Analisi dei Tweet e valutazione del sentiment dei tifosi nella sfida Tottenham-Manchester City

Cristiano Ruttico 809360, Enrico Ragusa 862702, Francesco Martinelli 873685

Introduzione

Twitter è un servizio di notizie e micro-blogging fornito dalla società Twitter con il quale gli utenti postano e interagiscono con messaggi chiamati tweet. Questi possono essere etichettati con l'uso di uno o più hashtag, parole o combinazioni di esse concatenate precedute dal simbolo # attraverso il quale l'utente crea un collegamento ipertestuale a tutti i messaggi che citano lo stesso hashtag. Per questo progetto il gruppo di lavoro ha scelto di commentare un evento sportivo avvenuto il 15 agosto 2021. Si tratta della partita di Premier League Tottenham-Manchester city valida per la prima giornata di campionato. È stato scelto questo evento, poiché l'evento rappresentava il primo big match della stagione 2021/2022 di Premier League. È certamente una sfida molto affascinante e, soprattutto, seguita. La partita ha acquistato una certa importanza mediata anche per il fatto che il giocatore più importante del Tottenham, Harry Kane, era al centro di una trattativa tra le due squadre.

Sommario

1. Domanda di ricerca

Ci si è proposti di impostare la domanda di ricerca principale sulla **sentiment analysis** dei tweet derivanti dalla sessione di streaming oltre al conteggio dei tweet. Inoltre, si è deciso di

integrare il sentiment con alcuni eventi avvenuti durante la partita in modo che fosse possibile **interpolare le azioni principali con il sentiment medio**, oppure con il numero di tweet per squadra. La visualizzazione principale di questo progetto, infatti, si propone di mostrare come gli avvenimenti della partita possono aver influito su quest'ultimo aspetto e sul sentiment medio.

Ci si è altresì proposti di riportare tutta una serie di statistiche secondarie sulle prestazioni dei giocatori, sul conteggio degli hashtag e del numero di rIcorrenze dei nomi dei giocatori.

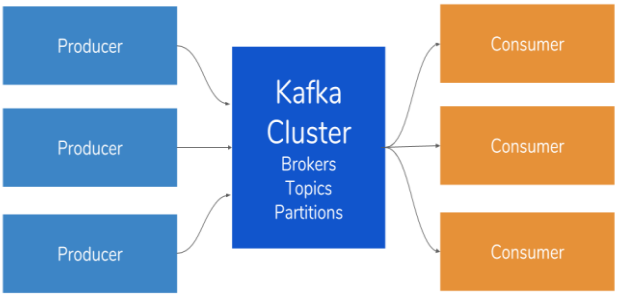
A tali fini le due V scelte per questo progetto sono **Velocity** e **Variety**.

2. Velocity

Le due V scelte per questo progetto sono **Velocity** e **Variety**. Per la raccolta dati L'architettura implementata per raccogliere i dati prevede l'uso di tre strumenti fondamentali: un sistema di messaggistica (**Kafka**), le API di **Twitter Api** e librerie di Python (kafka-python e **tweepy**) per integrare le componenti. La parte di Velocity è stata svolta attraverso lo streaming di dati tramite le API di Twitter e in particolare della libreria di Tweepy di Python; i dati sono poi stati immagazzinati nel sistema di storage Mongo db.

2.1 Kafka

Kafka è una piattaforma streaming distribuita che permette di archiviare ed elaborare flussi di record in tempo reale. All'interno di Kafka agiscono principalmente 3 elementi: **producer, consumer e il cluster**. Il primo si occupa di pubblicare e scrivere messaggi sui topic, il secondo è incaricato di leggere i messaggi da un topic a cui si è iscritto, mentre il terzo si occupa di organizzare i messaggi nei vari topic.



2.2 API

Le API utilizzate per la raccolta dei dati sono le **API di Twitter**. Per le Application Programming Interfaces (API) di Twitter, è stato necessario creare un account sviluppatore dove reperire le chiavi di accesso all'infrastruttura. Il sistema di API presenta due chiave per le API (una pubblica e una privata) e due token di accesso.

Il servizio è stato utilizzato per ottenere tutti i tweet durante l'orario temporale definito (30 minuti prima dell'inizio della partita fino a 1 dopo la fine), ed è stato stabilito di filtrare i tweet in base a due parametri: la **lingua** e gli **hashtag**. Per la lingua si è deciso di ottenere solo i tweet in lingua inglese mentre per gli hashtag si è deciso di filtrare i tweet in base agli hashtag ufficiali delle due squadre oltre a quelli della Premier League. In particolare, di seguito

riportiamo la tabella contenente gli hashtag utilizzati per le due squadre.

2.3 Librerie Python

Le principali librerie utilizzate per la raccolta dei dati sono **kafka-python** e **tweepy**. La libreria kafka-python è servita per implementare la struttura kafka in Python, mentre la seconda per accedere alle API di Twitter. L'architettura globale si incentra sulla realizzazione di dodici script Python per gestire lo stream dei dati. In particolare sono stati realizzati 4 producer: uno incaricato di reperire i tweet riguardanti il Manchester City, uno riguardante i tweet del Tottenham, uno riguardante il match in generale, e uno riguardante il calciatore Harry Kane. Per quanto riguarda il salvataggio dei dati si è provveduto a sviluppare 8 consumer, 2 per ogni producer; in particolare per ogni producer si è creato un consumer in grado di scrivere direttamente sul database mongo db e uno per scrivere direttamente in file json.

Riportiamo di seguito i **campi** che sono stati estratti dai tweet:

'user_id': Identificativo dell'utente

'username': Nome utente

'screen_name': Nome utente visualizzato su twitter

'text': Testo del tweet

'full_text': Testo completo del tweet

'hashtags': Hashtag presenti nel tweet

'retweeted': Assume valori 0 e 1 a seconda se sia stato retweetato o meno

'retweet_count': Numero di reetweet

'favorite': Numero di like

'followers': Numero di Followers

'created_at': Orario di pubblicazione del tweet (server di twitter)

Abbiamo quindi ottenuto un totale di **4 file json**: per la precisione uno riguardante i tweet contenenti gli hashtag del **Manchester City**, uno riguardante gli hashtag del **Tottenham**, uno per il **match** in generale, e uno per il giocatore Harry Kane (poiché al centro di voci di mercato riguardanti le due squadre). Si nota che i dati riguardanti Kane ai fini dell'analisi non sono poi stati utilizzati.

3. Variety

Per quanto riguarda la parte di variety si è scelto di **integrare** il file (in realtà sono più file derivanti da diverse sessioni) con i file (csv) di cronaca della partita. Si tratta di due dataset ricavati da informazioni presenti sul web raccolte dopo la partita; i dataset sono stati successivamente ripuliti, unificati in un solo file di cronaca e successivamente messi assieme ai risultati dello streaming.

4. Pulizia data-set di streaming

La prima fase relativa alla pulizia del dataset ha richiesto la **trasformazione dei file json in file csv**: in questa fase non sono stati riscontrati particolari ostacoli, poiché i file non presentavano annidamenti. Dopodiché, si è proceduto ad analizzare singolarmente i file delle due squadre.

I dati non erano ancora pronti per essere processati, bensì necessitavano di una parte di pre-processing in modo da rendere i data-set più facilmente fruibili.

4.1 Pulizia del Testo

La fase di **pulizia del testo** ha visto l'eliminazione della dicitura retweet, dei nomi preceduti da @ e dei separatori di riga.

Dopo questa prima procedura, tutte le lettere delle stringhe di testo sono state trasformate in lettere minuscole, sono state rimosse le cifre, le stopword, la punteggiatura (inclusi gli apostrofi) e qualsiasi riferimento a siti web. Tutte queste operazioni sono state eseguite tramite espressioni regolari. L'utilizzo di espressioni regolari ha visto l'utilizzo della funzione Lambda, associato alla sostituzione dei valori in questione con valori nulli.

Per questa fase si è scelto di non proseguire con stemming o tokenizzazione del testo. Lo stemming, l'estrazione delle radici delle parole (inglesi) non era necessario poiché, ancora una volta, non si voleva cogliere il significato semantico. La tokenizzazione, la divisione di una stringa di testo in una lista di parole, non è sempre indicata e può essere ignorata se si scelgono tecniche di pre-processamento differenti.

Si sottolinea, innanzitutto, che **i tweet talvolta non presentavano il testo completo**. Ciò che si è scelto di fare è stato creare un campo 'text_clean' riportante il testo completo laddove vi fosse, sennò riportante il testo troncato.

4.2 Pulizia degli Hashtag

Si è proceduto poi alla creazione un nuovo campo 'compl_hashtags', che **integrhi gli hashtag** considerati come hashtag in fase di acquisizione, con quelli ottenuti tramite uno script di regex e analisi del testo, che aveva il compito di controllare se ci fossero hashtag nel campo di testo pulito che non fossero stati ottenuti in fase di acquisizione.

4.3 Pulizia ed elaborazione del Tempo

Un campo particolarmente rilevante per procedere all'integrazione è quello relativo al **tempo**: dalla data di pubblicazione del tweet, infatti, si può risalire al minuto relativo della partita e all'evento verificatosi in tale momento. Si è deciso di compensare il fuso orario e di discretizzare il tempo su base di 15 secondi per ottenere delle fasce temporali (seppur molto granulari) piuttosto che singoli istanti. La discretizzazione ha richiesto che il formato date time fosse trasformato in stringa e successivamente in float. Estraendo singolarmente (in un DataFrame apposito) i minuti, i secondi e le ore (prima della trasformazione in interi) è stato possibile modificarne i valori. In particolare, tutti i secondi sono stati divisi per 15, sono stati arrotondati all'intero più vicino, sono stati rimoltiplicati per 15. Inoltre, la presenza di tuple con secondi uguali a 60 ha richiesto l'utilizzo di un filtro che assegnasse il valore 0 ai secondi e aggiungesse un minuto alla colonna dei minuti; lo stesso procedimento è stato eseguito con i minuti e le ore. Alla fine del procedimento i dati sono stati trasformati in stringa e ritrasformati in formato temporale (il formato originale).

Dopodiché, con una serie di funzioni applicate ad ogni tweet, sulla base dell'**orario di pubblicazione** è stata attribuita **un'etichetta con riferimento al momento della partita in corso**: quando pubblicato (pre-partita, primo tempo, secondo tempo, intervallo, post-partita).

È stato anche effettuato un procedimento che **riconducesse il conteggio dei minuti al minuto di riferimento della partita**, assumendo che la partita fosse iniziata alle 17:30 italiane e ipotizzando un minuto di compensazione per il tempo trascorso tra evento e commento sul Social Network. Dalle analisi svolte successivamente l'ipotesi adottata risulta ragionevole. Inoltre, per poter distinguere tra recupero del primo tempo e i primi minuti del secondo tempo (46° può essere sia il primo minuto recupero del primo tempo, sia il primo minuto del secondo tempo), si è scelto di riportare i minuti di recupero all'interno della colonna, ad esempio il minuto 46 del primo tempo diventa il minuto 45+1: si elimina così il conflitto di omonimia.

Per quanto riguarda il file relativo ai tweet sul match in generale, il percorso seguito è analogo ai precedenti, fatta eccezione per la parte di integrazione in cui si è dovuto tener conto della presenza di tweet già trattati nei precedenti file. Si è deciso di escludere fin da subito dal file sul match i tweet già trattati negli altri due (conservando però quelli presenti solo negli altri file in questa prima fase di integrazione). Si è proceduto in seguito alle medesime elaborazioni esposte sopra. Lo script di riferimento si può trovare in *Pre_processing_Match.ipynb*.

5. Analisi del sentiment

Per l'**analisi di sentiment** si è scelto di utilizzare la libreria *VaderSentiment*, in particolare

SentimentIntensityAnalyzer(). Con il **sentiment** si intende lo stato d'animo di un soggetto, che in questo caso può essere inteso come umore per gli eventi della partita. Attraverso l'utilizzo di Vader sul testo ripulito è stato analizzato il sentiment dei tweet. L'output della sentiment analysis consiste nel calcolo di quattro valori: i primi tre sono positive, negative e neutral che rappresentano la proporzione di parole associabili agli umori di riferimento nel testo (ad esempio parole come "horrible" contribuiranno ad aumentare il valore negative poiché parola associata al cattivo umore).

L'ultima misura è il **compound**, ovvero una misura del **sentiment aggregato** (calcolata sulla base degli altri tre indicatori), che assume valori compresi tra -1 e 1. Sulla base del compound si è attribuita ad ogni tweet una **label** che riassumesse l'umore dell'utente; le tre label assegnate sono state: "Positive" se il compound è superiore a 0.05, "Negative" se il compound è inferiore a -0.05 e "Neutral" negli altri casi. Questi range non sono arbitrari, bensì sono stati indicati dai creatori della libreria. I quattro valori sono comunque tutti presenti nel data-set sotto forma di dizionario.

6. Determinazione del Tifo

A questo punto facendo delle considerazioni sugli hashtag è stata implementata una funzione che attribuisse un tifo di riferimento al tweet tra City o e Tottenham o lo classificasse come neutrale: ad ogni tweet veniva attribuito il tifo sulla base degli hashtag citati e del compound. L'**ipotesi sul sentiment** consiste nella supposizione che un tifoso neutrale commenti la partita senza espressioni particolarmente di parte (eccessivamente positive o negative per una squadra o l'altra), perciò si pone la condizione di compound neutrale. L'**ipotesi sugli hashtag** si basa sul fatto che un tifoso neutrale tendenzialmente utilizza hashtag generici o di entrambe le squadre. Nonostante le ipotesi non siano deterministiche, da un'esplorazione dei tweet si è notato come l'approccio utilizzato fosse in grado di classificare con un consistente grado di correttezza il tifo.

Come ultima procedura si sono binarizzate le label "Positive", "Negative" e "Neutral" per rendere più agevoli il calcolo di valori aggregati necessari all'integrazione. Tutto ciò si può trovare nel file *Pre_processing_Spurs.ipynb* e *Pre_processing_City.ipynb*.

7. Pre-processing dataset di cronaca

I data-set utilizzati per trattare la **cronaca** sono due data-set csv, che trattano due parti diverse della degli eventi del match. Il primo data-set

comprende il computo di tutti i tiri (inclusi eventuali assist) mentre il secondo presenta tutte le altre azioni importanti (ad esempio calci d'angolo e cartellini).

Il **primo data-set**, *cronaca.csv*, presenta **esclusivamente i tiri** e non ha necessitato di particolari elaborazioni; è stato scaricato da *fbref.com* Le uniche modifiche apportate sono quelle ai nomi dei giocatori o a quelli delle azioni, eliminando eventi secondari e informazioni superflue ai fini dell'analisi (eventi precedenti al tiro come ad esempio passaggi o dribbling). Il data-set è stato ottenuto dal sito *fbref.com* il quale permetteva l'esportazione in formato csv.

Il **secondo dataset**, *scraping.csv*, è stato ricavato tramite scraping dalla pagina web *365scores.com* (un sito contenente dettagliatamente la cronaca di incontri calcistici). Il risultato dello scraping è stato un csv contenente due campi: uno con il minuto della partita e un secondo campo chiamato testo, il quale riportava una stringa di testo contenente la cronaca con gli eventi di quel minuto.

Il pre-processamento ha richiesto l'utilizzo di **espressioni regolari** sul campo del testo in modo da poter estrarre e immagazzinare in un'altra colonna il giocatore (o i giocatori nel caso di una sostituzione) e l'evento ad esso associato. In questo caso sono stati ignorati tutti i tiri perché già presenti nell'altro data-set (più completo sotto questo aspetto). Infine, sono stati aggiunti valori speciali per l'inizio della partita, del secondo tempo e per la fine della partita.

Dopo aver reso coerenti, nei due data-set, i nomi delle colonne e dei giocatori (risolvendo conflitti di sinonimia), è stata effettuata la **concatenazione** (ct.csv). La colonna dei minuti aveva però diverse azioni che si sovrapponevano nello stesso minuto. Ciò costituiva un ostacolo in fase di integrazione con i tweet poiché non esisteva una **chiave primaria** (cosa che era impossibile nel file dei tweet). Per ovviare a questa situazione senza perdere troppa informazione, si è deciso di conservare i data-set per squadra, eliminare le azioni meno rilevanti (nei pochi minuti dove si sovrapponevano) e in un caso (al 79° per la parte del Manchester City) fare un'approssimazione di un minuto. Dopo aver ricavato **due data-set distinti per ognuna delle due squadre**, è stato assegnato una colonna "Team" per la squadra.

8. Creazione e pulizia data-set per statistiche giocatori

Oltre ai data-set principali è stato creato un dataset con le **statistiche dei giocatori**. Per

farlo è stato necessario scaricare i dati da *fbref.com*.

Le statistiche della partita presenti per ogni giocatore sono: minuti giocati (Min), goal (Gls), assist (Ast), tiri tentati (Sh), tiri in porta (SoT), tocci effettuati (Touches), azioni di pressing (Press), tackle effettuati (Tkl), intercetti (Int), blocchi (Blocks), expected goals (xG), non-penalty expected goals (npxG), expected assist (xA), shot-creating actions (SCA), goal-creating actions (GCA), passaggi completati (Cmp), passaggi tentati (Att), percentuale di passaggi completati (Cmp%) ed altre statistiche eliminate in seguito. Inoltre, nel dataset erano anche inclusi il nome della squadra e la posizione del giocatore (in inglese).

Nello script *Statistiche.ipynb* sono state svolte alcune elaborazioni, in particolare: è stato creato il campo con la squadra di riferimento e i due dataset sono stati uniti. Ai giocatori che avevano svolto più ruoli sono stata attribuiti un primo e un secondo ruolo.

9. Integrazione

9.1 Tweet e calcolo valori aggregati

Nel file *Data_Tweet_Sentiment.ipynb* è stata eseguita l'**integrazione** dei file relativi ai tweet prodotti dalle precedenti elaborazioni. In particolare, sono stati **concatenati** i file, sono stati corretti i type dei campi e, infine, sono stati **eliminati i tweet duplicati**. Dopodiché si è **raggruppato** per Min_Cronaca e Tifo eseguendo le sum dei campi relativi al sentiment, si sono ottenuti così il numero di tweet positivi, negativi e neutri più un campo con la somma del compound, per minuto della partita e per tifoseria. Infine, sono stati calcolati **valori aggregati** utili per le successive elaborazioni ovvero un campo con il numero totale dei tweet (ottenuto come somma dei tweet divisi per sentiment) e un valore di compound medio.

9.2 Tweet e cronaca

Dal data-set creato come esposto al paragrafo precedente si è proceduto **all'integrazione con i file relativi alla cronaca**. In prima battuta si sono integrati tra loro i file relativi alla cronaca delle singole squadre. Dopodiché si è corretto nel data-set sulla cronaca il minuto della partita per uniformarlo a quello presente nel file dei tweet (quindi stringa con formato per il recupero del tipo es. "45+2"). Il file della cronaca, poi, sono stati integrati con il file dei tweet attraverso una **left-join sul minuto della partita**. La left join è stata eseguita tramite la funzione merge con left side il data-set dei tweet e right side i data-set della cronaca. Si è ottenuto così un file che mostrasse per ogni minuto della partita, il

sentiment e il contemporaneo evento accaduto nel match, il file offre così un'overview dell'andamento del sentiment durante il match in relazione all'evoluzione degli accadimenti della partita. Questa parte si può trovare nello script *Integrazione_Tweet_Cron_Stats.ipynb*.

9.3 Tweet, cronaca e statistiche

A partire dal file integrato tra Cronaca e Tweet, facendo riferimento al campo **Player**, si sono integrate le **statistiche relative al giocatore protagonista dell'evento** della partita per ambo le squadre eliminando prima le statistiche poco significative. Anche questa parte è presente nel file *Integrazione_Tweet_Cron_Stats.ipynb*.

9.4 Citazioni e statistiche

In seguito, ci si è posti l'obiettivo di **contare il numero di volte in cui ciascun giocatore veniva citato nei tweet** (script di riferimento *Citazioni_Giocatori.ipynb*). Attraverso la funzione **findall** di Python per ogni giocatore è stato conteggiato il numero di volte in cui ricorreva. Per trovare il numero esatto di ricorrenze di nome e cognome si è tenuto conto dei nomi composti e dei soprannomi (sia per includere tali ricorrenze, sia per evitare duplicati). Inoltre, ad ogni computo è stato corretto il nome per l'integrazione (per risolvere situazioni di conflitto dovute a sinonimia).

Come si può notare all'interno dello script *Integrazione_Tweet_Cron_Stats.ipynb*, come ultima integrazione abbiamo creato un data set che aggregasse le **statistiche del giocatore e il numero di volte che il giocatore era stato citato nei tweet**: è stata eseguito l'enrichment del dataset statistico attraverso una merge con metodo left sul campo Player, per aggiungere al file con le statistiche il numero di citazioni del giocatore. Da questo data-set si può vedere il numero di volte che un giocatore viene citato e la prestazione dello stesso attraverso le statistiche.

I data-set sono poi stati esportati e utilizzati per produrre visualizzazioni. Di sotto si riporta un esempio.

10 . Conclusioni

Dopo la presa visione del report si può concludere che il gruppo è riuscito a rispondere con successo alla domanda di ricerca. La sessione di streaming ha prodotto dei dataset rilevanti sui quali è stato possibile svolgere un'analisi approfondita, che va anche oltre gli scopi iniziali della domanda di ricerca.

L'analisi di sentiment ha fornito spunti interessanti, dove comparata con gli accadimenti della partita. Se si tiene conto del tifo, la visione di insieme risulta ancora più chiara, come si può notare nel primo grafico. Si vede senza ombra di dubbio che i tweet avevano un'elevata corrispondenza con gli eventi della partita e con il tifo. Inoltre, anche il conteggio del numero dei tweet in base al minuto ha messo in luce gli stravolgimenti interni alla partita.

Analisi delle statistiche di gioco, degli hashtag, delle occorrenze dei nomi hanno permesso una descrizione ancora più completa degli avventi della partita.

Il gruppo si può ritenere soddisfatto dei risultati ottenuti, in particolare della parti relative allo streaming dei tweet, alla sentiment analysis e all'integrazione con i file cronaca. Inoltre, le fasi di pre-processamento del testo sono riuscite a fornire dati di grande qualità.

Il gruppo si augura che la fruizione del proprio lavoro possa essere utile (in astratto) a tutti gli appassionati di calcio che seguono big match come quello descritto.

Appendice:

Cristiano Ruttico si è occupato dello streaming dei tweet, della pulizia degli hashtag, dell'algoritmo per determinare il tifo, della creazione dei dataset di cronaca e del loro pre-processing.

Francesco Martinelli si è occupato dello streaming dei tweet, della parte finale di integrazione dei vari dataset, della gestione dei minuti della partita, di una parte dell'assegnazione del tifo; si è anche occupato del file delle statistiche.

Enrico Ragusa si è occupato dell'analisi di sentiment, della creazione di una pipeline di pulizia per i file di testo, della discretizzazione del tempo dei tweet, ha preparato i file di cronaca per l'integrazione, ha effettuato il count dei giocatori.