Napoli, anch'essi eterni rivali della Juventus. Risulta anche interessante notare l'assenza di profili *bot* in questa statistica.

3. Sentiment analysis

Mentre nella sezione di *Social Network Analyisis* ci si è focalizzati sulla rete dei retweet e sul comportamento degli utenti relativamente a questo strumento, con la *Sentiment Analysis* ci si pone lo scopo di analizzare che cosa vogliano comunicare gli utenti tramite i loro tweet.

La *sentiment analysis* è stata eseguita seguendo un approccio **lexicon-based**, le fasi principali in particolare sono state le seguenti:

- **Tokenizzazione e Lemmatizzazione** del testo tramite la libreria **TreeTagger**
- preprocessing del testo: punteggiatura, stopwords, gestione hashtag e menzioni
- *sentiment score* calcolato tramite la libreria **VADER** e il lexicon italiano **Sentix**

Il concetto principale su cui si basa una *sentiment analysis* **lexicon-based** è ricondurre le parole contenute nelle singole frasi ai loro lemmi originali, per poterne cercare nel lexicon (o dizionario) il relativo punteggio di sentiment.
TreeTagger si è rivelato uno strumento efficace per questa analisi, effettua infatti una buona lemmatizzazione rispetto ad altre librerie testate come ad esempio *Spacy*, che implementa lemmatizzazione in italiano, ma nel nostro caso sembra aver performato peggio rispetto alla libreria scelta. La lemmatizzazione di *TreeTagger* avviene tramite l'individuazione della *Part-of-Speech(POS)* dei singoli termini all'interno di una frase, per verificarne la funzione sintattica e ricondurre la parola al suo lemma.
In un approccio di questo tipo risulta fondamentale verificare che gli strumenti siano adatti al contesto lessicale in cui si opera. Nel nostro caso, ovvero nell'analisi di tweet, avremo testi caratterizzati da periodi brevi, ricchi di punteggiatura e di elementi tipici del Web moderno come hashtag o menzioni ad utenti tramite *@username*. Per quanto riguarda gli strumenti scelti, *VADER* si presta particolarmente all'analisi di testi di questo tipo, considerando alcuni usi tipici del Web come l'uso di maiuscolo e punti esclamativi. Questi elementi sono considerati enfatizzanti rispetto ad un termine, per cui nelle analisi fatte le parole "bravo" "BRAVO" "BRAVO!" saranno considerate con un ordine crescente positivo, poiché le caratteristiche appena descritte ne esaltano la positività.
Sentix è invece un lexicon che fornisce punteggi di *sentiment analysis* per lemmi italiani, un progetto che ha come fonti principali *MultiWordNet, BabelNet, SentiWordNet*, rispettivamente due database lessicali e un'ontologia, tutti e tre tra le risorse più ampie nel panorama della *sentiment analysis*. Sentix nasce come strumento di un progetto più ampio, *Twita*, volto all' analisi di tweet in lingua italiana, si assume quindi possa essere uno strumento adeguato per le analisi di questo progetto.[5]
Bisogna comunque considerare che, rispetto agli strumenti proposti per la lingua inglese, le offerte disponibili per la lingua italiana sono ben più limitate.
Si presenta ora la pipeline che ha definito le fasi di lemmatization, tokenization e preprocessing del testo dei tweet, processo che ha portato ad ottenere il testo che verrà analizzato da *VADER* per effettuare la *sentiment analysis*. In particolare :

1. TreeTagger si occupa sia della tokenizzazione che della lemmatizzazione del testo, si ottiene in questa prima fase un testo tokenizzato e lemmatizzato, in particolare si effettuano le seguenti scelte: non si dividono punteggiatura e testo negli elementi tipici della piattaforma Twitter, ovvero @menzioni e #hashtag, al fine di sia poterli ricercare piu agilmente nelle fasi successive che di evitare eventuale corrispondenza tra i termini nel lexicon e quelli appena citati, che di

5. Twita, project page, howpublished = http://valeriobasile.github.io/twita/sentix.html, note = Accessed: 2021-02-01.

conseguenza verranno ignorati nell'analisi.
Per il funzionamento di VADER presentato precedentemente, si sceglie di non uniformare il testo in minuscolo e di appendere i punti esclamativi all'ultimo lemma presente nel testo prima di essi. Le emoji in questa fase non vengono estratte dal testo.

2. Vengono rimosse sia la punteggiatura che le *stopwords* (ovvero tutte le parole ignorabili in fase di sentiment come articoli o congiunzioni)
3. Viene trattata una peculiarità presente nel campo di testo dei retweet. Questi si presentano infatti nella forma "*RT utentetweetoriginale : testo originale*". Ai fini dell'analisi si sceglie di eliminare gli elementi di riferimento all'utente originale, mantenendo solo il testo del retweet che è la componente interessata alla *sentiment analysis*. I retweet risultano comunque identificabili dal campo presentato nella sezione di raccolta dati, ovvero '*retweeted_status*'
4. Gestione delle emoji: Queste vengono individuate e rimosse dal testo del tweet, successivamente salvate in una nuova colonna.

Si ottiene quindi un corpus pronto per la *sentiment analysis*, composto dal testo dei singoli tweet preprocessati. In questa fase si procede con il calcolo del sentiment score, usando le due risorse già citate, ovvero il lessico Sentix e la libreria VADER.
Dal lessico Sentix sono stati estratti e salvati in un dizionario la lista dei lemmi e la relativa polarity, ovvero un valore tra -1 ed 1 che indica la posizione del lemma nello "spettro" della sentiment, dove -1 è un valore totalmente negativo e +1 un valore totalmente positivo.
VADER, ovvero lo strumento che assegna un punteggio di sentiment ai nostri tweet con un approccio "lexicon e rule based"[6].Per quanto riguarda l'approccio lexicon based, VADER fornisce un suo lexicon, il quale viene aggiornato con il dizionario ottenuto da Sentix visto in precedenza.
VADER fornisce inoltre una serie di regole atte a valutare se ci sono elementi nel testo che esasperano o diminuiscono la sentiment vista nei lemmi. Alcune di queste sono state già menzionate, come il considerare testi in maiuscolo o punti esclamativi come elementi che esasperano il punteggio di sentiment.
La libreria fornisce anche ulteriori regole, tra cui la valutazione di elementi "*boosters*" e "*dampeners*", ovvero elementi che accrescono o attenuano il punteggio di sentiment delle parole a cui sono riferite.
Ad esempio "*very angry*" o "*kind of angry*", dove *very* e *kind of* sono due elementi rispettivamente *booster* e *dampner*.
Allo stesso modo sono definiti dei termini "negate", che chiaramente "invertono" la polarità del lemma. Tutto ciò è implementato per la lingua inglese, ne consegue che per la lingua italiana questi elementi non verranno mai trovati e pertanto ignorati.
Un altro elemento considerato da VADER sono le abbreviazioni tipiche del Web,

6. Vader, GitHub Repository, howpublished = https://github.com/cjhutto/vaderSentiment, note = Accessed: 2021-02-01.

come ad esempio le espressioni "*lol*" o "*pls*",che in questo caso risultano abbastanza "universali" tra le diverse lingue e pertanto sono considerate anche in un testo italiano. Si consideri che VADER permette la valutazione di singole stringhe di testo, fornendo un *compound score*, ovvero un valore di sentiment compreso tra -1 ed 1 normalizzato rispetto agli elementi nella stringa.
Un testo è classificato come neutro se il suo valore di compound risulta essere compreso tra -0.05 e 0.05. Se il valore di compound di un tweet risulta essere maggiore di 0.05, esso sarà classificato come positivo. Se il valore di compound di un tweet risulta essere inferiore a -0.05, esso sarà infine classificato come negativo.[7]
Tuttavia è interessante ai fini del progetto, non il *compound score* relativo ad un singolo tweet, bensì ottenere una valutazione rappresentativa di un certo periodo. A tal fine si è scelto di calcolare dei punteggi di *sentiment score* relativi a certi subset specifici, come ad esempio il pre-partita o il periodo successivo ad un gol.
Nel calcolare il valore medio dei *compound score* restituiti dalla *sentiment analysis* per ogni subset, si decide di utilizzare una stima intervallare piuttosto che una stima di tipo puntuale, dal momento che essa risulta più affidabile statisticamente. Dal momento che i valori medi di *compound score* sul set iniziale di dati si distribuiscono approssimativamente come una gaussiana, osservabile in Figura 3.1.1 e, dal momento che il numero di tweet è decisamente elevato, si sceglie di costruire l'intervallo di confidenza per la normale al 90%.

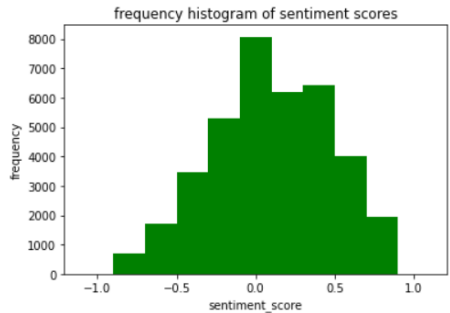


Fig 3.1.1

3.1 Analisi dei retweet

Si affronterà ora una questione cruciale per poter effettuare sentiment analisys sui dati da noi raccolti. Una buona parte dei tweet del nostro dataset sono infatti retweet, contenenti il testo del tweet originale. Si sono dovute fare delle assunzioni e prendere

7. Vader, GitHub Repository, howpublished = https://github.com/cjhutto/vaderSentiment, note = Accessed: 2021-02-01.

alcune decisioni per poterli trattare, infatti: un retweet non contiene testo aggiuntivo rispetto al tweet originale, questo ci permette di poter assumere che l'utente che effettua il retweet sia d'accordo con il contenuto del tweet originale e voglia dare ad esso maggior visibilita. Allo stesso tempo, si evince dalla *Social Network analysis*, la nostra rete presenta una struttura particolarmente *ego centric*, per cui alcuni tweet molto retweetati potrebbero influenzare eccessivamente l'analisi. Si propongono quindi due approcci diversi, al fine di valutare anche l'influenza dei retweet, nel primo questi verranno considerati come testo "autentico", per cui se un tweet è stato retweetato un certo numero di volte, quel testo verrà analizzato tante volte quanti i retweet presenti, assumento che gli utenti che effettuano retweet vogliano sostenere la stessa opinione o posizione. Un secondo approccio prevede l'esclusione dei retweet, mantenendo solo tweet con testo originale. Si noti che per tutte le situazioni analizzate tramite sentiment vengono paragonati i due approcci qui presentati.

3.2 Calcolo del sentiment score

Per la valutazione della sentiment si è scelto un approccio basato su eventi ed attori, andando a vedere se per protagonisti dell' evento il *sentiment score* cambia nel tempo. In particolare, una prima analisi è stata effettuata sugli allenatori. Si possono fare delle considerazioni interessanti su di essi, infatti Antonio Conte, allenatore dell'Inter, arriva alla partita da una serie di prestazioni non esaltanti, e la sua posizione è abbastanza criticata nell'ambiente. Al contrario, Andrea Pirlo, allenatore della Juventus, si presenta al match venendo da una situazione tranquilla, ma la pessima prestazione della squadra, porta i tifosi a mettere in dubbio la posizione dell'allenatore, tant'è che "Pirlo" è risultato essere un trend nel post partita, secondo in italia solo all'hashtag *#InterJuve*[8]. Per quanto concerne Antonio Conte, allenatore dell'Inter (Figura 3.2.1), i retweet sembrerebbero falsare i risultati, dal momento che ci si aspetta che il *sentiment score* nei confronti dell'allenatore della squadra che ha vinto una partita così importate in maniera netta, cresca passando dal pre-partita al post-partita in media. Essendo Conte in genere un allenatore spesso criticato, è accettabile anche il fatto che la differenza del *sentiment score* tra pre e post partita non sia statisticamente significativa, come evidenziano gli intervalli di confidenza. Tuttavia il punteggio di *sentiment score* medio risulta essere in ogni caso statisticamente positivo.

Per quanto riguarda i risultati relativi all'allenatore Andrea Pirlo si evince come sia differente la situazione in presenza di retweet e in assenza di essi (Figura 3.2.2). Infatti, escludendo i retweet dalla analisi in questo caso si ottengono risultati più credibili, dal momento che senza i retweet il *sentiment score* nel post-partita risulta in media più basso rispetto allo score del pre-partita. Questo risultato ha senso dal momento che la Juventus ha perso la partita, nonostante le aspettative fossero ben

8. Daily trends twitter 17 gennaio, ore 23, howpublished = https://getdaytrends.com/it/italy/2021-01-17/23/, note = Accessed: 2021-02-01.

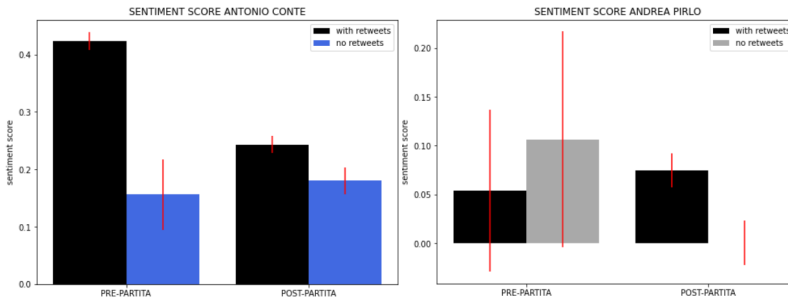


Fig 3.2.1 & 3.2.2

altre nel pre-partita, e questo ha portato gli appassionati ad esprimere giudizi più negativi nei suoi confronti. L'approccio, che poi verrà replicato anche nelle successive analisi, prevede il calcolo del compound score tramite la funzione *senitment_subset()*, selezionando gli hashtag più usati relativi ai due allenatori.
Un'ulteriore analisi è stata effettuata relativamente al giocatore **Arturo Vidal**, centrocampista dell'Inter ed ex della Juventus. Vidal è stato al centro di due eventi particolarmente polarizzanti e contrastanti riguardo la partita.
La prima riguarda un episodio del pre-partita, ovvero un presunto bacio dato allo stemma della Juventus sulla maglia dell'avversario ed ex compagno di squadra Giorgio Chiellini. Fatto altamente criticato dai tifosi interisti. Il secondo evento è il gol effettuato dallo stesso Vidal nei primi minuti della partita, il quale ha chiaramente suscitato una reazione opposta al primo evento.

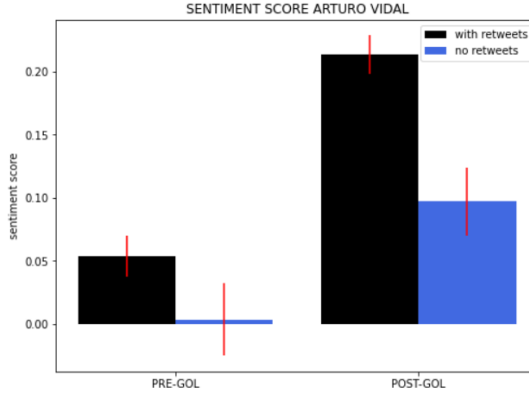


Fig 3.2.3

Si è scelto quindi di calcolare il compoud score relativo a Vidal, dividendo in due subset, uno precedente al gol ed un altro successivo ad esso. Dalla figura 3.2.3 si evince come sia inserendo nell'analisi i retweet che tralasciandoli, la crescita del sentiment score, passando da un periodo precedente al gol, ad un periodo post gol, sia statisticamente significativa, come ci si aspettava. Inoltre, si evince che il valore medio di *sentiment score* risulti essere nettamente migliore includendo i retweet nell'analisi. Questo accade a causa del tweet di festeggiamento pubblicato dall'Inter, chiaramente positivo e ritwittato da moltissimi utenti appartenenti alla community pro-Inter.

L'ultima analisi effettata relativa alla sentiment è stata svolta sul giocatore Cristiano Ronaldo, stella della Juventus, uno dei più popolari calciatori moderni e, di conseguenza, uno dei piu chiacchierati sulla piattaforma Twitter. Per lui si propone un apporoccio diverso, andando a dividere il nostro dataset iniziale in pre partita, primo tempo, intervallo, secondo tempo e post partita, scelta motivata anche dalla mancanza di azioni salienti da parte del giocatore.
Sia tenendo conto dei retweet che escludendoli, dalla Figura 3.2.4 differentemente da quanto ci si potesse aspettare, non si evince un vero e proprio pattern discendente in termini di *sentiment score* medio attraversando le diverse fasi. Le differenze tra i *sentiment score* calcolati con e senza retweet nella varie fasi della serata calcistica non risultano statisticamente significative. Tuttavia, si evince come nel secondo tempo il punteggio medio di sentiment risulti inferiore a zero, significativamente inferiore da un punto di vista statistico rispetto al punteggio del pre-partita:

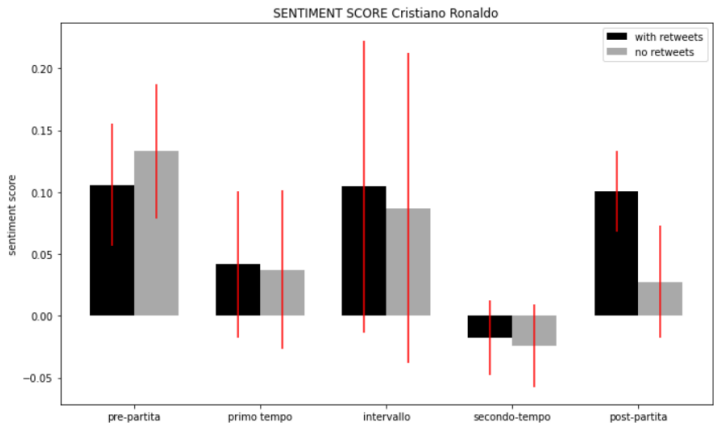


Fig 3.3

Wordcloud Sono stati effettuati degli WordCloud, ovvero delle rappresentazioni per comprendere quali sono state le parole più usate in diverse fasi dell'evento. Si sono scelte in particolare il pre-partita, la partita stessa e il post-partita, al fine di poter valutare quali sono stati gli argomenti più discussi nelle varie fasi.

Fig 3.4

Risulta particolarmente interessante questo primo WordCloud, dove risalta l'evento molto discusso del bacio di Arturo Vidal, nonché un altro episodio abbastanza chiacchierato, ovvero l'uso di fuochi d'artificio dei tifosi interisti presenti fuori dallo stadio.
Si notano anche molti termini che indicano il grado di tensione tra i tifosi e tra fazioni, con parole come "ansia" e qualche epiteto non molto elegante, tuttavia di uso comune nelle discussioni di calcio del web.
L'analisi non porta risultati interessanti invece per quanto riguarda la partita e il post partita, dove i termini più ricorrenti sono risultati essere banali e non significati, come "gol" o "partita".

Conclusioni e Sviluppi futuri

In conclusione, si può affermare che il progetto abbia fornito risposte abbastanza soddisfacenti alle domande di ricerca. Riassumendo, si è compreso che gli utenti che prendono parte alla rete dei retweet hanno un grado di interconnessione molto basso, tendendo a retwittare esclusivamente i tweet dagli account di Inter e Juventus, senza retwittarsi tra di loro. Se si prende parte alla rete dei retweet ci si trova all'interno di una rete sociale di tipo ego-centrica, in cui pochi utenti ottengono un grande numero di retweet, senza praticamente effettuarne a loro volta.
Effettuando le analisi di sentiment sui tweet degli utenti si è compreso quanto e in che modo i retweet possano influenzare gli score ottenuti, talvolta risultando fuorvianti per le analisi, soprattutto in quelle relative agli allenatori delle due squadre. Dai risultati della *sentiment analysis* effettuata in relazione alla serata di Ronaldo, si ottiene un risultato differente da quanto ci si attendeva. Ci si aspettava una reazione più negativa da parte degli utenti in seguito alla deludente prestazione del giocatore. Nell'analisi delle parole più utilizzate nel prepartita si ottiene il risultato atteso invece. Infatti, gli utenti italiani sembrano aver dato più importanza alla faccenda riguardante il presunto

bacio dello stemma della Juventus da parte di Vidal che all'imminente fischio d'inizio.
Per quanto riguarda i possibili sviluppi futuri, potrebbe essere utili a fini migliorativi dell'analisi il fatto di utilizzare una API Twitter Premium invece che l'API Standard V1.1, la quale è risultata essere limitante nella raccolta di determinati campi dei tweet. Una scelta di questo tipo potrebbe permette di estendere il lavoro ad ulteriori analisi relative ad altri strumenti di Twitter, come ad esempio le risposte o le "*quote*", ovvero retweet contenenti testo aggiuntivo.
Un secondo possibile sviluppo del progetto potrebbe riguardare l'analisi dei tweet provenienti dall'estero, al fine di comprendere quanto sia sentita e seguita la partita fuori dall'Italia, dal momento che le due squadre hanno tifosi in tutti i paesi del mondo. Chiaramente uno sviluppo di questo tipo implicherebbe uniformità nella valutazione della sentiment, risultando particolarmente limitata dall'eterogeneità delle librerie relative alle diverse lingue. Un altro possibile sviluppo potrebbe essere legato al lessico per la *sentiment analysis* in italiano, come infatti accennato in precendenza, la libreria VADER non nasce con il supporto all' italiano (o meglio, effettuerebbe analisi in italiano tramite una traduzione automatica di dubbiosa efficacia, che si è quindi scelto di ignorare), sarebbe pertanto interessante implementare manualmente tutte le funzionalita che VADER fornisce per la lingua inglese.
Premettendo comunque che gli strumenti per analisi in italiano non sono cosi diffusi e devono scontrarsi con la comunicazione del web, fatta dei suoi usi e costumi ed in continuo sviluppo, ci si ritiene soddisfatti rispetto alle aspettative iniziali.
Un elemento ormai diffusissimo nella comunicazione Web ma che si è dovuto accantonare in questo progetto è la comunicazione tramite emoji, una realta complessa e in continuo sviluppo, ma che è sempre piu parte integrante dei nuovi mezzi di comunicazione e che sarebbe interessante trattare, di pari passo con l'analisi testuale.

References

Daily trends twitter 17 gennaio, ore 23, howpublished = https://getdaytrends.com/it/italy/2021-01-17/23/, note = Accessed: 2021-02-01.

K Mean Clustering algorithm, howpublished = https://www.learndatasci.com/tutorials/k-means-clustering-algorithms-python-intro/, note = Accessed: 2021-02-01.

Slide del corso, howpublished = https://elearning.unimib.it/course/view.php?id=31236, note = Accessed: 2021-02-01.

Twita, project page, howpublished = http://valeriobasile.github.io/twita/sentix.html, note = Accessed: 2021-02-01.

Twitter API search, howpublished = https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets, note = Accessed: 2021-02-01.

Vader, GitHub Repository, howpublished = https://github.com/cjhutto/vaderSentiment, note = Accessed: 2021-02-01.