Università degli studi di
Milano-Bicocca

Decision Models

Progetto Finale

Fanta NBA

Autori:
Paolo Merola - 834098 - p.merola@campus.unimib.it
David Govi - 833653 - d.govi@campus.unimib.it

July 12, 2018

Sommario

L'impatto mediatico degli sport più popolari suscita da tempo in milioni di fan assidui il desiderio di improvvisarsi General Manager delle proprie squadre del cuore. Le fanta competition affollano il meta mercato sportivo, diversificandosi per disciplina, modalità, montepremi. Lo scopo rimane però sempre quello di assemblare, attenendosi ad una serie di constraints, la squadra più competitiva possibile.
Il basket NBA è enormemente popolare negli Stati Uniti ed è anche il più statisticamente tracciabile tra gli sport in questione, e pertanto quello con la maggiore offerta di competizioni manageriali. All'interno di questo panorama, DraftKings propone challenge dove si compete per diversi montepremi creando giornalmente la propria squadra.
Ci siamo chiesti come fosse possibile automatizzare il meccanismo di draft e comporre la miglior squadra possibile sulla base di proiezioni e dati strutturati, al fine di investigare quanto fosse possibile prescindere da particolari conoscenze di dominio nel rendersi competitivi alla pari di esperti di settore.

1 Introduzione

Gran parte degli sport di squadra più seguiti hanno una variante *fantasy*. Un fantasy sport è un'attività nella quale i partecipanti sono tenuti a creare formazioni scegliendo i giocatori all'interno di un parco di sportivi professionisti reali. La squadra così formata viene poi valutata sulla base delle prestazioni che i suoi membri ottengono nella realtà, indipendentemente gli uni dagli altri.
L'idea alla base dei fantasy sports è nota fin dalla fine della Seconda Guerra Mondiale, ma il primo esempio di partita di un fantasy risale alla fine degli anni '50 ad opera dell'imprenditore di Oakland Wilfred "Bill" Winkenbach, parziale proprietario della squadra Oakland Raiders; evidentemente, l'idea lo divertì al punto da fargli ideare una lega per il fantasy golf alla fine degli anni '50. Anche altri ebbero la stessa idea, nel 1960 si formò a Boston la prima lega di fantasy baseball, ad opera del professore di Harvard William Gamson.
Da allora i fantasy sports sono costantemente cresciuti in popolarità, numero di tornei e ricchezza dei premi; già in Italia il fantacalcio conta centinaia di migliaia di partecipanti e tornei che offrono premi fino a decine di migliaia di euro.

Negli U.S.A. è soprattutto il fantasy basket, in particolare il fanta NBA, a far da padrone, con milioni di partecipanti ed una ricca varietà di tornei, con regole che possono variare dall'elementare al machiavellico. Il torneo oggetto di questo studio, ospitato dal sito *draftkings.com*, propone regole relativamente semplici: per ogni giornata di gioco, il torneo fornisce un insieme di giocatori selezionabili, ciascuno con associati un costo e uno/due rouli ricoperti all'interno della squadra. Ogni partecipante al torneo ha un budget di 50000 crediti per selezionare esattamente 8 giocatori che ricoprano tutti i ruoli. Ma, a prescindere dalle regole, ognuno dei milioni di partecipanti si trova ad affrontare lo stesso problema: il draft dei giocatori.
In gergo NBA il draft è, letteralmente, la "pésca", ovvero la selezione dei giocatori. Normalmente, un giocatore di fantasy NBA seleziona la propria squadra in base alla propria conoscenza dei giocatori, in particolare formando le proprie personali previsioni sulle prestazioni che il giocatore avrà nel futuro.
Lo scopo di questo lavoro è di verificare se sia possibile e fattibile giocare a fantasy NBA in modo completamente automatico ed ottenere buoni risultati.

2 Dataset

La Kaggle competition *Daily Fantasy Basketball - DraftKings NBA* mette a disposizione dati provenienti da *draftkings.com*. I dati trattano giornate di gioco realmente avvenute durante la stagione precedente. Viene fornito un CSV per ognuna delle 18 giornate di gioco prese in considerazione. Ognuno di questi presenta informazioni sull'intero pool di giocatori a disposizione. Per ogni giocatore si conosce:

- *Name*: nome e cognome del giocatore;
- *Position*: ruolo/i del giocatore;
- *Salary*: costo del giocatore in crediti;
- *GameInfo*: informazioni sulla partita in cui appare il giocatore;
- *AvgPointsPerGame*: media fanta punti totalizzati;
- *Team*: squadra di appartenenza del giocatore;
- *Opponent*: squadra avversaria;

- *Projection*: punteggio atteso nella partita;

Con l'eccezione di *Projection*, tutte queste informazioni sono a disposizione di qualunque utente che abbia accesso alla dashboard di DraftKings durante la fase di selezione dei giocatori. *Projection* è invece un valore stimato dall'owner della competition, coadiuvato da altri cosiddetti "fanta nerd", e del quale non sono noti né i criteri di valutazione né quindi la validità come previsione.

3 Approccio Metodologico

Il problema da risolvere è consistito nella scelta della squadra che avrebbe realizzato il miglior punteggio. Un utente umano effettua questa scelta utilizzando le sue personali previsioni sulle prestazioni di ogni giocatore. Per gli algoritmi decisionali qui implementati, tale previsione viene quantificata nella colonna *Projection*. Per tanto, almeno in questa fase è stata implicitamente accettata la validità di tale valore.
Come accennato in precedenza, la formazione di una squadra è sottoposta a 3 vincoli stabiliti dal gestore del torneo. Essi sono:

- numero di giocatori esattamente pari ad 8;
- il costo complessivo dei giocatori scelti deve essere minore o uguale a 50°000 crediti;
- i giocatori devono ricoprire tutte le posizioni stabilite dal gestore del torneo come riportato nella *Tabella 1*.

Tutte le squadre che non soddisfino i 3 vincoli devono essere scartate. Questo si è rivelato di esecuzione non banale. Anzichè effettuare l'operazione esplicitamente, si è deciso di attribuire valore nullo a queste soluzioni. In tal modo però, la funzione che associa ad ogni soluzione il suo valore, chiamata "di valutazione", è risultata molto più complessa di quanto ci si sarebbe potuti attendere intuitivamente. In particolare è diventata altamente discontinua. Tuttavia, questa scelta è riuscita nell'intento di porre le soluzioni non valide nella condizione di essere scartate da qualunque algoritmo che utilizzi il valore di una squadra come parametro decisionale.

Posizione	Ruoli che possono coprire quella posizione
PG	PG
SG	SG
SF	SF
PF	PF
C	C
G	PG, SG
F	SF, PF
Util	PG, SG, SF, PF, C

Tabella 1: Posizioni da ricoprire nella squadra e ruoli che possono ricoprirle

Si sono utilizzati due approcci differenti al problema: combinatorio e di ottimizzazione.

3.1 Approccio Combinatorio

In generale, un approccio combinatorio consiste nel cercare la combinazione di elementi di maggior valore che risolva il problema; ovviamente, in questo studio l'approccio ha previsto la ricerca della combinazione di giocatori valida che ottenesse il valore massimo in termini di fantapunti.
La metaeuristica dell'algoritmo Genetico risolve il problema combinatorio emulando l'evoluzione di una popolazione sottoposta a selezione naturale. Più nello specifico, l'algoritmo parte da una popolazione iniziale costituita da cromosomi, ciascuno dei quali è una combinazione di geni elementari che costituiscono gli elementi da combinare in modo ottimale per risolvere il problema. Ogni cromosoma viene valutato e solo una percentuale della popolazione, costituita dai più adatti, sopravvive per andare a formare una nuova popolazione mescolando i geni sopravvissuti.
Nel problema della squadra di fantasy NBA, i geni sono i giocatori ed i cromosomi sono le squadre, che risultano essere liste di valori binari che indichino presenza/assenza del giocatore corrispondente all'interno del pool proposto.
L'algoritmo effettua i seguenti passaggi:

1. considera una popolazione iniziale, che al primo passo sarà costituita da un sottoinsieme casuale di tutte le possibili squadre, mentre ai passi successivi sarà la popolazione ottenuta al passaggio precedente;

2. valuta tutte le soluzioni della popolazione utilizzando la funzione di valutazione, poi conserva solo un certo numero di squadre che abbiano i valori più alti, scartando tutte le altre;

3. mescola tra loro i giocatori delle squadre selezionate al passaggio precedente, creando una nuova popolazione;

4. ripete dal passaggio 1 sulla nuova popolazione creata;

L'algoritmo opera quindi con l'assunzione che mescolando i geni delle migliori soluzioni ottenute si ottengano soluzioni migliori delle precedenti.

Di seguito i parametri da inserire nell'algoritmo:

- *numerosità della popolazione*: una popolazione più numerosa garantisce maggiori possibilità di trovare rapidamente una soluzione ottimale, ma aumenta più che polinomialmente i tempi di esecuzione;

- *funzione di valutazione*: l'algoritmo ha bisogno di valutare ciascuna soluzione per decidere come effettuare la selezione. Nel problema del fantasy NBA, tale funzione ha assunto anche il ruolo di controllore della validità di ogni squadra, assegnando valore nullo ad ogni squadra non valida.

- *elitismo*: la percentuale di soluzioni che sopravvivono ad ogni passaggio e che vengono utilizzate per creare la successiva generazione. Come per le dimensioni della popolazione, una percentuale maggiore di sopravvissuti garantisce maggiore possibilità di trovare una soluzione ottimale a discapito dei tempi di esecuzione;

- *mutation chance*: probabilità che due soluzioni mescolino i propri geni. Questo parametro non ha un trade-off ma occorre effettuare tuning per individuarne il valore migliore;

Nella ricerca della squadra migliore, i valori di elitismo e possibilità di mutazione sono stati entrambi settate a 0.1, valore che in letteratura è considerato un buono standard per la maggior parte dei problemi. La dimensione ottimale della popolazione invece è stata più difficile da ricavare: il valore di default di 100 si è rivelato troppo piccolo per ottenere soluzioni soddisfacenti, mentre una dimensione di 500 ha reso l'algoritmo troppo lento. Alla fine si è utilizzato 300, che rappresenta un buon compromesso tra le due esigenze.

3.2 Approccio di Ottimizzazione

Un approccio di ottimizzazione consiste nel massimizzare la funzione di valutazione e ottenere la soluzione che realizzi questo massimo. Dal momento che tale funzione risulta essere altamente discontinua, non è stato possibile implementare metodi di analisi differenziale. Piuttosto, è stato selezionato un algoritmo conosciuto in letteratura come Hill-climbing. L'algoritmo parte da un sottoinsieme di possibili soluzioni e inizia col considerarne una, poi effettua un incremento su di essa e valuta la nuova soluzione: se il valore ottenuto è maggiore, viene effettuato un nuovo incremento, e così via fino a che non è possibile trovare alcun miglioramento.
Questo algoritmo è noto per cadere facilmente in massimi locali, ovvero per essere troppo legato al sottoinsieme di soluzioni fornitogli inizialmente. Si è ovviato a questo difetto effettuando, per ogni giornata, 100 ottimizzazioni ciascuna con sottoinsiemi casuali e considerando alla fine la soluzione migliore trovata tra tutte le implementazioni.

3.3 Problemi e Tempi di Esecuzione

Il principale problema incontrato durante l'implementazione degli algoritmi su R è stata la lunghezza dei tempi di esecuzione. Per l'algoritmo Genetico, tali tempi possono variare da 10 minuti ad oltre 40 per ogni giornata, rendendo difficile un approccio di trial-and-error. Curiosamente, le giornate che hanno richiesto più di 40 minuti di esecuzione sono le numero 4 e 16, contenenti i pool di giocatori meno numerosi, rispettivamente 36 e 33; giornate come la numero 7, con i suoi 190 giocatori, hanno richiesto intorno ai 20 minuti. Evidentemente la lentezza dell'algoritmo è dovuta almeno parzialmente alla presenza di diverse soluzioni locali e/o a squadre dello stesso valore; per risolvere queste situazioni l'algoritmo deve aumentare notevolmente il numero di iterazioni, con conseguente dilatazione dei tempi di esecuzione.
L'algoritmo Hill-climbing ha invece posto un problema differente. L'algoritmo in sè è estremamente veloce, ma per essere efficiente deve essere iterato, con ovvio allungamento dei tempi di esecuzione. Il valore di 100 iterazioni è stato scelto come il migliore dopo un tuning che ha considerato valori da 10 a 10000 iterazioni: con 100 iterazioni il tempo di elaborazione è variato dagli 8 ai 12 minuti, mentre con 10000 ha richiesto circa 4 ore per elaborare una singola giornata, portando il tempo necessario per la conclusione ad oltre 3 giorni.

4 Prime Valutazioni e Considerazioni

Come già detto, le soluzioni ottimali ottenute dai due algoritmi sono calcolate sulla base della *Projection*. È stato quindi necessario ottenere i punteggi effettivi ottenuti dai giocatori durante le varie giornate per poter verificare il rendimento delle fanta squadre ottenute.
Per far questo è stato fatto scraping su *NBA.com*, raccogliendo dati dai tabellini delle varie partite del campionato NBA 2017/2018. Su questi dati statistici sono stati applicati i moltiplicatori previsti dal regolamento di DraftKings, fino ad ottenere una nuova feature denominata *Actual_fpts*. La somma di questi punteggi per tutte e 18 le giornate permette di operare un primo confronto tra i due algoritmi e paragonarli a loro volta con i risultati ottenuti dai due esperti di dominio chiamati in causa.

	Punti_attesi	Punti_ottenuti
Giornata 1	272.96	209.5
Giornata 2	237.03	179.5
Giornata 3	267.27	256.75
Giornata 4	258.28	248.5
Giornata 5	257.81	255.5
Giornata 6	293.25	279
Giornata 7	258.23	285.25
Giornata 8	235.51	249.75
Giornata 9	259.3	294.75
Giornata 10	263.99	265.25
Giornata 11	266.78	247.5
Giornata 12	265.16	283.75
Giornata 13	271.91	260
Giornata 14	243.5	249.75
Giornata 15	253.69	236.75
Giornata 16	257	250.75
Giornata 17	284.52	209
Giornata 18	254.6	233.75

Figura 1: Risultati dell'algoritmo Genetico

	Punti attesi	Punti effettivi
Giornata 1	176.46	239.25
Giornata 2	184.41	271
Giornata 3	218.01	264.75
Giornata 4	225.02	275.5
Giornata 5	209.34	262.25
Giornata 6	232.81	254
Giornata 7	246.3	269.25
Giornata 8	233.52	285
Giornata 9	246.2	287.25
Giornata 10	245.29	276.25
Giornata 11	228.53	242.25
Giornata 12	223.94	295.75
Giornata 13	224.53	290.75
Giornata 14	192.4	251.5
Giornata 15	218	253.5
Giornata 16	225.4	276.5
Giornata 17	247.79	295.5
Giornata 18	204.9	276

Figura 2: Risultati dell'algoritmo Hill Climbing

Giocatori	Risultati ottenuti
Hill Climbing (punti effettivi)	4866,25
Genetico (punti attesi)	4700,79
Genetico (punti effettivi)	4495
Esperto 1	4375,65
Esperto 2	4282,5
Hill Climbing (punti attesi)	3982,85

Tabella 2: Risultati complessivi su tutte le giornate ottenuti dagli algoritmi a paragone con gli esperti umani

Dalla *Tabella 2* si osserva immediatamente come entrambi gli algoritmi ottengano già risultati migliori degli esperti umani sul piano del punteggio. Nello specifico, l'algoritmo Hill Climbing ha effettivamente battuto anche i risultati dell'algoritmo Genetico. È anche interessante osservare, da *Figura 1* e *Figura 2*, quanto spesso i risultati effettivi siano distanti dalle proiezioni. Come già detto, i criteri con cui queste ultime siano state stimate non sono noti, e non è proprio di questo lavoro andare a stimarne di nuove tramite un'analisi approfondita.
Per quanto in ogni caso una certa divergenza fosse attesa, andando ad esaminare i valori giornata per giornata si osservano casi in cui questa è più marcata. La ragionevole spiegazione è che, per quanto eccelsi professionisti, i giocatori NBA sono pur sempre umani. Nella fattispecie, questi atleti giocano oltre tre partite a settimana, legittimando come questi siano soggetti a variazioni di rendimento, qualunque sia la performance che ci si possa aspettare da loro in un dato incontro. Un'altro punto è costituito dal fatto che alcuni giocatori siano naturalmente più incostanti e/o imprevedibili di altri. Su questa variabilità intrinseca dell'essere umano inciampano necessariamente sia algoritmi che esperti umani. Scopo ulteriore di questo lavoro è stato quindi implementare dei metodi ulteriori che permettessero di aggirare questi limiti concettuali.

5 Feature Creation ed Ulteriori Implementazioni

Sono state sviluppate alcune misure per cercare di controllare l'impatto dell'imprevedibilità delle previsioni; in particolare sono state sviluppate due nuove feature:

- *Consistency*: varianza dei punteggi effettivi realizzati dal giocatore durante la stagione in corso. Definisce la consistenza del giocatore, è indice della sua affidibilità in termini di continuità di rendimento.

- *Unpredictability*: distanza tra media fanta punti del giocatore e la proiezione disponibile su una data partita. Definisce un coefficiente di rischio della proiezione.

Queste due feature sono state prodotte per ogni giocatore, aggiornate giornata per giornata. Entrambe rappresentano un discreto passo avanti da un

punto di vista di contestualizzazione rispetto alle informazioni messe a disposizione da DraftKings.
Per poterne fare un utilizzo concreto, entrambe le metriche sono state standardizzate in un range [0,1]. L'idea è stata quella di utilizzare entrambe le feature per scalare i valori delle previsione esistenti per poi andare successivamente a verificare l'impatto di un'azione di questo tipo in termini di punti effettivi ottenuti da squadre selezionate sulla base di questi nuovi valori.
Non potendo conoscere a priori la rilevanza che queste metriche avrebbero avuto sul computo dei punti effettivi si è proceduto assegnando loro combinazioni di pesi diversi in modo da eseguire una sorta di tuning per gli algoritmi precedentemente implementati. Questo è stato fatto prendendo i valori $1 - \textit{Consistency}$ e $1 - \textit{Unpredicatability}$ e moltiplicandoli per i valori originali di *Projection* all'interno di una media ponderata. Per semplicità e mancanza di adeguato potere computazionale si sono scelte come pesi tre coppie di valori: (30,70),(50,50),(70,30).
Come ulteriore tentativo di controllare l'incertezza contenuta nelle predizioni si è tentato un approccio ancora differente, ponendo due ipotesi arbitrarie:

- il valore dei punti effettivamente totalizzati dal giocatore, seppur differente dalla proiezione, si trovi comunque nell'intervallo centrato sul valore della proiezione e di ampiezza sufficientemente piccola;
- la distribuzione di probabilità relativa a tale valore sia una normale.

Si è quindi costruita una nuova proiezione come valore casuale generato da una distribuzione normale di media pari al valore della proiezione e di varianza pari alla *Consistency*. Si è scelta la *Consistency* come varianza in quanto si tratta di un valore costruito per misurare la variabilità nel comportamento del giocatore. Seppur impropriamente, ci si riferirà alla metrica modificata in questo modo come *metrica normalizzata*
In tutti i 4 casi si sono ottenuti valori di *Projection* ridimensionati sui quali si sono calcolate nuovamente le soluzioni ottimali con gli algoritmi precedentemente presentati.

6 Risultati Finali e Valutazione

Giocatori	Risultati ottenuti
Hill Climbing	4866,25
Hill Climbing 30_70	4695,75
Genetico norm	4525
Genetico 30_70	4500,25
Genetico	4495
Esperto 1	4375,65
Esperto 2	4282,5
Genetico 50_50	4203,25
Genetico 70_30	4157,75

Tabella 3: Risultati complessivi su tutte le giornate ottenuti dagli esperti umani e dagli algoritmi con le proiezioni modificate

La *Tabella 3* riporta una classifica complessiva dei risultati ottenuti dagli esperti di dominio e dagli algoritmi. Si noti come l'algoritmo Hill Climbing non abbia fornito risultati con metriche normalizzate e con pesi (50,50) e (70,30). Il motivo è probabilmente il seguente: per costruzione della funzione di valutazione, le soluzioni non valide hanno valore nullo e sono molto più numerose di quelle con valore non nullo. siccome Hill Climbing rimane facilmente confinato in un intorno piccolo della soluzione di partenza, nel caso in cui inizi la propria computazione troppo lontano da una qualsiasi soluzione valida, allora rischia di non individuarne mai una.
Dopo l'aggiunta delle 4 proiezioni modificate si sono misurati nuovamente i tempi di esecuzione per singola giornata ma complessivi su tutte le 4 implementazioni dei due algoritmi. L'algoritmo Genetico ha richiesto di gran lunga il tempo maggiore, con tempi dai 40 minuti ad oltre 2 ore; di nuovo, le giornate che hanno richiesto maggior tempo sono state le numero 4 e 16, ad ulteriore conferma dell'ipotesi sulla presenza di soluzioni di egual valore in queste giornate. Hill Climbing invece ha richiesto dai 2 agli 8 minuti ma senza un legame evidente tra dimensionalità e tempo di esecuzione.
Numeri alla mano possiamo affermare che l'algoritmo Genetico si sia comportato al meglio con proiezioni normalizzate, mentre Hill Climbing abbia ottenuto dei risultati migliori con la sua versione originale.

7 Conclusioni e Lavori Futuri

Dai dati raccolti si può affermare con relativa sicurezza che sia in effetti possibile giocare a fantasy NBA in modo completamente automatico con buoni, se non ottimi, risultati. Entrambi gli algoritmi hanno infatti ottenuto punteggi degni di nota, superando nettamente gli utenti umani.
È chiaro come la validità delle soluzioni ottenibili sia strettamente dipendente dalla qualità delle previsioni ottenibili sul rendimento dei giocatori. Se già implementare degli indici per la loro contestualizzazione ha aiutato a ottenere risultati migliori nella formazione automatica delle squadre, sarebbe sicuramente interessante implementare gli algoritmi decisionali di cui sopra sulla base di previsioni prodotte tramite attente analisi e approfondite ricerche di dominio.
Infine, è comunque bene ricordare come, per quanto approfondita e curata un'analisi, piuttosto che una previsione, di questo tipo possa essere, i suoi risultati saranno sempre influenzati dall'imprevedibilità e dalla casualità delle performance degli atleti, componenti che del resto rendono magico lo sport.