REPORT WEB INTELLIGENCE

Studente: Scodeller Giovanni 864906

Corso: WEB INTELLIGENCE [CT0428]

Professore: Lucchese Claudio

Task

Il task del progetto è prevedere chi vincerà l'Open Australia 2020 simulando il torneo creando i singoli incontri, dal primo turno alla finale, per poi dire il nome del vincitore.

Ciò nonostante l'obbiettivo del progetto è stato leggermente modificato, invece di prevedere chi vincerà l'Open Australia 2020, è stato creata un'analisi e dei modelli per simulare e prevedere il vincitore dell'Open del 2019. Questa scelta è stata effettuata perché al momento dello sviluppo non era possibile sapere i giocatori che avrebbero partecipato, tuttavia per come sono state create alcune funzioni è possibile simulare anche il torneo del 2020 appena i dati saranno resi disponibili.

Introduzione

I dati utilizzati per il progetto erano facilmente reperibili scaricandoli dall'archivio Tennis-Data dove era possibile trovare le partite degli ATP dal 2001 al 2019 con descritte tutte le informazioni necessarie.

Early Data Analysis (EDA)

Per una prima esplorazione dei dati prediamo in esempio l'anno sportivo 2017.

Notiamo che ogni entri del Dataset corrisponde ad una determinata partita giocata in quell'anno, in totale nel 2017 sono state disputate 2633 partite, per quanto riguarda le colonne notiamo che contengono vari dati che aiutano a descrivere in maniera più specifica gli incontri.

Le feature del dataset possono essere divise in due categorie:

1. Quelle che descrivono i dati del torneo.

   Dove possiamo osservare le colonne:

   - Date, ATP, Location, Tournament, Series, Court, Surface, Best Of

     Queste colonne descrivono principalmente tutte le caratteristiche di dove un determinato match è stato svolto ed il tipo di campo in cui si ha giocato.

   - Winner, Loser, WRank, LRank, WPts, LPts

     Vengono descritte le informazioni relative ai giocatori che hanno giocato il match, i punti che hanno e la loro posizione nel ranking, è importante specificare che *viene già descritto chi ha vinto e chi ha perso*.

   - W1, L1, W2, L2, W3, L3, W4, L4, W5, L5, Wsets, Lsets, Comment

     Vengono descritti il numero di partite vinte per ogni set ed il numero totale di set ottenuti dal vincitore ed il perdente, inoltre viene descritto se il match è stato concluso regolarmente o per qualche altra ragione.

2. E quelle che descrivono tutte le quotazioni dei principali siti di scommesse.

Prime considerazioni

Analizzando le colonne possiamo effettuare delle prime considerazioni prima di iniziare a lavorare:

- I dati non contengono valori difficili da lavorare, infatti sono stringhe o variabili numeriche.
- Per come i dati sono salvati non esiste una feature da prevedere poiché l'informazione che uno dei due giocatori ha vinto o perso viene descritta da due feature differenti.
- Nel dataset potrebbero esserci dei dati mancanti per errori di registrazione o non sono stati volutamente inseriti (vedi set successivi se il match si è già concluso), bisognerà intervenire laddove i dati manchino inserendo i valori più opportuni per rendere il dataset coerente.

Aggiunta della variabile "target"

Per affrontare il task di predizione del risultato di una partita di tennis abbiamo bisogno di una variabile che ci dia un'informazione su chi dei due giocatori è il vincitore.

Non esistendo già questa informazione, poiché nei dati forniti viene implicitamente descritta nelle colonne "Winner "e "Loser", abbiamo bisogno di crearla. La nuova feature aggiunta denominata "**target**" viene costruita nel seguente modo: "Contiene il valore 0 se in quel match vince il primo giocatore descritto, 1 se vince il secondo".

In questo modo verrà aggiunta una colonna che attualmente conterrà il valore 0 per tutte le partite dell'anno poiché il primo giocatore descritto è **sempre** il vincitore.

Per rendere il dataset più coerente è stata creata la funzione ***generalFormatting*** utilizzata per cambiare il nome delle feature "Winner" e "Loser" in "Player1" e "Player2", inoltre sono state applicati altri metodi per formattare i dati. In particolare le tecniche utilizzate sono le seguenti:

- le colonne che contengono valori nulli vengono sistemate a criterio, se ad esempio un giocatore non ha punti in classifica entra con il minor numero di punti,
- le colonne che contengono delle variabili categoriali, come "Court", "Surface" e "Best of" viene applicato il metodo del OneHotEncoding che crea tante feature tante quante sono le classi di quella feature assegnando valori 0 o 1,

  Esempio: "Court" contiene "Indoor" e "Outdoor", verranno create due nuove feature "Court_Indoor" e "Court_Outdoor" ed eliminata quella di partenza
- altre variabili come "Series" viene applicata la tecnica del Label Enconding, ovvero viene data un'etichetta numerica ad ogni classe della feature.

Applicazione primo modello

Dopo aver effettuato una prima lavorazione dei dati è possibile applicare un modello di previsione banale che servirà ad aver una soglia di accuratezza minima per i modelli successivi. Semplicemente la nostra previsione si baserà sulla posizione nel ranking, se "P1_Rank" è più alto [1] in classifica

viene predetto 0, altrimenti 1.

Prevedendo che vinca sempre il giocatore più forte otteniamo un'accuratezza di $\approx 0.65\%$, più della metà degli incontri svolti nell'anno.

Creazione di nuove feature

A questo punto dell'analisi sono state create delle nuove feature per aggiungere delle informazioni al dataset.

Swap delle righe e feature "target"

Aggiungere l'informazione di chi vince e chi perde le partite purtroppo non basta per poter creare un task di classificazione perché attualmente i dati che abbiamo contengono una sola classe nella feature "target". Viene creata la funzione ***randomSwap*** per poter scegliere delle righe casuali ed invertire tutte le variabili di "Player1" e "Player2" in modo da far comparire la classe 1 nella feature "target" che ottiene il significato di "Player2 vince".

Rateo di vittoria fra giocatori

Utilizzando la funzione ***addRateo*** verrà creata una matrice di giocatori X giocatori, dove le righe e le colonne sono l'unione degli atleti che hanno giocato almeno una partita nell'anno precedente e nell'anno del torneo che si vuole predire. In questo modo grazie alle partite dell'anno precedente è possibile vedere quanto un giocatore è stato più forte o più scarso rispetto a tutti gli altri giocatori con cui a giocato. Bisogna far notare l'utilizzo di una tecnica importante che viene applicata in questa matrice di giocatori X giocatori, lo *smoothing* dei dati, consiste nell'aggiungere del rumore in modo tale da non avere eventi troppo rari o specifici, viene proposto un esempio: Se "Thompson J." contro "Dimitrov G." ha vinto 2 partite su 2 giocate nell'anno precedente, avrà il 100% di vittoria su "Dimitrov G." e "Thompson J." avrà lo 0% di vittoria e questo non è del tutto

corretto. Se però aggiungiamo una vittoria ad entrambi i giocatori otterremo che "Thompson J." ha vinto 3 partite e "Dimitrov G." 1 partita, in questo modo otterremmo un rapporto 66% / 33% sulla probabilità di vittoria rendendo la feature più consistente. Inoltre nel caso due giocatori non abbiano mai giocato tra di loro hanno entrambi un rateo di vittoria e sconfitta del 50%.

Rateo di vittoria dei giocatori in un campo specifico

Un'altra informazione che viene aggiunta è la bravura di un giocatore in un determinato tipo di campo, ne esistono di 3 categorie "Clay","Hard" e "Grass". Vengono raccolte le performance dei giocatori utilizzando i dataset dal 2001 all'anno precedente del torneo, in questo modo è possibile vedere se hanno una predisposizione a vincere su un determinato tipo di campo specifico. Anche qui viene applicato lo *smoothing* per i giocatori che non hanno mai svolto una gara in un determinato tipo di campo o hanno iniziato a partecipare ai tornei nell'anno che viene analizzato. Tutto questo appena descritto viene effettuato dalla funzione ***addFieldWR***.

Win Streak e Lose Streak dei giocatori

Tramite la funzione ***addPrevWin*** vengono aggiunte le informazioni di quante partite un giocatore sta vincendo di fila, ma anche quante ne sta perdendo, si pensa che un giocatore in forma è più propenso a vincere anche la partita successiva ed analogamente si può ragionare per un giocatore che non essendo in forma è più propenso a perdere.

Modelli di Supervised Learning

Prima dell'applicazione di qualsiasi modello sono state aggiunte le nuove feature precedentemente descritte ed inoltre vengono scartate le colonne che contengono informazioni sulle scommesse poiché quando andremo a simulare il torneo per determinare il vincitore dell'Open Australia non saranno dei dati disponibili dopo il primo Round, è stato scelto di rimuovere anche i nomi dei giocatori ed il nome del torneo. Inoltre i dati verranno divisi in train e test set, analogamente utilizzati per l'allenamento del modello ed il test per vedere i risultati ottenuti.

Albero di decisione

Proviamo ad applicare un albero di decisione sui nuovi dati a cui sono state aggiunte le feature, è stato deciso di utilizzare come parametro di terminazione il raggiungimento di un determinato numero di foglie e notiamo che il miglior modello ha un' accuratezza del $\approx 80\%$, un sostanziale aumento di precisione sul test set del circa $\approx 15\%$. Questo ci permette di dire che rispetto all'utilizzo delle feature di base, l'aggiunta di quelle create aiutino ad aumentare la descrizione dei dati.

Ensemble Methods

A questo punto possiamo provare ad utilizzare dei metodi insiemistici applicati all'albero di decisione per vedere se possiamo migliorarne ulteriormente la precisione.

La prima tecnica che proviamo ad applicare è il ***BAGGING*** essa ci permette di ridurre la varianza del nostro ablero e notiamo che nonostante si aumenti il numero di modelli utilizzati otteniamo un miglioramento massimo del $\approx \%3$, arrivando quindi ad un accuracy score di $\approx 83\%$, è stato provato ad applicare anche il metodo di **BOOSTING** notando un guadagno simile a quello precedente.

Random Forest

Come ultimo modello viene provato ad applicare il metodo delle random forest per vedere se riusciamo ad ottenere un miglioramento ulteriore per il nostro modello. Nonostante sia il metodo di previsione più complesso applicato in questa analisi otteniamo solo un aumento di precisione di un ulteriore $\approx 2\%$, quindi utilizzando questo metodo abbiamo un'accuratezza del $\approx 85\%$.

Recursive Feature Elimination (RFE)

A questo punto proviamo a vedere quali sono le N feature più importanti per la costruzione del nostro modello di previsione.

Se ad esempio scegliamo di selezionare le 6 feature più importanti possiamo notare che siano:

- P1_Pts
- P2_Pts
- P1_winningField
- P2_winningField
- P2_winningField
- P2_losingField

Simulazione del torneo

Per la simulazione è stata creata una funzione apposita chiamata ***tournamentSimulation*** essa prende in input il primo turno dell'Open Australia per poterne ricavare i partecipanti, il dataset del 2018 che servirà ad aggiornare il rateo tra due giocatori e come ultimo parametro un modello di decisione allenato che verrà utilizzato per prevedere l'esito degli incontri.

Il torneo viene costruito accoppiando i giocatori due a due per ogni turno del torneo, dopodiché viene utilizzato il modello di previsione per determinare il vincitore dell'incontro ed il perdente viene rimosso dalla lista dei giocatori, questo processo viene effettuato finché non rimarrà un solo giocatore che sarà il vincitore del torneo.

Considerazioni finali

Sono stati utilizzati diversi modelli di previsione in questo assigment e possiamo riassumerli nella seguente tabella:

Modello	Accuracy Score
Modello banale (Più forte vince)	65 %
Albero di Decisione (No feature aggiunte)	68 %
Albero di Decisione (Split = Gini)	80 %
Albero di Decisione (Split = Information Gain)	80 %
Albero di Decisione con Bagging	82 %
Albero di Decisione con Boosting	81 %
Random Forest	85 %
RFE	86 %

Possiamo dire che l'aggiunta delle nuove feature ha sicuramente migliorato la spiegazione dei dati in maniera abbastanza importante ed applicare metodi sempre più complessi aiuta ad avere un'accuratezza maggiore.

Miglioramenti futuri

Per migliorare ulteriormente il modello di previsione si potrebbero creare altre nuove feature utilizzando anche dataset esterni, probabilmente valorizzando di più le caratteristiche personali di un giocatore si potrebbe migliorare ulteriormente la precisione del modello.

---

1. Con alto viene indicato in verità il numero intero più basso. Il giocatore con Rank 1 è più alto in classifica del giocatore↩