Università Ca'Foscari Venezia

Dipartimento di Scienze Ambientali, Informatica e Statistica

Corso di Web Intelligence - A.A. 2019-2020

Docente: Prof. Claudio Lucchese

Relazione progetto

Data: 27/08/2020

Versione 1.0

867704 Matteo Schizzerotto
870592 Goran Gajic

Indice

1. Obiettivo progetto

L'obiettivo di questo progetto è predire il vincitore dell'Australian Open edizione 2020 tramite una simulazione *match by match* di tutte le partite.

In particolare, tramite l'analisi di partite già giocate, si possono ottenere le caratteristiche significative di tutti i giocatori di tennis di interesse. Attraverso queste è quindi possibile prevedere il risultato di match arbitrari e, simulando le singole partite, siamo in grado di individuare il vincitore di un generico torneo.

Utilizzando queste capacità si è dunque in grado di raggiungere l'obiettivo prefissato.

2. Introduzione

2.1 Dataset usati

I dataset presi in considerazione per le partite sono quelli degli anni dal 2017 al 2019 presenti nel sito Tennis-Data.co.uk.

Il dataset dei giocatori invece è stato preso da ausopen.com analizzando le loro API private usate per la generazione delle pagine.

Sono stati scelti questi dataset perchè rappresentavano il maggior numero di dati di interesse in rapporto alla dimensione dei dataset stessi.

2.2 Uso dei dati

Per ottenere tutte le partite sono stati concatenati i dataset dei tre anni.
A questo punto sono state rimosse le colonne non necessarie, tra queste:

- le colonne relative ai siti di scommesse;
- le colonne con il numero di set e punti per set delle singole partite;
- le colonne che identificano il torneo e le date.

Sì è deciso di rimuovere queste colonne in quanto non permettevano un miglioramento dell'analisi ai fini del progetto.

Per quanto riguarda invece il dataset dei giocatori sono state rimosse le informazioni che risulatano essere unicamente utili al sito ausopen.com e di secondaria importanza (es. allenatore e residenza).

3. Analisi iniziale

Dopo aver rimosso le colonne superflue, sono stati analizzati i dataset per trovare eventuali problemi o incongruenze. Questo è risultato necessario per evitare doppioni o errori di calcolo.

3.1 Osservazioni, problemi e risoluzione

Essendo state trovate delle inconsistenze tra i dataset dei giocatori e quelli delle partite - per esempio con i nomi degli atleti - si è deciso di modificarli in modo tale da renderli omogenei.

Inoltre, siccome nel dataset dei giocatori erano presenti dei JSON innestati è stato necessario normalizzarli per poter estrarre alcune colonne.

Le righe del dataset delle partite con NaN presente sulle colonne di interesse sono state scartate.

Per evitare di generare bias sono stati alternati vincitore e sconfitto all'interno del dataset.

4. Creazione colonne

Dopo aver risolto tutti i problemi elencati precedentemente, si è passati alla creazione di nuovi dati.
Questi ultimi vengono utilizzati in combinazione tra loro per la generazione di nuove *feature.*

Le *feature* verranno poi sfruttate dai classificatori per comprendere come estrarre il vincitore di ogni match.

4.1 Colonne dataset giocatori

Al dataset dei giocatori vengono aggiunte diverse colonne per comprendere meglio le peculiarità di ciascuno di essi. Tra queste le più importanti sono:

- percentuale assoluta di vittorie del giocatore;
- percentuale di vittorie negli ultimi tre anni;
- affinità alle diverse tipologie di terreno (*hard*, *grass*, *clay*)
- affinità a ogni tipologia di stadio (*indoor*, *outdoor*)

4.2 Colonne dataset partite

Nel dataset delle partite sono state sostituite le colonne relative alla tipologia di terreno e stadio con delle matrici categoriali che permettono di identificare più facilmente ed efficientemente queste caratteristiche.

4.3 Unione dataset

Dopo aver generato le colonne all'interno del dataset delle partite e quello dei giocatori si è proseguito con l'unione di questi due dataset.

Al posto di mantenere le *affinity* per ogni tipologia di campo e terreno sono state mantenute solamente le *affinity* legate al campo e terreno effettivamente presenti nelle singole partite.

Dopo aver unito i dataset, per ogni partita, sono stati calcolati i delta delle *feature* dei due giocatori coinvolti nel match (es. percentuale vittorie).

Questo permette di ridurre il numero complessivo di *feature* e al contempo aiuta i classificatori ad ottenere una *accuracy* migliore.

5. Classificatori

5.1 Classificatori utilizzati

I classificatori che sono stati utilizzati sono i seguenti:

- regressore lineare
- k-Nearest-Neighbor
- albero di decisione
- albero di decisione con begging
- albero di decisione con boosting
- random forest

Per individuare i migliori parametri per i singoli classificatori, viene calcolata l'*accuracy* di ogni modello attraverso una simulazione su un dataset di test alternando alcune possibili combinazioni di parametri.

5.2 Risultati dei classificatori

Per poter valutare l'accuracy dei classificatori è stato diviso il dataset in un *subset* di *train* e uno di test; allenando i classificatori sul *train* e confrontando i risultati con quello di test.

Nel complesso i classificatori hanno raggiunto al massimo 70% di *accuracy*. Questa accuracy è sicuramente migliore della *random chance* ma non permette una *confidence* assoluta quando i risultati vanno a dipendere l'uno dall'altro (es. in un torneo).

Alcuni dei modelli migliori sono:

- k-Nearest-Neighbor
- albero di decisione con begging
- random forest

6. Seconda creazione *feature*

6.1 Idea

Per cercare di ottenere una precisione migliore sono state generate nuovamente le *feature*.
Questa volta però vengono considerate solamente le partite precedenti alla singola partita presa in analisi. In questo modo si ottengono le *feature* correnti per la partita presa in esame e non quelle generiche di tutto il *timeframe* analizzato.

Questo idealmente dovrebbe portare ad una accuracy migliore grazie alla maggiore correttezza delle *feature* in relazione ad ogni partita.

6.2 Risultati ottenuti

In seguito a questa nuova creazione di *feature* è stato riscontrato un peggioramento sostanziale della precisione dei classificatori.

Infatti con nessuno è stato possibile raggiungere nuovamente il 70% di *accuracy* ma nel migliore dei casi si è arrivati ad un 60% scarso. In alcune simulazioni l'accuracy è scesa addirittura al di sotto del 50% - peggio della *random chance*.

Questa evoluzione del progetto peggiora notevolmente le capacità di previsione e non rappresenta quindi una modifica utile per gli obiettivi posti.

7. Vincitore Australian Open

7.1 Dataset di partenza

Per avere un elenco delle partite del torneo è stato usato il dataset relativo all'anno 2020 preso dal sito Tennis-Data.co.uk.
Questo al suo interno ha anche i risultati effettivi delle partite che potranno tornare utili per confrontare i risultati simulati e quindi calcolare l'accuracy all'interno del torneo.
Per quanto riguarda la simulazione, questi dati sono stati però ignorati.

Per ottenere le *feature* è stato sufficiente unire il dataset dei giocatori che al suo interno ha già tutte le *feature* relative al frame temporale 2017-2019.

Per semplificare le operazioni all'interno del torneo è bastato riordinare le partite come visibile sul sito ausopen.com.

7.2 Scelta classificatore

In base alle analisi effettuate, il classificatore che si attesta con maggiore costanza al di sopra del 70% di accuracy risulta essere il k-Nearest-Neighbor utilizzato con uno *scaler* per i dati di tipo MinMax.

È stato dunque usato per prevedere le partite del torneo facendo il fit sull'intero dataset di partite 2017-2019.

7.3 Risultati ottenuti

Simulando ogni round del torneo si è costruito così lo schema dei risultati.

Nella quasi totalità dei casi si ottiene Nadal come vincitore di una ipotetica finale contro Djokovic oppure Federer. Queste finali risultano credibili poiché presentano sempre gli atleti che sulla carta sono tra i più forti del torneo.

Anche nei primi round i risultati sono sempre realistici e non si sono notate situazioni paradossali.

7.4 Confronto con i risultati reali

Confrontando i risultati previsti dal sistema con quelli reali delle prime 64 partite, si ottiene una accuracy del 75-80% che risulta essere molto più alta di quanto ci si potesse aspettare con i dataset iniziali di *train* e test.

Il vincitore del torneo reale è stato Djokovic che nel sistema simulato invece perde quasi sempre in finale. Questo è dovuto nella maggior parte dei casi ad un ipotetico scontro diretto con Nadal che nella realtà ha però perso ai quarti di finale contro Thiem.

Nel complesso ci si ritiene soddisfatti dei risultati ottenuti anche se l'accuracy non è stata sufficiente per prevedere sempre correttamente il vincitore del torneo.