



## **UNIVERSITA' DEGLI STUDI DELLA BASILICATA**

Dipartimento di Scienze di Base e Applicate

Corso di Laurea Magistrale in Matematica

### **Tesina di Statistica e Machine Learning**

Analisi Predittiva delle Performance dei Giocatori in Clash Royale: Un'applicazione dei Modelli di Regressione Regolarizzata.

Docente:

Antonella Iuliano

Studente:

Donato Di Lonardo  
Matr. 69617

# Indice

<b>Indice.....</b>	<b>2</b>
<b>Capitolo 1: Introduzione al mondo di Clash Royale.....</b>	<b>4</b>
<b>Capitolo 2: Recupero dei Dati.....</b>	<b>5</b>
2.1. La Strategia di Recupero Dati.....	5
<b>Capitolo 3: Organizzazione e Preparazione dei Dati.....</b>	<b>6</b>
3.1. Struttura dei Dati Grezzi dall'API.....	6
3.2. Operazioni di Pulizia e Trasformazione dei Dati.....	7
3.3: Descrizione del Dataset e Plot.....	8
3.4. Plot delle Variabili Chiave.....	12
3.4.1. Distribuzione dei Trofei.....	12
3.4.2. Distribuzione del Livello di Esperienza.....	13
3.4.3. Distribuzione dei Ruoli nel Clan.....	13
3.4.4. Distribuzione del Costo Medio del Deck.....	14
3.4.5. Top 10 Carte Preferite.....	14
3.4.6. Leghe e Trofei nel Percorso delle Leggende.....	15
3.4.6.1. Distribuzione dei Livelli di Lega nella Stagione Scorsa.....	15
3.4.6.2. Distribuzione dei Migliori Trofei nel Percorso delle Leggende.....	15
<b>Capitolo 4: Preparazione e Preprocessing dei Dati per fare Regressione.....</b>	<b>17</b>
4.1 Gestione dei Valori Mancanti (NA).....	17
4.2 Suddivisione del Dataset (Training e Test Set).....	18
4.3 Verifica delle Assunzioni Preliminari per la Regressione Lineare.....	18
4.3.1 Verifica della Dipendenza Lineare e Ottimizzazione delle Variabili.....	19
4.3.2 Gestione della Multicollinearità Attraverso l'Analisi VIF.....	22
4.3.3 Verifica delle Ipotesi del Modello Lineare Iniziale.....	24
4.3.4 Verifica delle Ipotesi del Modello Lineare con Variabile Risposta Trasformata.	27
4.4 Interpretazione del Modello Lineare Addestrato.....	29
4.4.1 Interpretazione della Significatività dei Coefficienti.....	29
4.5 Valutazione del Modello e Selezione dei Predittori.....	33
4.5.1 Metriche di Adattamento e Significatività Complessiva.....	33
4.5.2 Criteri di Selezione del Modello Basati sull'Informazione (AIC, BIC, Mallows' Cp).....	35
4.5.3 Validazione Incrociata (Cross-Validation) per la Stima dell'Errore Predittivo....	36
<b>Capitolo 5: Selezione del Miglior Sottoinsieme di predittori (Penalizzazione L0).....</b>	<b>38</b>
5.1 Selezione del Sottoinsieme di Variabili (Best Subset Selection con regsubsets).....	39
5.1.1 Metodologia.....	39
5.1.2 Risultati e Interpretazione.....	39
5.2 Selezione Step (Basata su AIC).....	43
5.2.1 Metodologia (step).....	43
5.2.2 Risultati e Interpretazione.....	43
5.3 Considerazioni di selezione.....	44
<b>Capitolo 6: Regressione Penalizzata (Ridge, Lasso, Elastic Net).....</b>	<b>46</b>
6.1 La routine glmnet.....	47
6.2 Regressione Ridge ( $\alpha=0$ ).....	48

6.3 Regressione Lasso ( $\alpha=1$ ).....	51
6.4 Regressione Elastic Net ( $\alpha=0.5$ ).....	53
6.5 Confronto tra i Modelli Penalizzati.....	55
6.6 Conclusioni sui metodi penalizzati.....	57
<b>Capitolo 7: Selezione del Modello Finale e Applicazione.....</b>	<b>59</b>
7.1 Confronto e Valutazione dei Modelli Esplorati.....	59
7.1.1 Modello OLS con Variabile Risposta Trasformata (Capitolo 4).....	59
7.1.2 Modelli con Penalizzazione L0 (Capitolo 5).....	60
7.1.3 Modelli con Penalizzazione L1/L2/Elastic Net (Capitolo 6).....	61
7.3 Valutazione del Modello Finale sul Set di Test Indipendente.....	62
7.4 Conclusioni Finali.....	63

# Capitolo 1: Introduzione al mondo di Clash Royale



Fra i giochi mobile, pochi titoli hanno raggiunto la popolarità e la longevità di Clash Royale. Rilasciato da Supercell nel 2016, questo gioco di strategia ha conquistato milioni di giocatori in tutto il mondo grazie ad un avvincente mix di scontri nell'arena, collezione di carte e duelli PvP (Player versus Player, ovvero "giocatore contro giocatore") frenetici.

Le partite si svolgono in un'arena virtuale dove due avversari si sfidano utilizzando un mazzo composto da otto carte, ciascuna può essere una truppa, un incantesimo o una struttura difensiva. L'obiettivo è distruggere le torri nemiche, in particolare la Torre del Re, prima che l'avversario faccia lo stesso ed alla fine vince il match chi infligge più danni.

Prima di partecipare agli scontri, ogni giocatore organizza un mazzo di otto carte giocabili ed una carta di supporto aggiuntiva (nota anche come truppa della torre). Ogni carta è caratterizzata da un nome, un costo in elisir (la risorsa necessaria per schierarla durante una battaglia), un livello (che il giocatore può e deve migliorare investendo risorse di gioco per aumentarne l'efficacia) e altre proprietà specifiche che ne definiscono il ruolo e le abilità.

Il successo di Clash Royale non risiede solo nella sua semplice ma strategica meccanica di gioco, ma anche nei meccanismi di progressione e nella frequente introduzione di nuove carte, sfide, cosmetici e bilanciamenti che mantengono l'esperienza avvincente e sempre al passo coi tempi. La vasta community di giocatori genera ogni giorno big data relativi a stili di gioco, creazione di mazzi, performance individuali e progressione. Questo patrimonio di informazioni è un'opportunità unica per condurre analisi statistiche e scoprire relazioni intrinseche sul comportamento dei giocatori e sulla dinamica del gioco.

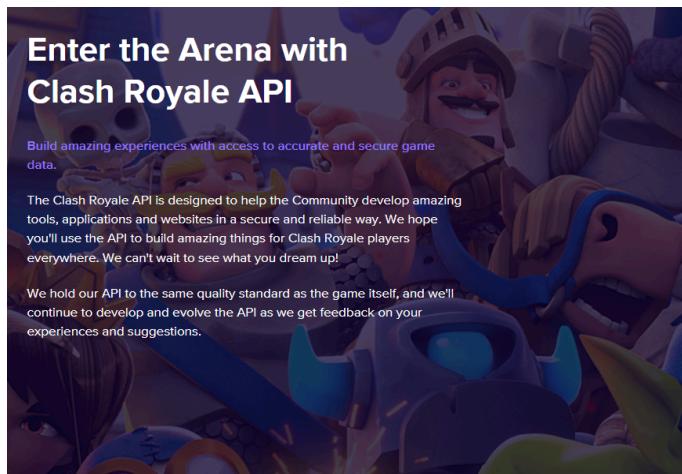


Questa relazione ha l'obiettivo di esplorare proprio questi dati, concentrandosi in particolare sulle variabili chiave che caratterizzano l'avanzamento e le attitudini dei player. Attraverso l'analisi di un vasto campione di utenti, cercheremo di identificare pattern, associazioni e tendenze che possano offrire una comprensione profonda dell'esperienza di gioco e delle strategie adottate da chi scende in campo nelle arene di Clash Royale.

## Capitolo 2: Recupero dei Dati

L'analisi di un ecosistema di gioco così vasto e dinamico come Clash Royale necessita l'accesso a una fonte di big data aggiornata e affidabile. Fortunatamente la casa produttrice del gioco (Supercell) mette a disposizione una API (Application Programming Interface) pubblica dedicata proprio a Clash Royale.

Nell'ambito di Clash Royale, possiamo immaginare l'API come un ponte tra i server del gioco e le applicazioni esterne (come il nostro ambiente di analisi in R), infatti consente un accesso a una vasta gamma di dati relativi al gioco. Questi dati includono informazioni sui giocatori, sui clan, sulle battaglie, sulle carte e sugli eventi in corso.



### 2.1. La Strategia di Recupero Dati



Una caratteristica fondamentale di Clash Royale è che i giocatori si possono organizzare in clan. Questa peculiarità si è rivelata utile per la fase di recupero dati poiché l'API di Clash Royale non offre una funzionalità diretta per scaricare dati relativi ad un insieme massivo di giocatori. L'API permette di ottenere i dati di un singolo giocatore, per farlo, inoltre, è indispensabile conoscerne il tag del giocatore (un codice alfanumerico univoco che lo identifica).

Per superare questa limitazione e raccogliere informazioni su un numero elevato di giocatori, la strategia adottata ha seguito un approccio a più fasi basato proprio sui clan:

- Raccolta dei Tag dei Clan:** Il punto di partenza è stato richiedere all'API un elenco di clan. Per garantire un campione diversificato e rilevante, sono state ottenute informazioni di **960 clan**, ciascuno con un numero di membri compreso tra 10 e 50.
- Estrazione dei Tag dei Giocatori dai Clan:** Successivamente, per ogni clan fornito dall'API, sono state scaricate le informazioni sui suoi membri. Da queste informazioni, sono stati estratti i tag individuali di ciascun membro del clan. Questo processo ha portato alla raccolta di un totale di 12812 tag di giocatori distinti.
- Recupero dei Dati Individuali dei Giocatori:** Infine, per ciascuno dei 12812 tag giocatore raccolti, è stata effettuata richiesta all'API per ottenere tutte le informazioni dettagliate corrispondenti a quel giocatore.

Con questo processo iterativo abbiamo costruito una prima versione del dataset fondamentale per la nostra analisi.

# Capitolo 3: Organizzazione e Preparazione dei Dati

Dopo aver recuperato i dati grezzi dei giocatori dall'API di Clash Royale, il passo successivo è stato quello di organizzarli in un formato più adatto per l'analisi statistica in R. I dati forniti dall'API si presentano in una struttura gerarchica complessa di liste e sotto-liste, ottimale per lo scambio di informazioni tramite API, ma che richiede una trasformazione significativa per essere efficacemente esplorato e manipolato con gli strumenti di analisi dati tradizionali.

## 3.1. Struttura dei Dati Grezzi dall'API

```
Player <--> {
    clan
    legacyTrophyRoadHighScore
    currentDeck
    currentDeckSupportCards
    arena
    role

    wins
    losses
    totalDonations
    leagueStatistics
    cards

    supportCards
    currentFavouriteCard
    badges
    tag
    name
    expLevel
    trophies
    bestTrophies
    donations
    donationsReceived
    achievements
    battleCount
    threeCrownWins
    challengeCardsWon
    challengeMaxWins
    tournamentCardsWon
    tournamentBattleCount
    warDayWins
    clanCardsCollected
    starPoints
    expPoints
    totalExpPoints
    currentPathOfLegendSeasonResult
    lastPathOfLegendSeasonResult
    bestPathOfLegendSeasonResult
    progress
}
```

```
PlayerClan > {...}
integer
PlayerItemList > [...]
PlayerItemList > [...]
Arena > {...}
string
Enum:
    > Array [ 5 ]
    integer
    integer
    integer
PlayerLeagueStatistics > [...]
PlayerItemList <--> [PlayerItemLevel <--> {
    id
    rarity
    count
    level
    starLevel
    evolutionLevel
    used
    name
    JsonLocalizedName > {...}
    maxLevel
    elixirCost
    maxEvolutionLevel
    iconUrls
    > {...}
}]
PlayerItemList > [...]
Item > {...}
PlayerAchievementBadgeList > [...]
string
string
integer
integer
integer
integer
integer
integer
PlayerAchievementProgressList > [...]
integer
integer
integer
integer
integer
integer
integer
integer
integer
PathOfLegendSeasonResult > {...}
PathOfLegendSeasonResult > {...}
PathOfLegendSeasonResult > {...}
> {...}
```

Questa organizzazione in liste e sotto-liste presenta criticità per un'analisi diretta in R:

- **Difficoltà di Accesso:** Proprietà annidate rendono l'estrazione dei dati complessa.
- **Complessità per Operazioni Vettorializzate:** Le operazioni statistiche efficienti in R necessitano di dati in formato tabellare, formato che non è compatibile con la struttura a liste nidificate.

Per comprendere le sfide di questa fase, è utile esaminare la struttura dei dati di un singolo giocatore così come ricevuta dall'API.

L'interfaccia organizza le informazioni in modo che diverse proprietà siano raggruppate in liste nidificate. Ad esempio, il tipo di dato `Player` nel modello dell'API, come illustrato nella documentazione, contiene numerose proprietà come `tag`, `name`, `trophies`, `wins`, e include anche sotto-liste per dettagli più complessi come `currentDeck`, `cards` e `badges`.

La figura a lato mostra un estratto del modello dei dati del Giocatore, evidenziando la natura gerarchica delle informazioni.

Per superare queste sfide, il prossimo passo è stata la trasformazione di questa struttura gerarchica in un formato "piatto" e tabellare, nello specifico un data frame, dove ogni giocatore rappresenta una riga e ogni proprietà diventa una colonna distinta. Questo non solo semplifica l'accesso e la manipolazione dei dati, ma abilita anche l'uso di tecniche di analisi e visualizzazione avanzate.

### 3.2. Operazioni di Pulizia e Trasformazione dei Dati

Il processo di preparazione dei dati è stato eseguito con una serie di passaggi sistematici, implementati in R utilizzando principalmente la funzione `lapply` per iterare su ogni giocatore e manipolarne le proprietà, e il pacchetto `dplyr` per la conversione finale in un formato tabellare ottimizzato. Le principali operazioni eseguite includono:

- **Estrazione e Standardizzazione delle Informazioni sulle Carte:**

- È stato recuperato un elenco completo di tutte le carte disponibili nel gioco dall'API.
- Per ogni giocatore, sono state aggiunte nuove colonne booleane/dummy che indicano se il giocatore utilizza o meno una specifica carta nel suo `currentDeck`.
- È stata calcolata la media del costo in elisir delle carte presenti nel mazzo corrente (`meanCostDeck`).
- Le liste originali `currentDeck` e `currentDeckSupportCards` sono state rimosse una volta estratte le informazioni rilevanti, per appiattire la struttura.



- **Calcolo del Livello Reale delle Carte e Statistiche Correlate:**

- Per ogni giocatore, sono state poi calcolate la media del livello delle carte (`meanLevelCards`) e il conteggio specifico delle carte possedute per i livelli chiave (es. `CardsLevel15`, `CardsLevel14`, `CardsLevel13`).
- È stato anche contato il numero di carte con evoluzione (`CardsEvo`) e il numero totale di carte di supporto possedute (`NumberSupportCards`).
- Le liste originali `cards` e `supportCards` sono state rimosse dopo queste estrazioni.

- **Derivazione di Variabili Temporali e di Progressione:**

- È stata calcolata una variabile `daysSinceRegistration`, basata sulla proprietà `progress` della prima medaglia (`badges`), che indica il numero di giorni trascorsi dall'iscrizione del giocatore. Questa medaglia si sblocca dopo un anno di gioco quindi per i giocatori iscritti da meno di un anno non conosciamo questa variabile che è stata impostata a 180 (circa la mediana di 365).

- Le informazioni dettagliate relative all'andamento del giocatore nel percorso competitivo di Clash Royale sono state estratte e convertite in variabili numeriche separate.
- **Semplificazione e Rimozione di Campi Non Necessari:**
  - La complessa struttura della `currentFavouriteCard` è stata semplificata, mantenendo solo il nome della carta preferita del player.
  - Campi gerarchici o non direttamente utili per l'analisi (come `arena`, `badges`, `achievements`) sono stati rimossi per appiattire ulteriormente la struttura.

Al termine di queste trasformazioni, ogni giocatore è stato rappresentato da una lista "piatta" contenente circa 173 variabili semplici direttamente analizzabili. Questo formato è stato poi convertito in un `tibble` (un tipo di data frame ottimizzato in R) utilizzando `dplyr::bind_rows()`, garantendo la coerenza dei tipi di dati e la robustezza del processo anche in presenza di dati mancanti. Il dataset finale è stato salvato in formato `.parquet` per un'archiviazione efficiente e un rapido caricamento in future sessioni di analisi.

### 3.3: Descrizione del Dataset e Plot

Questo paragrafo ha lo scopo di illustrare in dettaglio ogni variabile presente nel dataset, fornendo il contesto necessario per interpretare correttamente i risultati delle analisi future. Comprendere il significato di ciascuna colonna è fondamentale per trasformare i dati grezzi in modelli significativi. Inoltre, verranno presentate alcune visualizzazioni chiave per mostrare le distribuzioni delle variabili più rilevanti.

Il dataset piatto (file: `player_data.parquet`) è composto da circa 12.812 osservazioni (giocatori) e 174 variabili. Di seguito, una descrizione dettagliata di ciascuna colonna:

- **tag** (Stringa): Una stringa alfanumerica che identifica in modo univoco ciascun giocatore.
- **name** (Stringa): Il nickname scelto dal giocatore al momento dell'iscrizione.
- **expLevel** (Intero): Il livello di esperienza del giocatore, un numero progressivo (da 1 a 70) che aumenta man mano che il giocatore progredisce nel gioco. L'interpretazione di questa variabile come numerica e non fattoriale è mossa da due aspetti fondamentali: essa è la traduzione di una variabile latente continua (i punti esperienza) quindi in quanto discretizzazione di una variabile continua è da considerare numerica (questa sua ridondanza è anche la ragione per cui sarà rimossa da quasi tutti i modelli); essa ha un'elevata relazione lineare con **trophies** quindi considerarla numerica è equivalente a gestirla tramite contrasti polinomiali e fermarsi al primo e più significativo ordine (questo assicura che nei modelli in cui è inserita l'approssimazione introdotta è minima).
- **trophies** (Intero): Il numero di trofei attualmente posseduti dal giocatore in una delle modalità di gioco principali ("Percorso dei Trofei"). I trofei possono essere guadagnati o persi in base all'esito dei match, con un massimo di 9000 trofei.

- **bestTrophies** (Intero): Il numero massimo di trofei che il giocatore ha raggiunto nella modalità "Percorso dei Trofei". Questo valore può essere superiore a **trophies** se il giocatore ha perso trofei dopo aver raggiunto il suo picco.
- **wins** (Intero): Il numero totale di vittorie ottenute dal giocatore dall'iscrizione.
- **losses** (Intero): Il numero totale di sconfitte subite dal giocatore dall'iscrizione.
- **battleCount** (Intero): Il numero totale di match giocati dal giocatore dall'iscrizione.
- **threeCrownWins** (Intero): Il numero di partite vinte annientando completamente l'avversario (distruggendo tutte e tre le torri).
- **challengeCardsWon** (Intero): Il numero totale di carte vinte nelle "Challenge" (sfide con regole speciali, dove si gioca finché non si totalizzano 3 sconfitte o 12 vittorie).
- **challengeMaxWins** (Intero): Il numero massimo di vittorie ottenute in una singola "Challenge" prima di perdere tre volte.
- **tournamentCardsWon** (Intero): Il numero totale di carte vinte nei "Tornei" (modalità di gioco non competitiva basata su un formato a torneo).
- **tournamentBattleCount** (Intero): Il numero totale di battaglie giocate nei "Tornei".
- **role** (Fattore): Il ruolo del giocatore all'interno del proprio clan, con livelli: "member" (membro), "elder" (anziano), "coLeader" (co-capo) o "leader" (capo).
- **donations** (Intero): Il numero di carte donate ai membri del clan nel mese corrente.
- **donationsReceived** (Intero): Il numero di carte ricevute dai membri del clan nel mese corrente.
- **totalDonations** (Intero): Il numero totale di carte che il giocatore ha donato ai membri del clan dall'iscrizione.
- **warDayWins** (Intero): Il numero di partite vinte dal giocatore nelle "Guerre tra Clan", una modalità non competitiva incentrata sulla collaborazione di gruppo.
- **clanCardsCollected** (Intero): Il numero di carte ottenute dal giocatore tramite le donazioni degli altri membri del clan.
- **currentFavouriteCard** (Fattore): Il nome della carta più utilizzata dal giocatore nelle sue ultime partite.
- **starPoints** (Intero): Una valuta speciale che i giocatori possono accumulare e investire in miglioramenti cosmetici delle carte.

- **expPoints** (Intero): I punti esperienza attuali che il giocatore sta accumulando e che verranno "consumati" per salire al successivo **expLevel**.
- **legacyTrophyRoadHighScore** (Intero): Il numero massimo di trofei ottenuti in una vecchia modalità competitiva di gioco (questo valore sarà **NA** per gli utenti più recenti dato che questa modalità non è più disponibile).
- **totalExpPoints** (Intero): I punti esperienza totali accumulati dall'iscrizione (a differenza di **expPoints**, questo valore include anche i punti esperienza già spesi per salire di **expLevel**).
- **Variabili [NomeCarta]** (Fattore - **TRUE/FALSE**): Un set di variabili binarie, una per ogni carta presente nel gioco (es. **Knight**, **Archers**, ... fino a **Goblin Curse**). Il valore della colonna sarà **TRUE** se la carta è inclusa nel mazzo principale attualmente in uso dal giocatore, **FALSE** altrimenti.
- **meanCostDeck** (Numerico): Il costo medio in elisir delle 8 carte presenti nel mazzo principale del giocatore. Questo valore influenza la velocità con cui il giocatore può schierare le truppe, ed è un fattore chiave nella strategia di gioco. Un costo bilanciato è generalmente preferibile, né troppo basso né troppo alto.
- **Variabili [NomeCartaSupporto]** (Fattore - **TRUE/FALSE**): Analogamente alle carte del mazzo, queste sono 4 variabili binarie (**Tower Princess**, **Cannoneer**, **Dagger Duchess**, **Royal Chef**), una per ogni carta di supporto. Il valore sarà **TRUE** se la carta di supporto è attualmente in uso dal giocatore, **FALSE** altrimenti.
- **daysSinceRegistration** (Intero): Il numero di giorni trascorsi dall'iscrizione del giocatore al gioco fino alla data di estrazione dei dati. Per i giocatori con meno di un anno di gioco questa variabile è di default 180.
- **yearsSinceRegistration** (Intero): Il numero di anni trascorsi dall'iscrizione del giocatore al gioco fino alla data di estrazione dei dati. Meno granulare della variabile precedente che spesso viene esclusa in fase di regolarizzazione.
- **is\_new\_player** (Fattore - **TRUE/FALSE**): una variabile binaria che vale true se l'utente gioca a Clash Royale da meno di 365 giorni.
- **cardsOwned** (Intero): Il numero totale di carte uniche che il giocatore ha sbloccato (possiede) nel gioco (va da 8 a 119).
- **meanLevelCards** (Numerico): Il livello medio di tutte le carte sbloccate dal giocatore. Le carte hanno un livello che va da 1 a 15, questi livelli sono standardizzati per permettere confronti equi tra diverse rarità.
- **CardsLevel15, CardsLevel14, CardsLevel13, CardsLevel12, CardsLevel11, CardsLevel10** (Numerico): Conteggi del numero di carte che il

giocatore ha sbloccato e portato a uno specifico livello (dal 10 al 15). Questi dati possono essere particolarmente utili per inferenze sulla progressione e l'investimento del giocatore.

- **CardsEvo** (Numerico): Il conteggio delle carte "evolute" possedute dal giocatore. Le carte evolute sono versioni potenziate di carte esistenti e sono significativamente più forti (anche se sono state aggiunte da poco e per questo relativamente poco diffuse).
- **meanLevelSupportCards** (Numerico): Il livello medio delle carte di supporto sbloccate dal giocatore.
- **SupportCardsLevel15, SupportCardsLevel14, SupportCardsLevel13** (Numerico): Conteggi delle carte di supporto sbloccate e portate a uno specifico livello (13, 14 o 15). Dato che sono conteggi con un numero esiguo di modalità, queste variabili saranno trasformate in fattoriali.
- **NumberSupportCards** (Numerico): Il numero totale di carte di supporto sbloccate dal giocatore, anche questa variabile ha solo 4 modalità e quindi converrà interpretarla come fattoriale.
- **currentLeagueNumber** (Numerico): Il numero di leghe oltrepassate dal giocatore nel mese corrente nella modalità competitiva "Percorso delle Leggende". Questo numero va da 1 a 10 (quindi sarà opportunamente trasformato in variabile fattoriale); superando la decima lega, i giocatori accumulano trofei specifici di questa modalità.
- **currentLeagueTrophies** (Numerico): Il numero di trofei specifici della modalità "Percorso delle Leggende" ottenuti dal giocatore nel mese corrente (rilevante solo dopo aver raggiunto la decima lega).
- **currentLeagueRank** (Numerico): Il rank mondiale ottenuto dal giocatore nel mese corrente nella modalità "Percorso delle Leggende" (se applicabile, per i top player).
- **lastLeagueNumber, lastLeagueTrophies, lastLeagueRank** (Numerico): Le stesse informazioni di **currentLeagueNumber, currentLeagueTrophies, currentLeagueRank** ma relative al mese appena terminato.
- **bestLeagueNumber, bestLeagueTrophies, bestLeagueRank** (Numerico): Le stesse informazioni, ma relative al mese in cui il giocatore ha ottenuto la sua migliore performance storica nella modalità "Percorso delle Leggende".

### 3.4. Plot delle Variabili Chiave

Le seguenti visualizzazioni forniscono un'anteprima delle distribuzioni di alcune delle variabili più significative del dataset, permettendo di identificare pattern e caratteristiche generali del campione di giocatori.

#### 3.4.1. Distribuzione dei Trofei

La Figura 1 mostra l'istogramma dei trofei dei giocatori, mentre la Figura 2 ne illustra la distribuzione della densità.

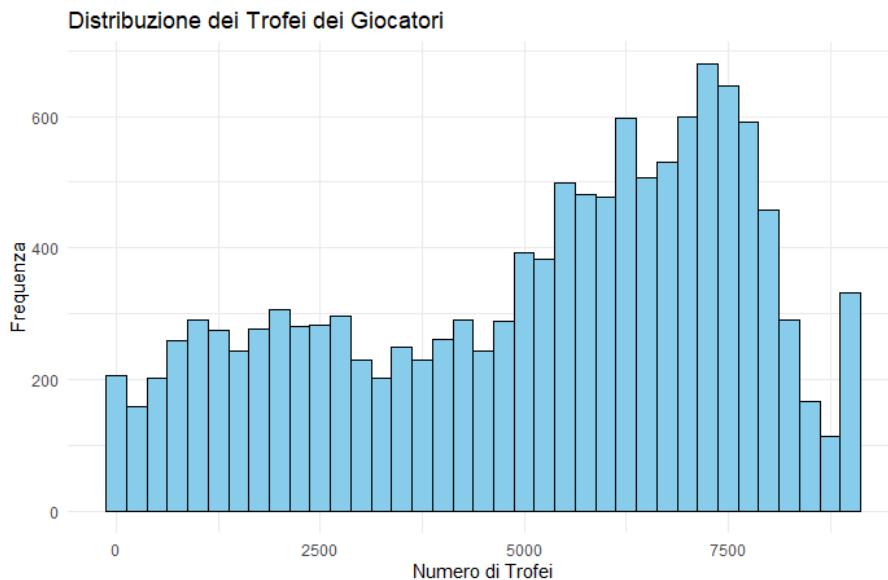


Figura 1: Distribuzione dei Trofei dei Giocatori (Istogramma)

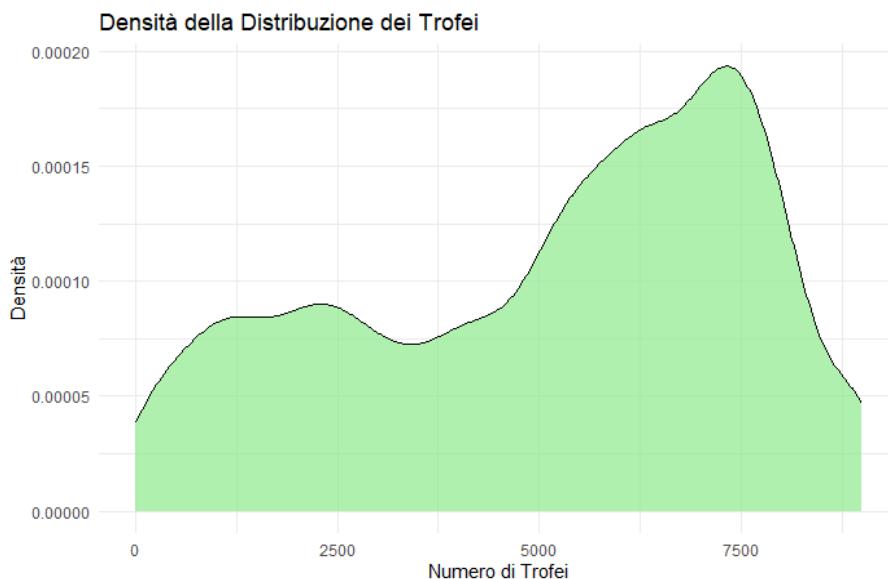


Figura 2: Densità della Distribuzione dei Trofei dei Giocatori

Osservazioni: Si nota una distribuzione a campana con picco intorno ai 6500-7000 trofei, indicando una maggioranza di giocatori di livello intermedio/alto. Il motivo è probabilmente

che il gioco permette facilmente di scalare in questa modalità dato che dopo una soglia ti impedisce di perdere trofei. Notiamo un picco vicino a 2000 che è dove di solito i giocatori meno appassionati si annoiano delle meccaniche del gioco ed abbandonano.

### 3.4.2. Distribuzione del Livello di Esperienza

La Figura 3 mostra la distribuzione del livello di esperienza dei giocatori.

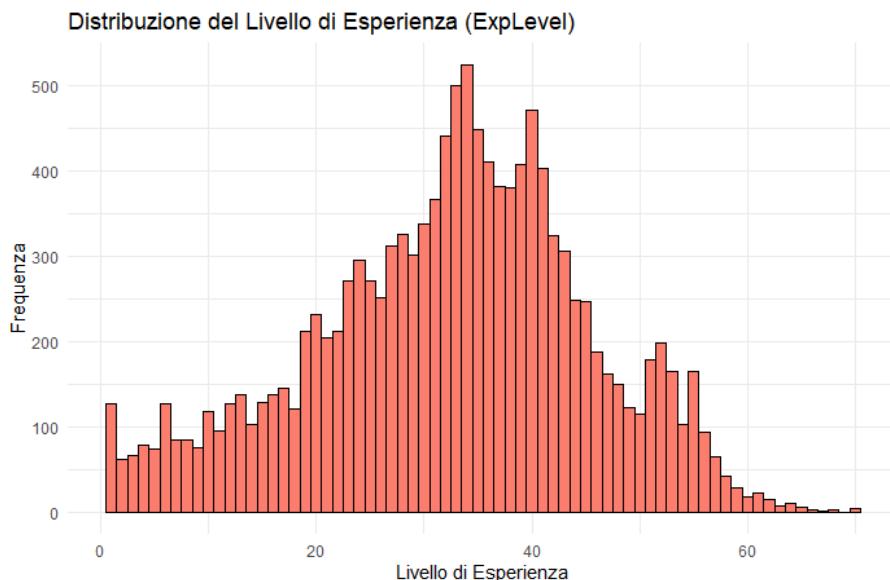


Figura 3: Distribuzione del Livello di Esperienza (ExpLevel)

Osservazioni: Si nota una distribuzione a campana molto piccata nella zona di esperienza che va dal livello 30 al livello 40. La coda sinistra è più spessa sottolineando una quantità relativamente ampia di giocatori alle prime armi.

### 3.4.3. Distribuzione dei Ruoli nel Clan

La Figura 4 illustra la ripartizione dei ruoli dei giocatori all'interno dei clan.

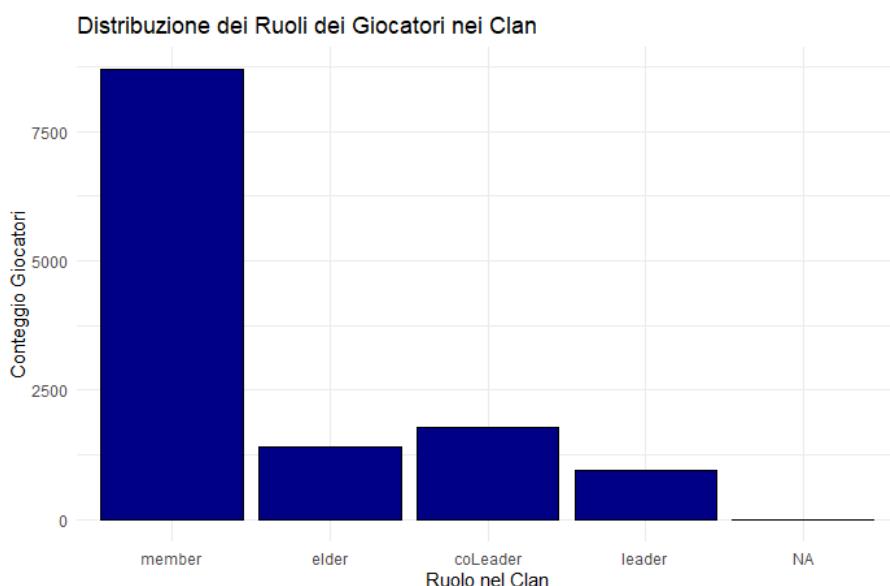


Figura 4: Distribuzione dei Ruoli dei Giocatori nei Clan

**Osservazioni:** La stragrande maggioranza dei giocatori è "Member", con quote minori per i ruoli di "Elder", "Co-Leader" e "Leader", il che è atteso data la gerarchia dei clan. Inoltre un giocatore è senza clan, quindi il suo ruolo è mancante.

#### 3.4.4. Distribuzione del Costo Medio del Deck

La Figura 5 mostra la distribuzione del costo medio in elisir dei deck dei giocatori.

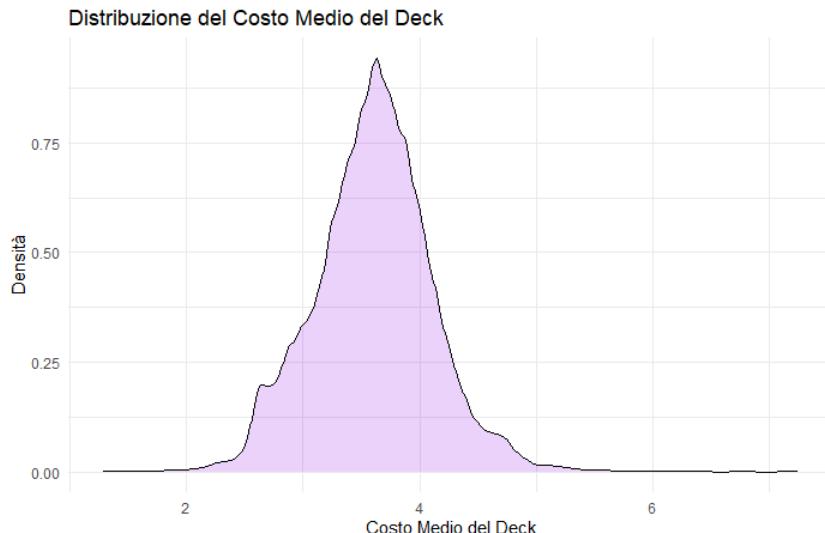


Figura 5: Densità della distribuzione del Costo Medio del Deck

**Osservazioni:** Si osserva un picco molto elevato attorno a 3.0-3.5 elisir, indicando che la maggior parte dei giocatori predilige mazzi con un costo medio per un gioco flessibile, evitando mazzi troppo "pesanti" o troppo "leggeri". Un deck pesante rende il giocatore poco reattivo ed incapace di rispondere all'avversario. Un deck troppo leggero rende il giocatore reattivo ma incapace di usare carte sufficientemente forti da fronteggiare l'avversario.

#### 3.4.5. Top 10 Carte Preferite

La Figura 6 presenta le 10 carte più frequentemente indicate come "preferite" dai giocatori nel dataset.

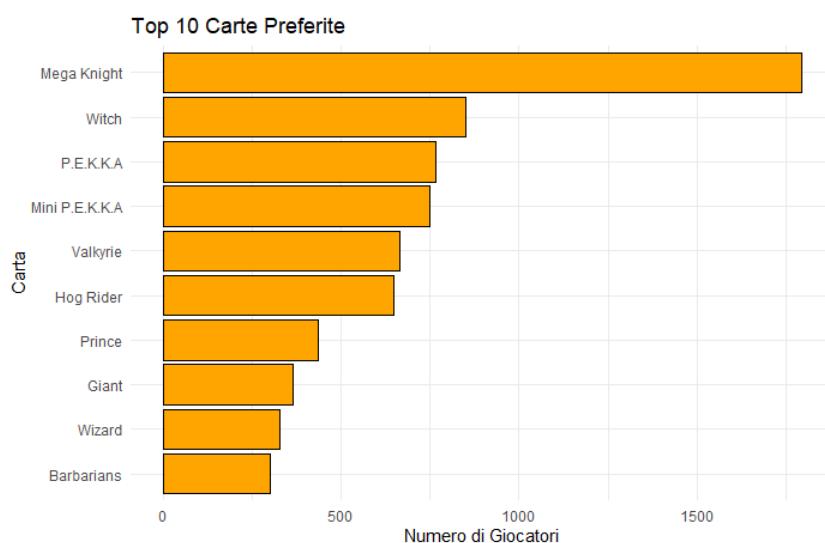


Figura 6: Top 10 Carte Preferite dai Giocatori

**Osservazioni:** La carta più popolare è il Gran Cavaliere (Mega Knight). Carta “no-skill” nel senso che per utilizzarla non serve ragionarci molto, infatti è da tempo nel metà di gioco attuale.

### 3.4.6. Leghe e Trofei nel Percorso delle Leggende

Analizziamo ora le performance dei giocatori nella modalità più competitiva di Clash Royale: il Percorso delle Leggende. Abbiamo scelto di visualizzare la distribuzione del numero di leghe superate nella scorsa stagione (`lastLeagueNumber`) per i giocatori che hanno superato il livello iniziale (escludendo quindi Lega 1 che rappresenta il livello di ingresso minimo). Successivamente, esamineremo la distribuzione dei trofei massimi raggiunti in questa modalità (`bestLeagueTrophies`), focalizzandoci sui giocatori che hanno effettivamente raggiunto la lega massima nella loro stagione migliore.

#### 3.4.6.1. Distribuzione dei Livelli di Lega nella Stagione Scorsa

La Figura 7 mostra la distribuzione dei giocatori per il livello di lega raggiunto nella scorsa stagione nel Percorso delle Leggende, escludendo il livello 1 (che contava oltre 10.000 giocatori).

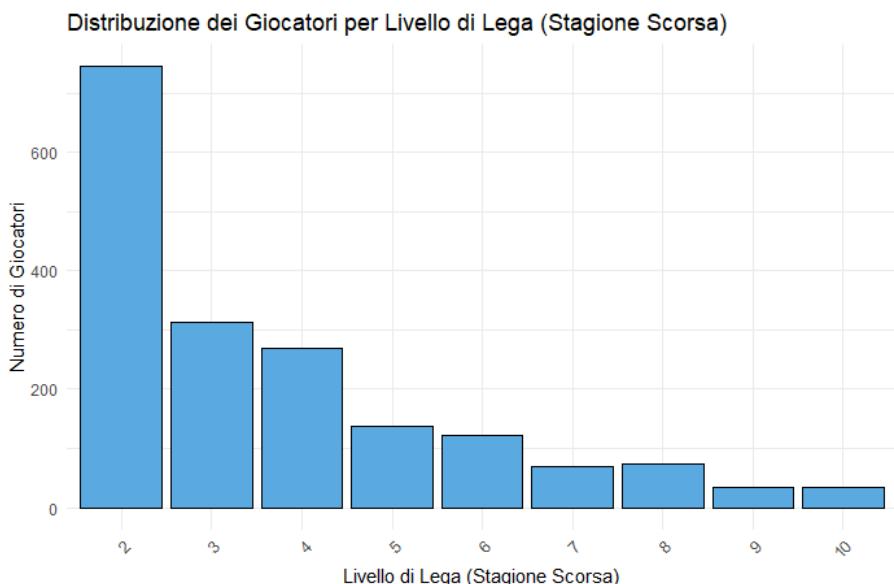
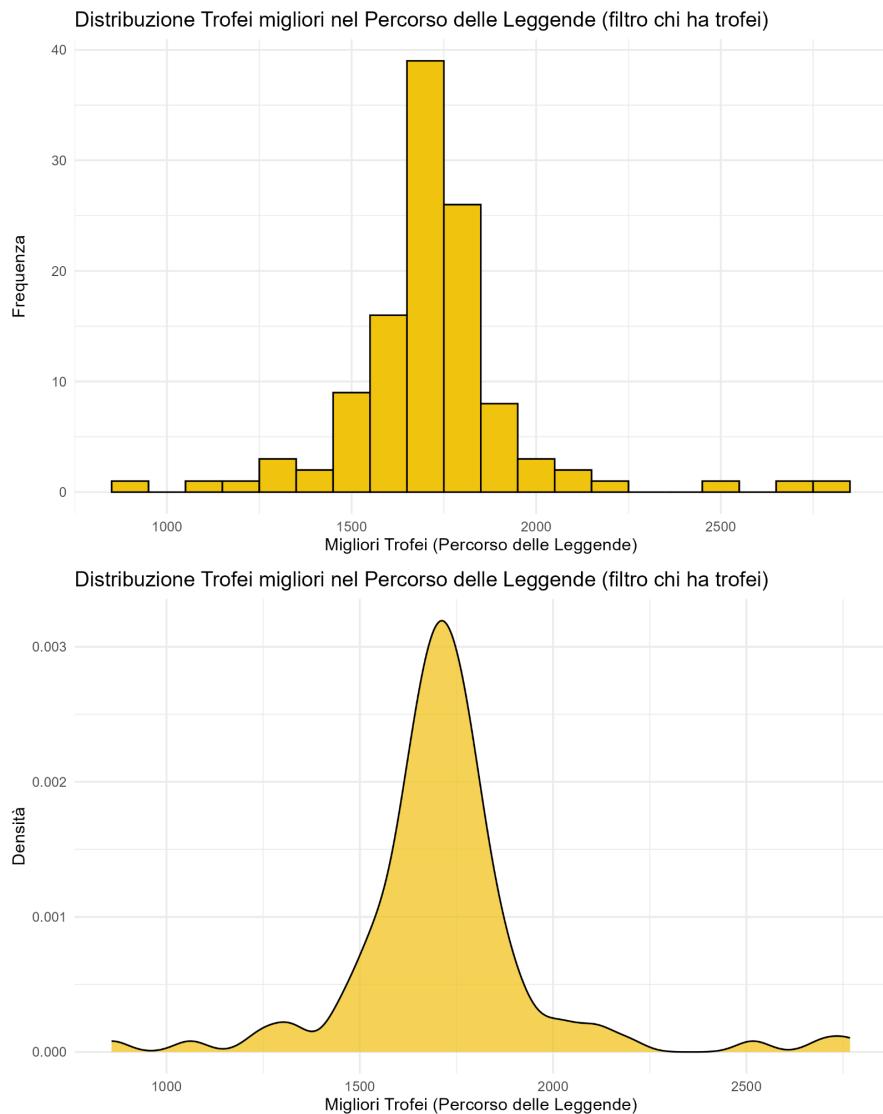


Figura 7: Distribuzione dei Giocatori per Livello di Lega (Stagione Scorsa) - Escludendo Lega 1

**Osservazioni:** Si può notare una concentrazione di giocatori nelle leghe basse, suggerendo che la maggior parte dei giocatori non riesce a stare al passo con i ritmi competitivi di questa modalità che ogni mese si resetta e riporta tutti i giocatori a lega 1.

#### 3.4.6.2. Distribuzione dei Migliori Trofei nel Percorso delle Leggende

La Figura 8 illustra la distribuzione dei trofei massimi raggiunti dai giocatori nel Percorso delle Leggende, considerando solo coloro che hanno effettivamente accumulato trofei (`bestLeagueTrophies > 0`).



*Figura 8: Distribuzione dei Migliori Trofei nel Percorso delle Leggende*

**Osservazioni:** La distribuzione dei trofei è estremamente piccata e con code molto lunghe e sottili. Il motivo è che raggiunta la lega 10 il gioco consegna automaticamente a chi ci arriva circa 1700 trofei. Una lunga coda verso l'alto indica pochi giocatori con un numero estremamente elevato di trofei, questo riflette la natura piramidale del sistema di classificazione della modalità competitiva.

# Capitolo 4: Preparazione e Preprocessing dei Dati per fare Regressione

Questo capitolo descrive le fasi di ulteriore preparazione e preprocessing a cui è stato sottoposto il dataset piatto. Infatti il dataset era già in un formato pulito e idoneo per l'analisi inferenziale, però ci sono alcuni valori assenti (NA) da gestire.

## 4.1 Gestione dei Valori Mancanti (NA)

L'analisi preliminare del dataset ha rivelato la presenza di valori mancanti (NA) in diverse variabili. La gestione di questi NA è stata effettuata seguendo strategie mirate a preservare la qualità del dataset senza compromettere troppo l'ammontare del dataset stesso. La Tabella riassume la distribuzione iniziale dei valori mancanti per le colonne che effettivamente presentano mancanze.

Variabile	Conteggio NA Iniziale
role	1
currentFavouriteCard	90
starPoints	26
legacyTrophyRoadHighScore	4791
meanCostDeck	4
meanLevelSupportCards	193
daysSinceRegistration	2643

Le decisioni prese per la gestione dei NA sono state le seguenti:

- Rimozione delle righe con NA:** Per le variabili `role`, `currentFavouriteCard`, `starPoints`, `meanCostDeck` e `meanLevelSupportCards`, i valori mancanti erano riconducibili a situazioni specifiche. Data la natura e il numero limitato di questi NA (un totale di 286 righe affette, pari a circa il 2.2% del dataset originale), si è optato per la rimozione delle righe corrispondenti. Questa scelta minimizza la perdita di informazioni complessiva garantendo al contempo l'integrità delle variabili rimanenti per l'analisi.
- Rimozione della colonna `legacyTrophyRoadHighScore`:** Questa variabile presenta un'alta percentuale di valori mancanti (4791 NA, circa il 37% delle osservazioni) ed è legata a una modalità di gioco non più attiva. Per la sua scarsa rilevanza e l'elevata incompletezza, l'intera colonna è stata rimossa dal dataset.
- Gestione di `daysSinceRegistration`:** Questa variabile presentava 2643 valori mancanti (circa il 20.6% delle osservazioni). I valori mancanti sono dovuti ad una specifica limitazione dell'API che non è in grado di fornirci il valore di questa variabile per gli utenti iscritti al gioco da meno di un anno. Non volendo rimuovere un numero

così elevato di osservazioni né l'intera colonna (potenzialmente rilevante), si è adottato, già nella fase precedente di pulizia, una strategia combinata:

1. È stata creata una variabile fattoriale binaria `is_new_player` (TRUE se `daysSinceRegistration` era inizialmente NA, FALSE altrimenti) per distinguere i giocatori con meno di un anno di attività.
2. I valori NA di `daysSinceRegistration` sono stati sostituiti con `180`, un valore rappresentativo (circa la metà di 1 anno).
3. È stata creata una nuova variabile `yearsSinceRegistration` per offrire un'ulteriore rappresentazione dell'anzianità per la modellazione. Infatti l'API fornisce questo valore per ogni giocatore.

Un'altra problematica riscontrata è legata al fatto che per definizione le variabili numeriche del nostro dataset debbano essere non negative. Ma facendo un controllo si nota che c'è un giocatore a cui per un bug (dell'API o di Clash Royale stesso) il numero di starPoints risulta negativo. Eseguendo un filtraggio con condizione `starPoints >= 0` otteniamo una tabella priva di variabili numeriche negative (conformemente col loro significato).

A seguito di queste operazioni, il dataset risultante è privo di valori mancanti e di anomalie legate a valori negativi, garantendo una base dati pulita e completa.

## 4.2 Suddivisione del Dataset (Training e Test Set)

Per garantire una valutazione imparziale e robusta del modello di regressione il dataset pulito è stato suddiviso in due porzioni distinte: un set di training e un set di test. Questa pratica è essenziale per stimare la capacità di generalizzazione del modello su dati non visti né durante la fase di apprendimento né durante la fase di scelta del modello.

La suddivisione è stata eseguita in modo casuale, utilizzando un `set.seed()` per assicurare la riproducibilità dei risultati. La ripartizione adottata è stata la seguente:

- **Training Set:** 80% delle osservazioni del dataset pulito. Questo set verrà utilizzato per l'addestramento e la scelta del modello.
- **Test Set:** 20% delle osservazioni rimanenti. Questo set sarà “blindato” e verrà utilizzato solo alla fine del processo di modellazione per una valutazione finale e non distorta delle prestazioni del modello.

Dopo la pulizia, il dataset conta circa 12.525 osservazioni. La divisione ha prodotto un set di training di circa 10.020 osservazioni e un set di test di circa 2.505 osservazioni. Questa dimensione del test set è sufficientemente robusta per una stima affidabile dell' errore di predizione del modello.

## 4.3 Verifica delle Assunzioni Preliminari per la Regressione Lineare

Prima di procedere con la costruzione del modello di regressione lineare multivariata, è fondamentale verificare alcune delle sue assunzioni chiave sulle relazioni tra le variabili e sulla struttura dei dati. Questo passaggio assicura che i risultati del modello siano validi e affidabili. Per questa analisi, la variabile dipendente scelta è il **numero di trofei attuali**

(**trophies**), in quanto ritenuta la metrica più diretta delle prestazioni competitive influenzata dai dati in nostro possesso.

Le verifiche preliminari si concentreranno su:

- **Linearità dei dati:** si assume che la relazione tra predittori (X) e risultato (Y) sia lineare.
- **Normalità dei residui:** si assume che gli errori residui siano distribuiti normalmente.
- **Omogeneità della varianza dei residui:** si assume che i residui abbiano una varianza costante (omoschedasticità).
- **Indipendenza degli errori residui:** si assume che gli errori residui siano incorrelati.

L'assenza di queste assunzioni significherebbe:

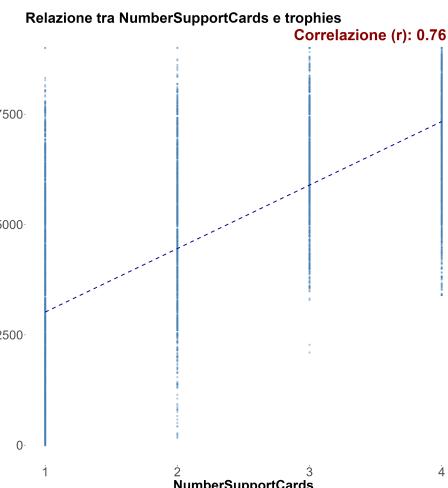
- Non-linearità nella relazione tra risultato e predittori.
- Eterschedasticità: varianza non costante degli errori.
- Presenza di valori limite nei dati che possono essere:
  - Outliers: osservazioni in cui la variabile risposta (Y) è insolita o estrema, dato il suo valore della variabile preditrice (X). In altre parole, l'osservazione si discosta molto dall'iperpiano di regressione fittato (il valore del residuo è molto grande).
  - High-leverage points: osservazioni che hanno un valore insolito o estremo per una o più variabili predittrici (X), rispetto agli altri dati nel dataset.

#### 4.3.1 Verifica della Dipendenza Lineare e Ottimizzazione delle Variabili

La fase successiva della preparazione dei dati si è concentrata sullo studio della relazione tra le variabili predittive (X) e la variabile target **trophies** (Y). L'obiettivo primario era migliorare la linearità di queste relazioni, un prerequisito fondamentale per un'efficace modellazione con la regressione lineare.

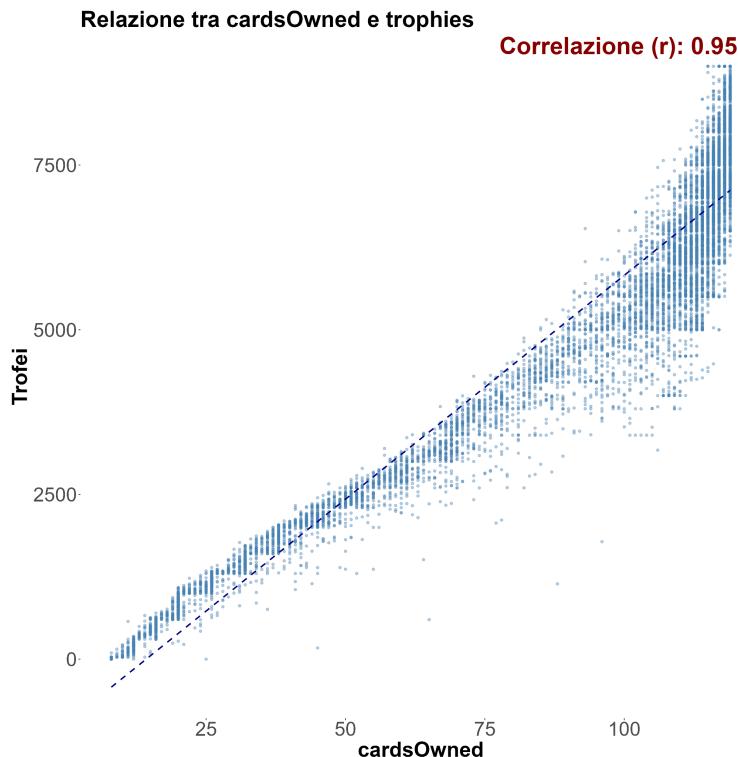
È stata iniziata un'analisi visiva approfondita degli scatter plot di ogni variabile indipendente rispetto a **trophies**. Questo esame ha permesso di identificare quattro categorie principali di variabili ed il relativo trattamento:

- **Variabili Numeriche Convertite in Fattori Ordinati:** 7 variabili numeriche, sebbene presentano valori discreti, mostrano un numero limitato di modalità distinte e un comportamento più affine a categorie ordinali che a scale continue. Trattarle come fattori ordinati permette al modello di catturare meglio le differenze qualitative tra i loro livelli. Queste variabili sono state convertite in tipo fattore (**factor**) e sono state mantenute con il loro nome originale. I livelli sono stati specificati in ordine crescente per mantenere l'ordinalità implicita: **bestLeagueNumber**



(livelli da 1 a 10), `currentLeagueNumber` (livelli da 1 a 10), `lastLeagueNumber` (livelli da 1 a 10), `NumberSupportCards` (livelli da 1 a 4), `SupportCardsLevel13` (livelli da 0 a 4), `SupportCardsLevel14` (livelli da 0 a 4) e `SupportCardsLevel15` (livelli da 0 a 4).

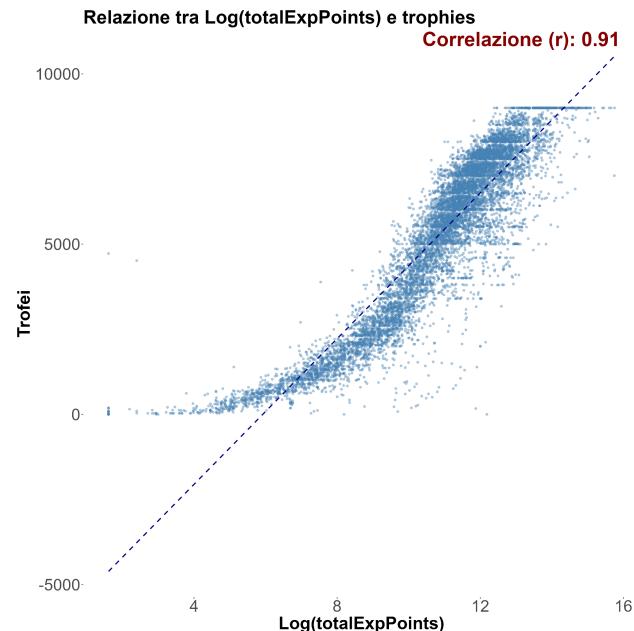
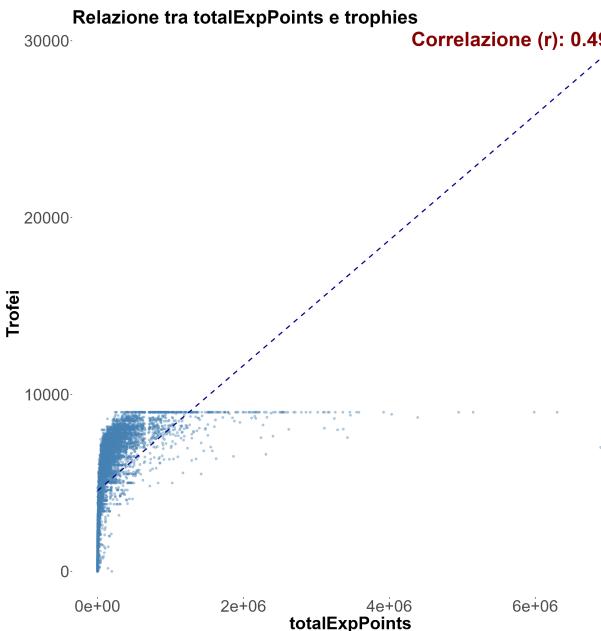
- **Relazioni Già Lineari o Sufficientemente Lineari:** rispetto a 9 variabili si è notato subito che `trophies` ha una chiara dipendenza lineare. Queste sono state mantenute nel loro formato originale. Tra queste figurano `CardsLevel10`, `CardsLevel11`, `CardsLevel12`, `CardsLevel13`, `cardsOwned`, `challengeMaxWins`, `daysSinceRegistration`, `expLevel`, `meanCostDeck`.



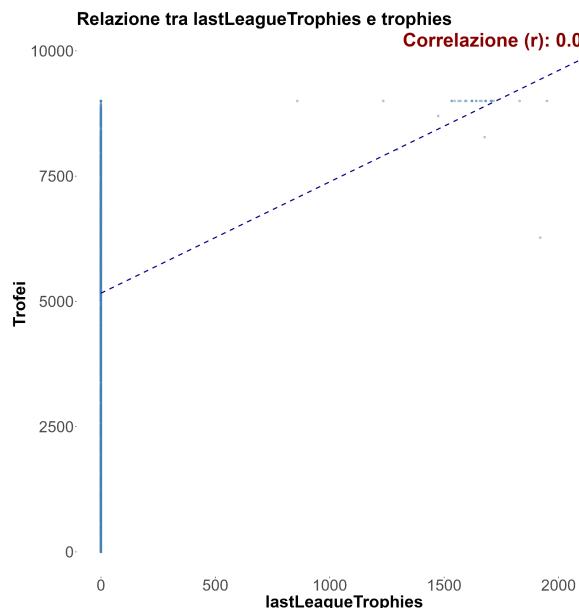
- **Relazioni Non Lineari che Necessitano di Trasformazione:** 18 variabili presentavano curve non lineari (esponenziali, logaritmiche e potenze) che potevano essere linearizzate tramite trasformazioni matematiche. Per ciascuna di queste, sono state testate diverse trasformazioni (logaritmo, radice quadrata, radice quarta e quadrato) e la scelta finale è ricaduta su quella che visivamente migliorava maggiormente la linearità e, di conseguenza, il coefficiente di correlazione di Pearson. A titolo esemplificativo, è stato osservato un significativo miglioramento della correlazione per `battleCount`, la cui correlazione con `trophies` è passata da 0.52 (originale) a 0.88 dopo l'applicazione della trasformazione `(log1p(battleCount))^2`. Le altre trasformazioni applicate per migliorare la linearità sono:

- **Logaritmo (`log1p`):** `CardsLevel14`, `CardsLevel15`, `challengeCardsWon`, `clanCardsCollected`, `totalDonations`, `donations`, `threeCrownWins`, `starPoints`, `totalExpPoints`, `tournamentBattleCount`, `wins`.
- **Radice Quarta ( $^{(1/4)}$ ):** `expPoints`, `losses`.

- **Radice Quadrata (sqrt):** CardsEvo, yearsSinceRegistration.
- **Quadrato (^2):** meanLevelCards, meanLevelSupportCards.



- **Relazioni Insoddisfacenti che Portano alla Rimozione:** 9 variabili, anche dopo aver provato diverse trasformazioni per linearizzare la loro relazione con **trophies**, non hanno mostrato una dipendenza soddisfacente. Per non introdurre rumore o variabili non informative nel modello, è stata optata la loro rimozione definitiva dal dataset. Queste variabili includono **donationsReceived**, **warDayWins**, **currentLeagueRank**, **lastLeagueRank**, **bestLeagueRank**, **bestLeagueTrophies**, **lastLeagueTrophies**, **currentLeagueTrophies** e **tournamentCardsWon** (a lato un esempio di variabile piatta che assume lo stesso valore per quasi tutto il dataset e che per questo è stata rimossa).



A seguito di questa fase di ottimizzazione, il dataset di training (**train\_data\_processed**) è stato aggiornato. Dalle 173 variabili originali, 9 sono state rimosse completamente, 7 sono state trasformate in fattoriali mentre 18 sono state sostituite dalle loro versioni trasformate. Il dataset finale per la modellazione conta ora 164 variabili:

- **trophies:** variabile target;

- **tag e name**: identificatori unici e poiché includere variabili con un valore quasi unico per ogni osservazione non ha alcun senso predittivo non saranno variabili indipendenti dei nostri modelli;
- **bestTrophies**: è una variabile critica da rimuovere dai nostri modelli in quanto estremamente simile alla variabile target (includerla nel modello renderebbe il modello banalmente predittivo solo grazie a questa variabile, senza imparare nulla dalle altre caratteristiche);
- 160 variabili predittive di cui 27 numeriche (43 iniziali - 7 trasformate in fattoriali - 9 eliminate) e 133 fattoriali.

#### 4.3.2 Gestione della Multicollinearità Attraverso l'Analisi VIF

La presenza di multicollinearità, ovvero una forte correlazione tra le variabili predittive in un modello di regressione, è un aspetto critico da gestire per garantire la stabilità dei coefficienti stimati e la validità delle inferenze. Per affrontare questo problema, è stata condotta un'analisi approfondita del Variance Inflation Factor (**VIF**). Essendo il modello caratterizzato da numerose variabili categoriche (fattoriali), è stato utilizzato il **GVIF<sup>^(1/(2\*Df))</sup>** (per i gradi di libertà), e l'obiettivo è stato quello di rimuovere le variabili con valori superiori a 5, un indicatore riconosciuto di multicollinearità problematica. Il processo di riduzione della multicollinearità è stato iterativo, procedendo per passaggi successivi per eliminare le variabili più problematiche:

Variabile	VIF aggiustato	Df	GVIF
meanCostDeck	787.	1	28.1
sq_log_battleCount	413.	1	20.3
expLevel	216.	1	14.7
log_totalExpPoints	157.	1	12.5
log_wins	156.	1	12.5
fourth_root_losses	137.	1	11.7
sqrt_yearsSinceRegistration	107.	1	10.4
Mega Knight	103.	1	10.1
P.E.K.K.A	61.7	1	7.85
daysSinceRegistration	60.1	1	7.75

1. **Prima Fase di Pulizia:** L'analisi iniziale ha rivelato valori di **VIF\_Adjusted** estremamente elevati per diverse variabili (anche superiori a 700). In questa fase sono state rimosse le seguenti variabili, considerate altamente ridondanti o con un impatto sproporzionato sulla multicollinearità generale:
  - **meanCostDeck**: Il costo medio del deck era fortemente correlato con le variabili booleane inserite (infatti conoscendo il costo di ogni carta e le carte

contenute nel deck del giocatore si risale in modo diretto al costo medio del deck).

- **sq\_log\_battleCount**: Essendo un indicatore del numero totale di battaglie presenta un'eccessiva correlazione con gli indicatori del numero di vittorie e del numero di sconfitte (possiamo approssimare il numero di partite giocate proprio come la somma di queste due dato che in Clash Royale i pareggi sono più unici che rari).
- **expLevel**: Il livello di esperienza è quasi sinonimo di **log\_totalExpPoints**; si è optato per mantenere quest'ultima come variabile in quanto continua e quindi in grado di cogliere maggiori sfumature.
- **sqrt\_yearsSinceRegistration**: Tra le due variabili che misuravano l'anzianità del giocatore (**daysSinceRegistration** e **sqrt\_yearsSinceRegistration**), è stata conservata la prima che è continua.

Variabile	VIF aggiustato	Df	GVIF
log_wins	8.15	1	66.5
log_totalExpPoints	6.90	1	47.7
fourth_root_losses	5.08	1	25.8
cardsOwned	4.62	1	21.4
log_threeCrownWins	4.59	1	21.1
The Log	3.62	1	13.1
sq_meanLevelCards	3.62	1	13.1

2. **Seconda Fase di Pulizia:** A seguito della prima rimozione, i VIF sono diminuiti drasticamente, ma alcune variabili mostravano ancora valori superiori a 5. Sono state quindi rimosse ulteriori variabili per affinare il modello:

- **fourth\_root\_losses**: Data la forte correlazione con **log\_wins** (poichè il gioco ha uno script che tende a rendere il numero di sconfitte proporzionale a quello di vittorie), si è scelto di mantenere solo **log\_wins** considerata più informativa.
- **log\_threeCrownWins**: il numero di vittorie con tre corone è un dato interessante ma sicuramente lo si può rimuovere in favore di un dato molto più significativo quale il numero di vittorie.
- **sq\_meanLevelCards** e **sq\_meanLevelSupportCards**: Tra le variabili rimaste vi sono i conteggi del numero di carte possedute dal giocatore per i livelli più importanti, facendo la media di questi conteggi pesati si può ottenere una buona approssimazione di queste due variabili che quindi andiamo a rimuovere.

A seguito di queste multiple fasi di pulizia, il modello presenta ora VIF a livelli significativamente inferiori alla soglia critica di 5 per tutte le variabili predittive. Le variabili più

informative come `log_totalExpPoints` (5.50), `cardsOwned` (4.20) e `log_wins` (4.10) sono state mantenute data la loro importanza teorica nel descrivere il progresso e la performance del giocatore. Anche le restanti variabili, legate alla presenza o assenza di una carta nel deck del giocatore, hanno ora `VIF_Adjusted` che si attestano ben al di sotto di 5.

Variabile	VIF aggiustato	Df	GVIF
<code>log_totalExpPoints</code>	5.50	1	30.2
<code>cardsOwned</code>	4.20	1	17.6
<code>log_wins</code>	4.10	1	16.8
The Log	3.62	1	13.1
Valkyrie	3.46	1	12.0
Fireball	3.46	1	11.9

Questa fase di pulizia ha permesso di ottenere un modello più robusto e interpretabile, riducendo la ridondanza tra i predittori senza sacrificare informazioni cruciali per la previsione dei trofei. Il modello risultante è ora pronto per una valutazione più approfondita delle sue assunzioni tramite l'analisi dei grafici diagnostici.

#### 4.3.3 Verifica delle Ipotesi del Modello Lineare Iniziale

La validità delle inferenze basate su un modello di regressione lineare multipla dipende dal soddisfacimento di diverse assunzioni fondamentali: linearità (analizzata nel paragrafo precedente), omoschedasticità, normalità e indipendenza dei residui. Per il modello lineare inizialmente costruito, che utilizza `trophies` come variabile risposta, abbiamo condotto un'analisi diagnostica approfondita sia tramite ispezione visiva dei grafici dei residui sia attraverso test statistici formali.

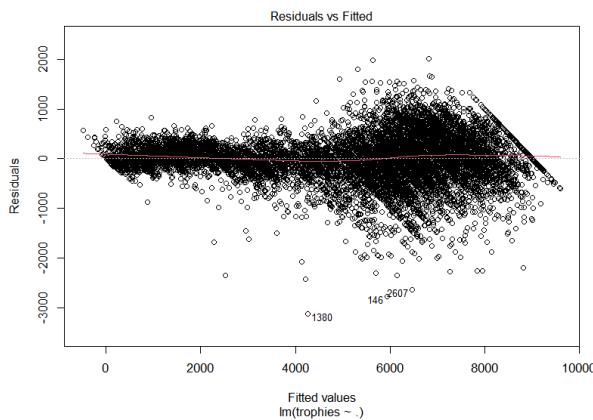


Figura 9: Residui vs. Valori Fittati

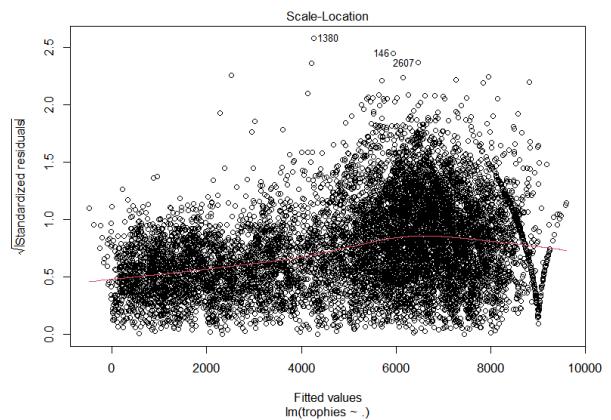


Figura 10: Scala-Posizione

a) **Linearità e Omoschedasticità (Analisi Visiva)** L'esame del grafico dei "Residui vs. Valori Fittati" (Figura 9) ha rivelato un leggero pattern a "imbuto" (eteroschedasticità) e una deviazione dalla distribuzione casuale dei punti attorno alla linea zero. Questo suggerisce

una leggera violazione sia dell'assunzione di linearità (tra predittori e risposta) sia dell'omoschedasticità, ovvero la varianza dei residui non è costante al variare dei valori predetti. Il grafico di "Scala-Posizione" (Figura 10) ha confermato la presenza di eteroschedasticità, mostrando una linea rossa non orizzontale, indicativa di una variabilità dei residui che cambia con i valori predetti.

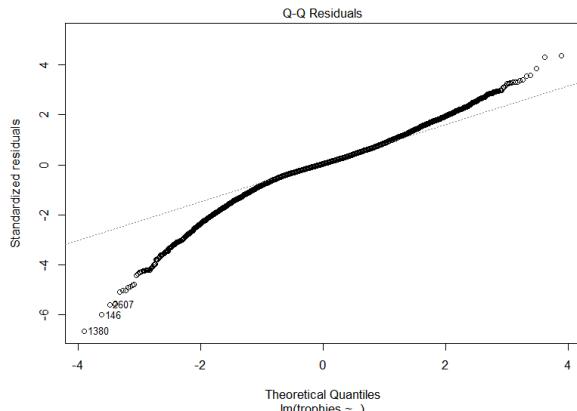


Figura 11: Q-Q plot dei residui

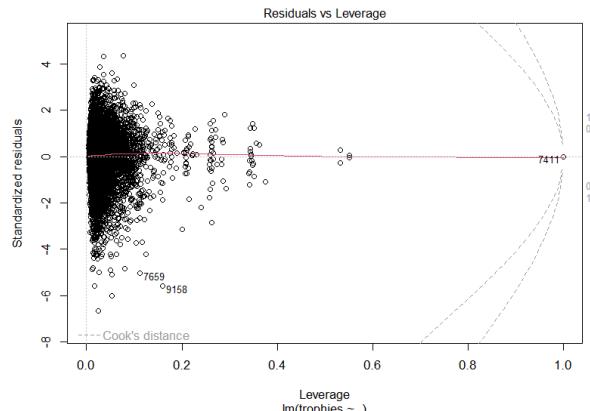


Figura 12: Residui vs. Leverage

**b) Normalità dei Residui (Analisi Visiva)** Il Q-Q plot dei residui (Figura 11) ha mostrato una leggera deviazione dalla linea di riferimento diagonale, assumendo una forma a "S". Questo indica che i residui non seguono una distribuzione normale, in particolare con code più pesanti del previsto e una leggera asimmetria. La leggera assenza di normalità è stata supportata dall'istogramma dei residui (Figura 13), che ha presentato una distribuzione leggermente asimmetrica a sinistra (asimmetria negativa), con una lunga coda che si estende verso valori residui negativi elevati, leggermente lontana dalla forma a campana di una distribuzione normale.

Istogramma dei Residui

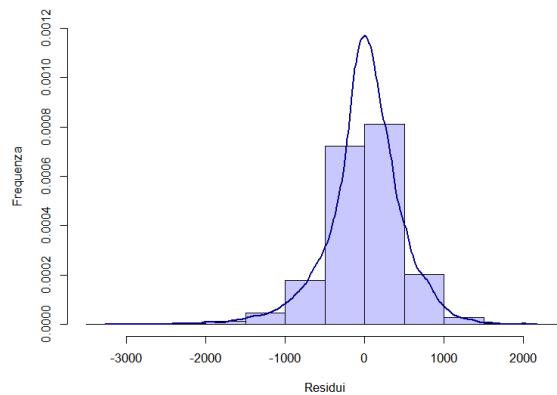


Figura 13: Istogramma dei residui

**c) Punti influenti (Outlier e High-leverage points)** L'analisi del grafico "Residui vs. Leverage" (Figura 12) è cruciale per identificare osservazioni che potrebbero influenzare indebitamente le stime del modello. In questo grafico:

- **Outlier:** Sono identificati da residui standardizzati elevati (punti al di fuori della fascia [-2,2]). Numerosi punti si trovano al di fuori di queste fasce, indicando la presenza di molti outlier (tra cui le osservazioni **146, 1380, 2607, 7659, 9158**) nel modello originale.
- **High-leverage points (Cook's Distance):** Sono osservazioni che hanno valori insoliti per una o più variabili predittive (elevato leverage), indicati dai punti situati più a destra sul grafico come l'osservazione **7411**. Le linee tratteggiate concentriche rappresentano le "distanze di Cook". I punti che superano queste soglie (tipicamente 0.5 o 1) sono considerati altamente influenti sul modello.

Le osservazioni con indice **3815, 5412, 8604 e 9529** sono state segnalate come potenziali problemi nella realizzazione dei grafici Q-Q plot, Scala-Posizione e *Residui vs. Leverage*. La rimozione di questi ed altri punti che mostrano un'alta influenza potrebbe essere considerata nelle fasi successive per migliorare la robustezza del modello.

**d) Test di Specificazione Formali** Per quantificare e confermare le osservazioni visive, sono stati eseguiti diversi test statistici formali:

- **Test t sulla media dei residui:** Il test ha prodotto un p-value approssimato a 1 ed una media campionaria di  $8.178651 \times 10^{-15}$ , questo risultato conferma che la media dei residui non è statisticamente diversa da zero. Questa è un'assunzione basilare che il modello soddisfa correttamente, dato l'inserimento dell'intercetta.
- **Test di Anderson-Darling per la normalità:** Dato l'ampio campione di 10020 osservazioni, il test di Shapiro-Wilk non è computazionalmente adatto. Si è quindi optato per il test di Anderson-Darling, che ha fornito un p-value estremamente basso ( $A=75.622$ , p-value  $<2.2 \times 10^{-16}$ ). Questo risultato conferma statisticamente la non normalità dei residui, validando quanto osservato visivamente nel Q-Q plot e nell'istogramma (leggera asimmetria negativa). In letteratura, però, viene osservato che per quando il numero di osservazioni è superiore a 5000 (o anche meno, a seconda delle fonti), quasi ogni test di normalità (Shapiro-Wilk, Kolmogorov-Smirnov, Anderson-Darling, ecc.) rifiuterà l'ipotesi nulla di normalità, anche se la deviazione dalla normalità è minima e non ha un impatto pratico sulle conclusioni del modello. Questo perché con un campione molto grande si ha molta "potenza" per rilevare anche le più piccole deviazioni. Tuttavia, per campioni grandi, il Teorema del Limite Centrale assicura che le distribuzioni campionarie dei coefficienti di regressione tendano alla normalità anche se i residui non sono perfettamente normali. Questo significa che, anche con piccole deviazioni dalla normalità dei residui, le stime e le conclusioni basate sul modello lineare rimangono abbastanza robuste per campioni di grande dimensione.
- **Test di Breusch-Pagan per l'eteroschedasticità:** Il test ha restituito un p-value estremamente basso ( $BP=1874, df=302, p\text{-value } <2.2 \times 10^{-16}$ ). Questo risultato conferma la presenza di eteroschedasticità significativa, indicando che la varianza degli errori non è costante e dipende dai valori dei predittori. Questa è una violazione critica che mina l'efficienza delle stime dei coefficienti e la validità degli errori standard (un miglioramento si potrebbe avere rimuovendo outliers e high-leverage points).
- **Test di Durbin-Watson per l'autocorrelazione:** Il test ha mostrato un valore Durbin-Watson  $DW=1.2359$  con un p-value  $<2.2 \times 10^{-16}$ . Un valore DW significativamente inferiore a 2 e un p-value così basso suggeriscono una autocorrelazione dei residui. Questo indica che i residui consecutivi sono correlati tra loro, violando l'assunzione di indipendenza degli errori e potenzialmente portando a errori standard sottostimati e a inferenze non valide.

In sintesi, l'analisi diagnostica e i test di specificazione hanno rivelato che il modello lineare costruito utilizzando **trophies** come variabile risposta non trasformata soffre di leggere violazioni delle assunzioni chiave della regressione lineare e la presenza di qualche outlier e punto influente. Queste violazioni possono compromettere l'affidabilità delle stime dei coefficienti e la validità delle inferenze statistiche derivate dal modello.

#### 4.3.4 Verifica delle Ipotesi del Modello Lineare con Variabile Risposta Trasformata

A fronte delle violazioni delle assunzioni nel modello iniziale (Sezione 4.3.2), è stato sviluppato un nuovo modello lineare utilizzando la trasformazione logaritmica della variabile risposta,  $\log(\text{trophies}+1)$ . Questa trasformazione mirava a linearizzare le relazioni, stabilizzare la varianza e avvicinare la distribuzione dei residui alla normalità. Di seguito, vengono presentati i risultati dell'analisi diagnostica e dei test di specificazione per questo modello migliorato.

**a) Linearità e Omoschedasticità (Analisi Visiva)** L'esame del grafico dei "Residui vs. Valori Fittati" (Figura 14) mostra un leggero miglioramento rispetto al modello precedente. I punti sono ora distribuiti in modo più casuale attorno alla linea zero, senza il pattern a "imbuto" osservato in precedenza. Anche l'andamento della linea rossa è leggermente più piatto. Il grafico di "Scala-Posizione" (Figura 15) supporta questo miglioramento, con una linea rossa leggermente più orizzontale, indicando una riduzione dell'eteroschedasticità. La varianza dei residui appare ora più costante al variare dei valori fittati.

**b) Normalità dei Residui (Analisi Visiva)** Il Q-Q plot dei residui (Figura 16) rivela che la maggior parte dei punti si allinea, ora un po' più di prima, alla linea di riferimento diagonale. Sebbene le code della distribuzione possano ancora mostrare leggere deviazioni (in particolare a sinistra), l'allineamento complessivo è migliore, indicando una maggiore aderenza alla normalità. L'istogramma dei residui (Figura 17) conferma che questo miglioramento sia minimo dato che la distribuzione presenta ancora una leggera asimmetria negativa ma con le code leggermente meno estese rispetto al modello non trasformato. Questo indica un'approssimazione alla normalità più accettabile per scopi pratici.

**c) Punti influenti (Outlier e High-leverage points)** L'analisi del grafico "Residui vs. Leverage" (Figura 18) mostra che, sebbene la trasformazione abbia contribuito a

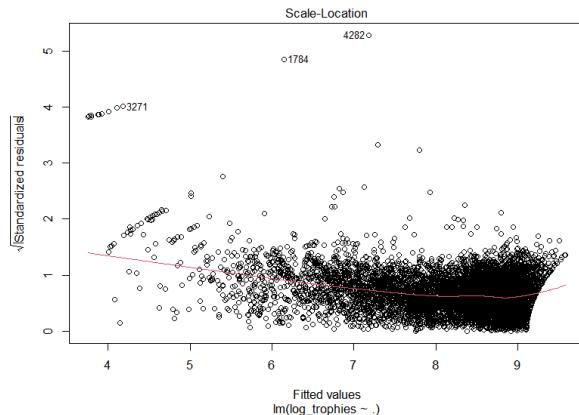


Figura 14: Residui vs. Valori Fittati

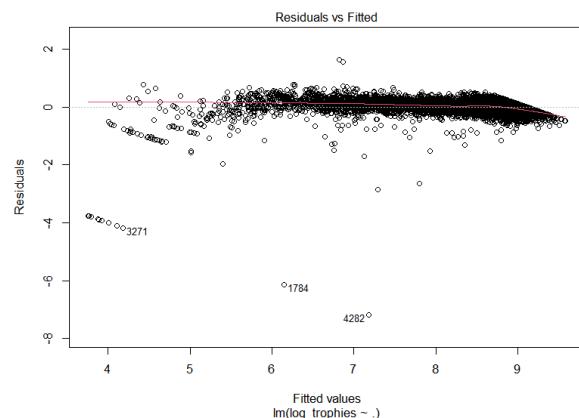


Figura 15: Scala-Posizione

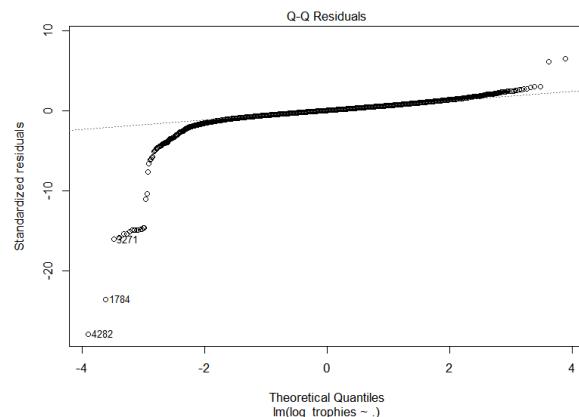


Figura 16: Q-Q plot dei residui

(Figura 17) conferma che questo miglioramento sia minimo dato che la distribuzione presenta ancora una leggera asimmetria negativa ma con le code leggermente meno estese rispetto al modello non trasformato. Questo indica un'approssimazione alla normalità più accettabile per scopi pratici.

raggruppare i punti e a ridurre l'entità dei residui più estremi, alcuni outlier e high-leverage points persistono. Sono ancora visibili punti identificati come problematici (come 1784, 3271, 4282). Tuttavia, la scala dei residui standardizzati è generalmente più contenuta e le distanze di Cook più vicine allo zero per la maggior parte delle osservazioni rispetto al modello iniziale. Ciò suggerisce che, pur rimanendo delle osservazioni estreme, la loro influenza complessiva sul modello è stata potenzialmente ridotta o meglio gestita grazie alla stabilizzazione della varianza e alla linearizzazione.

**d) Test di Specificazione Formali** I test formali sono stati ripetuti sul nuovo modello con `log_trophies` per quantificare l'impatto della trasformazione:

- **Test t sulla media dei residui:** Come nel modello precedente, il test ha prodotto un p-value approssimabile a 1 ed una media campionaria dei residui di  $8.375 \times 10^{-18}$ , questo risultato conferma che la media dei residui rimane statisticamente non diversa da zero. Questa assunzione continua ad essere soddisfatta.
- **Test di Anderson-Darling per la normalità:** Il test ha ancora restituito un p-value estremamente basso ( $A=419.07$ , p-value  $< 2.2 \times 10^{-16}$ ). Nonostante il p-value significativo, che come già osservato è atteso per campioni di grandi dimensioni come il nostro, l'analisi visiva dei grafici Q-Q e dell'istogramma indica una buona approssimazione alla normalità. Per campioni così ampi, l'affidamento sull'ispezione visiva e sulla robustezza del Teorema del Limite Centrale è considerato più appropriato rispetto al p-value del test formale.
- **Test di Breusch-Pagan per l'eteroschedasticità:** Anche per questo modello, il p-value è risultato estremamente basso ( $BP=815.9$ ,  $df=302$ , p-value  $< 2.2 \times 10^{-16}$ ), suggerendo la presenza di eteroschedasticità statisticamente significativa. Tuttavia, è interessante notare che il valore della statistica BP è diminuito rispetto al modello originale (da 1874 a 816.9). Questo indica una mitigazione dell'eteroschedasticità;

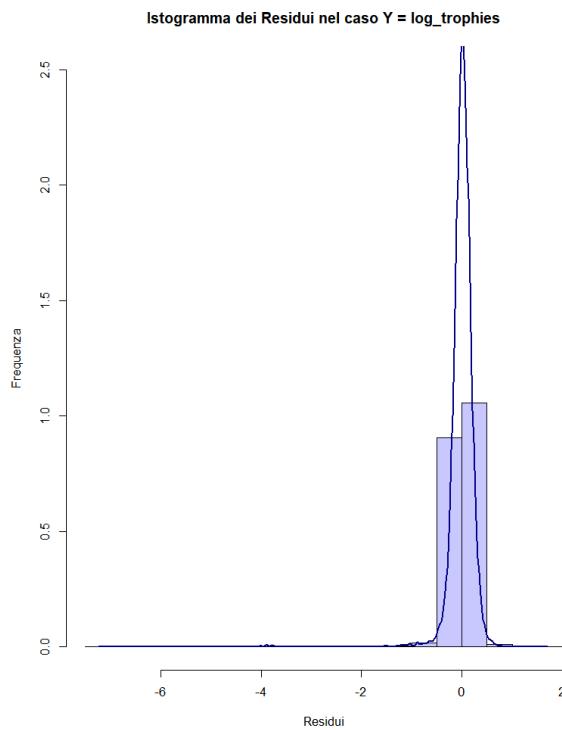


Figura 17: Istogramma dei residui

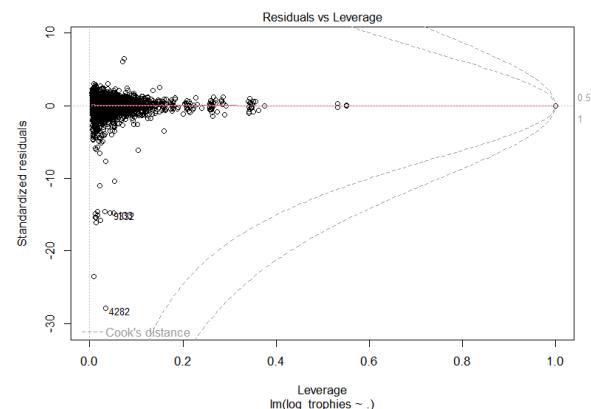


Figura 18: Residui vs. Leverage

sebbene non sia stata eliminata a livello statistico formale, la sua entità è stata ridotta, migliorando l'efficienza delle stime.

- **Test di Durbin-Watson per l'autocorrelazione:** Il test ha prodotto un valore Durbin-Watson DW=1.6881 con un p-value  $<2.2 \times 10^{-16}$ . Confrontando con il modello precedente (DW=1.2359), il valore è ora più vicino a 2, suggerendo una riduzione dell'autocorrelazione. Nonostante il p-value indichi ancora significatività, la vicinanza a 2 è un indicatore di un miglioramento nella riduzione della dipendenza seriale dei residui.

Concludiamo, dunque, che la trasformazione logaritmica della variabile `trophies` ha portato a un miglioramento apprezzabile nelle prestazioni del modello rispetto alle assunzioni della regressione lineare. Le analisi visive mostrano un netto avvicinamento alla linearità, una riduzione dell'eteroschedasticità e un'approssimazione alla normalità dei residui adeguata. Questo modello è ora in una posizione migliore per fornire stime valide e un'interpretazione significativa degli effetti delle variabili predittive.

## 4.4 Interpretazione del Modello Lineare Addestrato

Una volta addestrato il modello di regressione lineare con la variabile risposta trasformata logaritmicamente ( $\log(trophies+1)$ ), è interessante interpretare i coefficienti stimati per comprendere come le diverse variabili influenzano i trofei del giocatore. L'interpretazione dei coefficienti in un modello con variabile risposta trasformata in log richiede attenzione, poiché una variazione unitaria nella variabile predittore non corrisponde a una variazione additiva diretta nella variabile risposta originale, ma piuttosto a una variazione esponenziale.

Per un coefficiente  $\beta_i$  associato a una variabile  $X_i$ :

- Se  $X_i$  è una variabile numerica, un aumento di un'unità in  $X_i$  è associato a una variazione percentuale stimata di circa  $(\exp(\beta_i)-1) \times 100\%$  nella variabile risposta (trofei), mantenendo costanti tutte le altre variabili.

$$\begin{aligned} \log(Y + 1) &= \beta_0 + X_1\beta_1 + \dots + X_p\beta_p \Rightarrow Y + 1 = \exp(\beta_0 + X_1\beta_1 + \dots + X_p\beta_p) \\ &\Rightarrow Y + 1 = \exp(\beta_0)\exp(X_1\beta_1)\dots\exp(X_p\beta_p) \\ Y' + 1 &= \exp(\beta_0)\exp((X_1 + 1)\beta_1)\dots\exp(X_p\beta_p) = \exp(\beta_0)\exp(X_1\beta_1)\exp(\beta_1)\dots\exp(X_p\beta_p) \\ &\Rightarrow Y' + 1 = (Y + 1) \cdot \exp(\beta_1) \Rightarrow \frac{Y'+1-(Y+1)}{Y+1} = \frac{(Y+1)\cdot\exp(\beta_1)-(Y+1)}{Y+1} = \exp(\beta_1) - 1 \end{aligned}$$

- Se  $X_i$  è una variabile binaria (dummy), la variabile risposta per la categoria "true" è stimata essere  $(\exp(\beta_i)-1) \times 100\%$  più grande (o più piccola, se  $\beta_i$  è negativo) rispetto alla categoria di riferimento.

### 4.4.1 Interpretazione della Significatività dei Coefficienti

Il modello lineare addestrato con  $\log(trophies+1)$  come variabile dipendente presenta un  $R^2$  di 0.9146 e un  $R^2$  aggiustato di 0.9119. Questi valori indicano che circa il 91.19% della

variabilità nei trofei (in scala logaritmica) è spiegata dalle variabili incluse nel modello, suggerendo un'ottima capacità esplicativa. L'F-statistic del modello, pari a 344.4 con un p-value estremamente basso ( $<2.2e^{-16}$ ), conferma la significatività complessiva del modello: esso è statisticamente significativo nel predire i trofei.

Analizziamo ora i coefficienti delle variabili predittive, focalizzandoci sulla loro significatività statistica (indicata dal p-value) e sulla direzione dell'effetto (segno del coefficiente).

Variabili Altamente Significative (p-value < 0.001):

- **cardsOwned (0.01009)**: Questo è uno dei predittori più forti e positivi. Per ogni carta posseduta in più, i trofei attesi aumentano proporzionalmente di circa  $\exp(0.01009)-1 \approx 1.01\%$ . Questo suggerisce che avere un maggior numero di carte sbloccate nel gioco è fortemente associato a trofei più elevati.
- **log\_totalExpPoints (0.2491)**: Un aumento dei punti esperienza totali è associato a un aumento proporzionale significativo nei trofei. Questo è molto logico, in quanto l'esperienza è una misura della quantità di tempo e impegno dedicato al gioco.
- **log\_wins (0.2707)**: Similmente a **log\_totalExpPoints**, un aumento delle vittorie è associato a un aumento proporzionale molto forte nei trofei. Questo coefficiente ha uno dei valori più alti, riflettendo l'importanza diretta delle vittorie nell'accumulo di trofei.
- **daysSinceRegistration (-0.00002670)**: Questo coefficiente è leggermente negativo e altamente significativo. Sebbene piccolo in valore assoluto, indica che all'aumentare dei giorni dall'iscrizione, i trofei tendono a diminuire leggermente (circa 0.00267% per giorno). Questo potrebbe riflettere una tendenza dei giocatori più vecchi ad essere meno attivi.
- **challengeMaxWins (-0.004800)**: Un aumento nel numero massimo di vittorie nelle sfide è associato a una leggera diminuzione proporzionale nei trofei. Questo è un risultato inaspettato poiché ci si aspetterebbe una correlazione positiva tra successo nelle sfide e quello nei trofei. Però, difatti avere un numero alto di **challengeMaxWins** significa solo essere bravi ma per scalare nei trofei non basta, serve costanza ed un impegno più prolungato nel tempo. In sintesi una interpretazione di questo fenomeno è aver individuato una classe di giocatori occasionali che giocano le challenge ma non la ladder.
- **CardsLevel13 (-0.006195)**, **CardsLevel12 (-0.002066)**, **CardsLevel11 (-0.002291)**, **CardsLevel10 (-0.002941)**: Tutti questi coefficienti sono negativi e altamente significativi. Essi indicano che avere un maggior numero di carte a livelli *inferiori* (rispetto al livello massimo 15) è associato a un numero minore di trofei. Questo è coerente con l'idea che i giocatori con trofei più alti tendono ad avere un deck più "maxato" (portato a livello massimo).
- **SupportCardsLevel15.L (-0.2650)**: La componente lineare della variabile **SupportCardsLevel15** ha un coefficiente negativo e molto significativo (infatti il suffisso L suggerisce che **SupportCardsLevel15** viene gestita a contrasti polinomiali ortogonali). Questo indica una tendenza lineare significativa e decrescente nella relazione tra il numero di carte di supporto portate al livello 15 (il

massimo) e `log(trophies+1)`. Sembra un risultato contorto e potrebbe suggerire che i giocatori che si concentrano molto sul massimizzare solo le carte di supporto potrebbero non avere un bilanciamento ottimale nel loro deck per la scalata dei trofei. Senza contare che nel periodo in cui il Percorso dei Trofei era in auge il livello delle carte era limitato a 14.

- **NumberSupportCards.Q (0.03412), NumberSupportCards.C (-0.02810):** Le componenti quadratiche e cubiche del numero di carte di supporto sono significative e con segni opposti. Questo indica una relazione non lineare complessa tra il numero di carte di supporto e i trofei, suggerendo che un certo numero oculato di carte di supporto sia ottimale, e sia troppo poche che troppe possano essere svantaggiose (di fatti averne troppe significa aver disperso troppo le proprie risorse ed al contempo averne troppe poche significa non avere una varietà di carte sufficientemente ampia per essere competitivi).
- **log\_totalDonations (-0.02812):** Un aumento delle donazioni totali è associato a una diminuzione proporzionale nei trofei. Anche questo è contorto, poiché si potrebbe pensare che i giocatori attivi nel donare siano anche giocatori esperti. Questo è un fenomeno che mostra che l'altruismo in un gioco competitivo non è ripagato.
- **log\_tournamentBattleCount (-0.02951):** L'aumento del conteggio delle battaglie nei tornei è associato a una diminuzione proporzionale nei trofei. Questo è un altro risultato sorprendente. Potrebbe essere che i giocatori che partecipano a molti tornei trascorrono meno tempo nella scalata dei trofei analogamente a quanto detto per le challenge.
- **fourth\_root\_expPoints (-0.01427):** La trasformazione radice quarta dei punti esperienza ha un coefficiente negativo. Dato che `log_totalExpPoints` è positivo, questo suggerisce che potrebbe esserci una contaminazione dovuta alla correlazione tra le due variabili. Un'altra spiegazione potrebbe essere che `expPoints` contiene solo i punti esperienza che non sono stati usati per aumentare di livello mentre `totalExpPoints` conteggia anche quelli già spesi. Effettivamente avere pochi punti esperienza potrebbe significare che si è saliti di livello e quindi una maggiore presenza in game.
- **sqrt\_CardsEvo (-0.05881):** La radice quadrata del numero di carte evolute ha un coefficiente negativo. Questo è molto sorprendente, poiché le carte evolute sono tra le più potenti del gioco, e ci si aspetterebbe che averne di più sia associato a trofei più alti. La spiegazione più semplice è che le carte evolute siano state aggiunte al gioco da relativamente poco tempo, quindi molti giocatori hanno scalato il percorso dei trofei pur non avendo carte evolute.

Variabili Significative (p-value < 0.01 o < 0.05 ma  $\geq 0.001$ ):

- **currentFavouriteCardArchers (-0.2402) e currentFavouriteCardArrows (-0.2049):** Avere Arcieri o Frecce come carta preferita è associato a una diminuzione significativa dei trofei, questo suggerisce che l'uso di queste carte potrebbe essere più comune a livelli di trofei inferiori o in strategie meno efficaci ai livelli più alti.
- **currentFavouriteCardFireball (-0.1598):** Similmente, avere Palla di Fuoco come carta preferita è associato a una diminuzione significativa dei trofei.

- **currentFavouriteCardGiant** (-0.1209): Avere Gigante come carta preferita è associato a una diminuzione significativa dei trofei, difatti il gigante è stata una carta molto potente agli inizi del gioco ma molto depotenziata negli ultimi anni (nessuno la gioca a livello competitivo).
- **GoblinsTRUE** (0.09847), **WitchTRUE** (0.05129), **BarbariansTRUE** (0.07199), **SkeletonsTRUE** (0.06298), **ValkyrieTRUE** (0.06722), **Skeleton ArmyTRUE** (0.04716), **BomberTRUE** (0.07117), **WizardTRUE** (0.04191), **Mini P.E.K.K.A TRUE** (0.04135), **Spear GoblinsTRUE** (0.04990), **Hog RiderTRUE** (0.04103), **GuardsTRUE** (0.04756), **Dark PrinceTRUE** (0.04396), **Battle RamTRUE** (0.1344), **Mega MinionTRUE** (0.08733), **FishermanTRUE** (0.1183), **Archer QueenTRUE** (0.06312), **Electro SpiritTRUE** (0.05335), **CannonTRUE** (0.08603), **Goblin HutTRUE** (0.1212), **Inferno TowerTRUE** (0.07316), **Bomb TowerTRUE** (0.07495), **TeslaTRUE** (0.06847), **TombstoneTRUE** (0.08215), **Goblin CageTRUE** (0.09662), **ZapTRUE** (0.05135), **GraveyardTRUE** (0.07459), **Tower PrincessTRUE** (0.06796): Questi sono i coefficienti per le carte possedute (nel deck corrente) che sono significativamente e positivamente associate a trofei più alti; suggerisce che avere queste carte nel proprio arsenale (non necessariamente come preferita) è un indicatore di un pool di carte più forte e probabilmente di una maggiore versatilità strategica.
- **Goblin DrillTRUE** (-0.1401): Avere la carta **Goblin Drill** è associato a una diminuzione significativa dei trofei. Questo è un risultato interessante e potenzialmente contro-intuitivo, dato che era una carta molto forte al suo rilascio. Ad ora è una carta molto efficace ma bisogna usarla molto cautamente perché è facile sbagliare nel utilizzarla.
- **bestLeagueNumber.Q** (-0.07095): La componente quadratica del miglior numero di lega stagionale è significativa e negativa, suggerendo una relazione non lineare con i trofei. Potrebbe indicare che, dopo un certo punto, l'aumento nel miglior numero di lega non porta a un proporzionale aumento, o addirittura può causare una diminuzione, nei trofei suggerendo che un'eccessiva focalizzazione su questa modalità potrebbe non tradursi direttamente in un incremento del percorso dei trofei. Ciò significa che aumentare il miglior numero di lega è sintomo di una certa bravura che si riflette in un aumento anche nei trofei, ma perseverando nel miglioramento nel percorso delle leggende, come visto per le challenge e i tornei, causa una dispersione di risorse e tempo non favorendo la crescita dei trofei.

Variabili Marginalmente Significative (p-value < 0.1 ma  $\geq 0.01$ ):

- **ArchersTRUE** (-0.03736), **Elixir CollectorTRUE** (-0.05503), **Minion HordeTRUE** (-0.03972), **MinionsTRUE** (-0.03040), **SparkyTRUE** (0.04569), **BowlerTRUE** (0.05760), **Electro DragonTRUE** (0.05338), **Battle HealerTRUE** (0.07695), **Skeleton DragonsTRUE** (0.04731), **Giant SnowballTRUE** (0.04422), **CloneTRUE** (0.05302), **Barbarian HutTRUE** (-0.1218): Queste variabili mostrano una tendenza verso la significatività, ma con un livello di confidenza leggermente inferiore. Anche questi coefficienti, per quanto meno significativi, potrebbero portare l'attenzione nell'individuare le carte più forti nel gioco.

Variabili Non Significative:

- Molte delle variabili, in particolare la maggior parte delle carte preferite e alcune delle carte utilizzate nel deck corrente, le variabili relative ai ruoli nel clan (`roleelder`, `rolecoLeader`, `roleleader`) e diverse trasformazioni polinomiali delle leghe (`currentLeagueNumber`, `lastLeagueNumber`, `bestLeagueNumber` escluse quelle menzionate) non mostrano una significatività statistica. Ciò significa che non possiamo concludere che abbiano un impatto statisticamente significativo sui trofei. Questo non implica necessariamente che non abbiano alcun effetto, ma che il modello non è in grado di rilevarne uno con sufficiente certezza.

## 4.5 Valutazione del Modello e Selezione dei Predittori

Dopo aver addestrato il modello lineare e interpretato la significatività e l'impatto dei suoi coefficienti sulla variabile risposta  $\log(\text{trophies}+1)$ , è fondamentale valutare l'efficacia complessiva e la capacità di generalizzazione. Questa sezione è dedicata all'analisi delle metriche chiave che consentono di quantificare la bontà di adattamento del modello ai dati di training e la sua significatività statistica e la sua potenziale performance su dati non visti. Esamineremo metriche tradizionali di adattamento, test di significatività globale, criteri di selezione del modello basati sull'informazione e approcci di validazione più robusti come la cross-validation.

### 4.5.1 Metriche di Adattamento e Significatività Complessiva

Per valutare l'efficacia complessiva del modello di regressione lineare addestrato per prevedere  $\log(\text{trophies}+1)$ , esaminiamo le metriche di adattamento e il test di significatività globale.

#### R<sup>2</sup> e R<sup>2</sup> Aggiustato:

L'R<sup>2</sup> (R-quadro) è una misura della proporzione della varianza della variabile dipendente che viene spiegata dal modello. Un valore di R<sup>2</sup> di **0.9146** indica che circa il **91.46%** della variabilità nei trofei (in scala logaritmica) è spiegata dai predittori inclusi nel modello. Questo è un valore estremamente elevato, suggerendo un'ottima capacità del modello di catturare le relazioni presenti nei dati.

L'R<sup>2</sup> aggiustato (adjusted R<sup>2</sup>), con un valore di **0.9119**, tiene conto del numero di predittori nel modello e della dimensione del campione, fornendo una stima più realistica della capacità esplicativa del modello sulla popolazione. La minima differenza tra l'R<sup>2</sup> e l'R<sup>2</sup> aggiustato (solo una leggera diminuzione) rafforza la conclusione che l'aggiunta delle numerose variabili al modello è complessivamente ben giustificata e non sta portando a un overfitting eccessivo dovuto a predittori superflui. Questi valori elevati indicano un'eccellente capacità predittiva e descrittiva del modello.

#### Test F e tabella ANOVA Sintetica (Significatività Globale):

Il test F valuta la significatività complessiva del modello. L'ipotesi nulla ( $H_0$ ) è che tutti i coefficienti di regressione (eccetto l'intercetta) siano uguali a zero, implicando che nessuno dei predittori abbia un'influenza significativa sulla variabile risposta. Con un F-statistic pari a

344.43 (con gradi di libertà  $df_1=302$  e  $df_2=9717$ ) e un p-value estremamente basso ( $<2.2 \times 10^{-16}$ ), possiamo rifiutare con forza l'ipotesi nulla. Questo risultato conferma inequivocabilmente che il modello è statisticamente significativo e che almeno un sottoinsieme dei predittori inclusi ha un grande impatto sulla variabile risposta  $\log(\text{trophies}+1)$ . Per calcolare la F-statistic si può seguire la tabella ANOVA che riassume,

nella prima colonna, la variabilità totale di  $Y$   $(SS_T = \sum_{i=1}^n (Y - \bar{Y})^2)$  ripartendola in variabilità spiegata dal modello  $(SS_R = \sum_{i=1}^n (\hat{Y} - \bar{Y})^2)$  ed errore  $(SS_E = \sum_{i=1}^n (Y - \hat{Y})^2)$ . Poi nella colonna successiva ci sono i gradi di libertà di queste ultime due variabilità rilette come variabili chi-quadro. Nelle ultime due colonne vi è rispettivamente una normalizzazione rispetto ai gradi di libertà ed il calcolo della F-statistic come rapporto di queste due normalizzazioni.

Natura della variabilità	Somma dei quadrati	df	Media dei quadrati	F-statistic
Modello	$SS_R = 7147.086$	$p = 302$	$\frac{SS_R}{p} = 23.66585$	
Errore	$SS_E = 667.660$	$n-p-1 = 9717$	$\frac{SS_E}{n-p-1} = 0.0687$	$F = \frac{SS_R/p}{SS_E/(n-p-1)} = 344.4282$
Totale	$SS_T = 7814.746$	$n-1 = 10019$		

### Analisi Sequenziale ANOVA per l'Importanza dei Predittori

Per comprendere il contributo individuale e sequenziale delle variabili predittive alla varianza spiegata del modello, presentiamo la tabella ANOVA con i predittori ordinati per valore decrescente di **F value**. Questa analisi aiuta a identificare le variabili che, quando aggiunte al modello contribuiscono maggiormente a spiegare la variabilità della variabile dipendente **log\_trophies**.

Colonna 1	Df	Sum Sq	Mean Sq	F value	Pr(>F)
challengeMaxWins	1	1304,05	1304,051	18978,90739	<2.2e-16
cardsOwned	1	1105,49	1105,486	16089,02497	<2.2e-16
Giant	1	509,21	509,213	7410,986232	<2.2e-16
Archers	1	268,38	268,376	3905,899591	<2.2e-16
log_totalExpPoints	1	244,42	244,417	3557,197228	<2.2e-16
Musketeer	1	162,13	162,127	2359,564344	<2.2e-16

#### 4.5.2 Criteri di Selezione del Modello Basati sull'Informazione (AIC, BIC, Mallows' Cp)

Oltre alle metriche di adattamento e significatività complessiva, è utile valutare il modello utilizzando criteri di informazione che penalizzano la complessità del modello, favorendo un buon equilibrio tra la bontà di adattamento ai dati e il numero di parametri utilizzati. Per una valutazione più completa, confronderemo anche questi criteri per il nostro modello `model_lm_log_trophies` con quelli di un `model_lm_full` (che include tutte le variabili iniziali, eliminate a causa dell'alta VIF o altre considerazioni iniziali come `meanCostDeck`, `sq_log_battleCount` ecc.).

**Akaike Information Criterion (AIC):** L'AIC stima la qualità relativa dei modelli statistici per un dato set di dati. Un valore di AIC più basso indica un modello migliore. Per il nostro modello lineare `model_lm_log_trophies`, l'AIC calcolato è:  $AIC=1903.77$ ; per il modello con più predittori l'AIC calcolato è:  $AIC_{full} = -6995.88$ .

**Bayesian Information Criterion (BIC):** Il BIC è simile all'AIC ma impone una penalità maggiore per il numero di parametri, soprattutto per campioni di grandi dimensioni, tendendo a favorire modelli più parsimoniosi. Un valore di BIC più basso è preferibile. Per il nostro modello lineare `model_lm_log_trophies`, il BIC calcolato è:  $BIC=4096.32$ ; per il modello con più predittori il BIC calcolato è:  $BIC_{full} = -4745.63$ .

**Mallows' Cp (Cp):** Il Cp di Mallows è una statistica utilizzata per valutare l'adeguatezza di un modello di regressione (`model_lm_log_trophies`) con un sottoinsieme di predittori rispetto a un modello completo (`model_lm_full`). Il valore ideale per il Cp è approssimativamente uguale al numero di parametri ( $p$ ) nel modello che si sta valutando, suggerendo un buon equilibrio tra bias (errore dalle variabili omesse) e varianza (errore dovuto all'eccessiva complessità del modello). Per il nostro `model_lm_log_trophies` ( $p=303$  includendo l'intercetta) confrontato con `model_lm_full`, il Cp calcolato è:  $Cp=14223.6$

#### Interpretazione Congiunta dei Criteri di Informazione:

- **Confronto AIC e BIC:** Sorprendentemente, il `model_lm_full` presenta valori di AIC e BIC significativamente più bassi rispetto al nostro `model_lm_log_trophies`. Poiché un valore più basso per AIC e BIC indica un modello preferibile (che spiega meglio i dati con una data complessità o è più parsimonioso), questo suggerisce che il `model_lm_full`, nonostante abbia più parametri (311 vs 303), offre un migliore equilibrio tra adattamento e complessità secondo questi criteri.
- **Mallows' Cp:** Il valore del Cp di 14223.6, che è enormemente maggiore del numero di parametri del nostro `model_lm_log_trophies` ( $p=303$ ), è un'indicazione molto forte e coerente con quanto osservato con AIC e BIC. Un Cp così elevato suggerisce che il `model_lm_log_trophies` soffre di un bias significativo. Ciò significa che la rimozione delle variabili che ha portato dal modello completo al `model_lm_log_trophies` (anche se motivata da alta multicollinearità, come indicato dal VIF) ha comportato una perdita considerevole di informazione predittiva. In altre parole, il modello più parsimonioso (`model_lm_log_trophies`) potrebbe

essere meno accurato nelle sue previsioni a causa dell'omissione di predittori che, pur causando collinearità, contribuivano a spiegare una parte importante della varianza della variabile risposta.

Dunque l'analisi congiunta di AIC, BIC e Cp di Mallows, confrontando il `model_lm_log_trophies` con il modello completo, rivela che il nostro processo di selezione delle variabili, pur mirando a ridurre la multicollinearità e a creare un modello più parsimonioso, potrebbe aver introdotto un compromesso sfavorevole in termini di bias di previsione. I risultati suggeriscono che il `model_lm_full`, pur essendo più complesso e potenzialmente con problemi di multicollinearità, potrebbe fornire un adattamento migliore e un bias inferiore rispetto al `model_lm_log_trophies`.

Questo non invalida necessariamente il `model_lm_log_trophies` se l'obiettivo principale era la interpretabilità dei coefficienti non-collineari. Tuttavia, se l'obiettivo primario è la massima capacità predittiva con il minor errore predittivo possibile, il `model_lm_full` o un altro approccio di selezione delle variabili (ad esempio, regolarizzazione  $L^2$  o  $L^1$ ) meriterebbe un'ulteriore considerazione.

#### 4.5.3 Validazione Incrociata (Cross-Validation) per la Stima dell'Errore Predittivo

Per ottenere una stima robusta e affidabile dell'errore predittivo del modello di regressione lineare sulla variabile risposta trasformata ( $\log(\text{trophies}+1)$ ), è stata impiegata una metodologia di Validazione Incrociata (Cross-Validation) a 10 folds. Questo approccio permette di valutare la capacità di generalizzazione del modello su dati non visti, fornendo una misura più realistica delle sue prestazioni predittive rispetto alla sola valutazione sul set di addestramento. La procedura è stata implementata utilizzando la funzione `train()` del pacchetto `caret` in R, che automatizza il processo di suddivisione dei dati, addestramento del modello e raccolta delle metriche in ogni fold.

I risultati aggregati della 10-fold Cross-Validation sono i seguenti:

- RMSE medio (Root Mean Squared Error): 0.269
- $R^2$  medio (Coefficiente di Determinazione): 0.9049
- MAE medio (Mean Absolute Error): 0.1483

Per una valutazione comparativa, sono state calcolate anche le metriche di performance del modello addestrato sull'intero set di dati di training originale:

- RMSE sul Training Set: 0.2581
- MAE sul Training Set: 0.1428

#### Analisi e Interpretazione dei Risultati:

I valori di RMSE e MAE ottenuti dalla cross-validation (0.269 e 0.1483 rispettivamente) sono leggermente superiori rispetto a quelli calcolati sul solo training set (0.2581 e 0.1428). Questa è una differenza attesa e un indicatore positivo. L'errore sul training set tende a sottostimare l'errore di previsione su nuovi dati, poiché il modello si adatta specificamente ai dati su cui è stato addestrato. La cross-validation fornisce una stima più realistica delle

prestazioni del modello su dati non visti, ed è normale che questa stima sia più elevata. La piccola entità di questa differenza suggerisce che il modello non sta overfittando eccessivamente i dati di training e possiede una solida capacità di generalizzazione a nuove osservazioni.

In sintesi, i risultati della cross-validation confermano che il modello lineare sviluppato è robusto e presenta un'eccellente accuratezza predittiva per la stima dei trofei dei giocatori di Clash Royale.

Bisogna, però, osservare che gli errori stimati sono sempre in scala logaritmica e quindi vanno interpretati come percentuali. La ri-trasformazione in scala originale porta ad ottenere un MSE su dati di training particolarmente elevato: 807877.8.

Un'ottima idea per ottenere metodi maggiormente predittivi sarà la regolarizzazione.

# Capitolo 5: Selezione del Miglior Sottoinsieme di predittori (Penalizzazione L<sup>0</sup>)

Nel contesto dell'analisi predittiva, la selezione delle variabili gioca un ruolo cruciale nella costruzione di modelli robusti, interpretabili ed efficienti. L'obiettivo è identificare un sottoinsieme di predittori che sia il più informativo possibile per la variabile risposta, evitando al contempo l'overfitting e riducendo la complessità del modello. Questo approccio è spesso definito come "penalizzazione L<sup>0</sup>" o selezione del miglior sottoinsieme, poiché mira a minimizzare il numero di coefficienti diversi da zero.

In questo capitolo, esploreremo in dettaglio le metodologie di selezione del miglior sottoinsieme, confrontando due approcci principali disponibili in R: la funzione `regsubsets()` del pacchetto `leaps` e la funzione `step()`. Entrambe mirano a identificare i predittori più influenti, ma attraverso strategie di ricerca e criteri di ottimizzazione distinti.

Per le nostre analisi, utilizzeremo il dataset `new_train_data_processed` (salvato nel file `player_data_new_processed.parquet`), una versione pre-processata dei dati originali dei giocatori di Clash Royale. È importante notare che, per migliorare la robustezza e l'affidabilità dei modelli, abbiamo rimosso 10 osservazioni problematiche identificate in fasi precedenti come outlier o punti ad alta influenza. Il dataset finale per questa analisi consiste quindi in 10.010 osservazioni e 35 variabili.

Questo dataset è stato accuratamente preparato per ottimizzare la modellazione e include:

- Variabile Risposta: **trophies**, che rappresenta il punteggio di trofei attuale del giocatore e sarà il nostro target predittivo.
- Variabili Predittive (34 predittori totali gestiti nei modelli come 42):
  - 30 variabili numeriche: Questo sottoinsieme di predittori numerici è il risultato delle trasformazioni (logaritmiche, radici, elevazioni al quadrato) e di alcune delle rimozioni effettuate nel Capitolo 4, progettato per catturare relazioni non lineari e migliorare la stabilità del modello.
  - 4 variabili fattoriali:
    - **bestLeagueNumber**: Una variabile ordinale a 10 livelli, che cattura la lega più alta mai raggiunta dal giocatore. Questa verrà gestita attraverso 9 contrasti polinomiali ortogonali, permettendo al modello di catturare trend lineari e non lineari attraverso i livelli della lega.
    - **Hog Rider, Elixir Golem, Mega Knight**: Tre variabili binarie (TRUE/FALSE) che indicano la presenza di una determinata carta nel deck del giocatore.

L'obiettivo di questo capitolo sarà duplice: da un lato, identificare il sottoinsieme di variabili che offre il miglior compromesso tra capacità predittiva e parsimonia del modello; dall'altro, confrontare le scelte di selezione effettuate dai diversi algoritmi, ponendo le basi per le analisi di regolarizzazione più avanzate che verranno esplorate nel Capitolo 6.

## 5.1 Selezione del Sottoinsieme di Variabili (Best Subset Selection con `regsubsets`)

La selezione del miglior sottoinsieme è una tecnica che mira a identificare il miglior modello per ogni possibile numero di predittori. Questo approccio garantisce di trovare il sottoinsieme di variabili più performante per una data dimensione del modello, basandosi su criteri come l'RSS (Residual Sum Squares), R<sup>2</sup> (Coefficiente di Determinazione), R<sup>2</sup> aggiustato, il Cp di Mallows e il BIC (Bayesian Information Criterion).

### 5.1.1 Metodologia

La funzione `regsubsets()` dalla libreria `leaps` è stata utilizzata per esplorare i sottoinsiemi di variabili. Per impostazione predefinita, `regsubsets()` considera un numero limitato di variabili, tuttavia in questo studio il parametro `nvmax` è stato impostato a `42` per considerare tutti i possibili predittori presenti nel dataset `new_train_data_processed`. Questo ha permesso di analizzare il "miglior" modello per ogni dimensione, da 1 fino a 42 variabili.

### 5.1.2 Risultati e Interpretazione

#### Selezione delle Variabili per Dimensione del Modello:

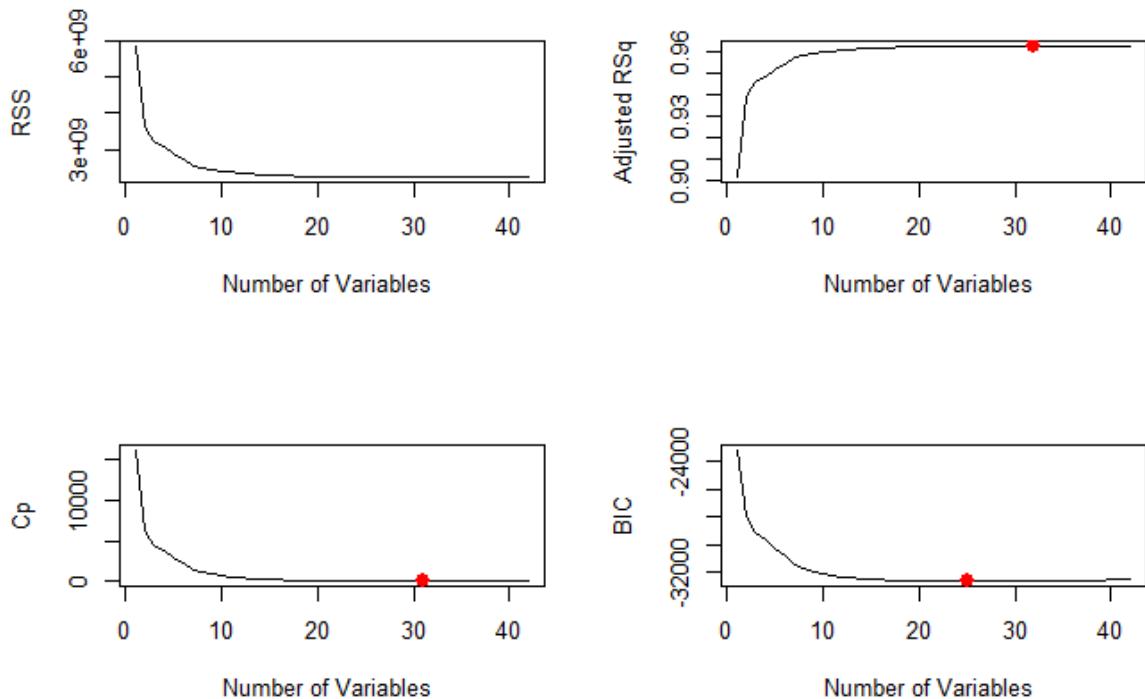
Eseguendo `regsubsets()` con la modalità `exhaustive` addestra tutti i  $2^{42}$  modelli possibili e restituisce il migliore per ogni dimensione dell'insieme dei predittori. Il risultato di questo addestramento è stato definito come `model.full`.

L'analisi della matrice booleana generata da `summary(model.full)$which` rivela la progressione dell'inclusione delle variabili nei modelli ottimali per ogni dimensione:

- **Modello con 1 variabile:** Il predittore più influente, selezionato per primo, è `cardsOwned`. Questo suggerisce una forte relazione iniziale tra il numero di carte possedute e i trofei del giocatore.
- **Modello con 2 variabili:** Oltre a `cardsOwned`, viene aggiunta `bestLeagueNumber.L`. Questo indica l'importanza della componente lineare della variabile polinomiale `bestLeagueNumber` nel prevedere i trofei.
- **Modello con 3 variabili:** Viene sostituita `bestLeagueNumber.L` con `bestLeagueNumber.Q` (componente quadratica di `bestLeagueNumber`, suggerendo una relazione non lineare con i trofei) ed aggiunta `log_CardsLevel15`.
- **Modello con 4 variabili:** `log_CardsLevel15` viene sostituita con `log_CardsLevel14`, torna la componente lineare al posto della quadratica di `bestLeagueNumber` e si aggiunge al modello `log_totalDonations`, sottolineando l'importanza delle donazioni totali.
- **Modello con 5 variabili:** `log_CardsLevel15` e `log_CardsLevel14` restano entrambe e di tutte le precedenti l'unica assente è `bestLeagueNumber.Q` che comunque tornerà dopo.

Man mano che il numero di variabili aumenta, `regsubsets()` introduce progressivamente altri predittori, come `log_threeCrownWins`, `log_wins`, `CardsLevel10`, `log_starPoints` (modello a 10 variabili includendo le già nominate 6). Il processo continua fino al modello con 42 variabili che include tutti i predittori disponibili.

### Andamento delle Statistiche di Adattamento:



L'esame delle statistiche di adattamento per ogni dimensione del modello fornisce indicazioni cruciali sulla qualità del modello:

- **Coefficiente di Determinazione:** Il valore di  $R^2$  aumenta monotonicamente all'aumentare del numero di variabili nel modello. Inizia da un valore di circa 0.9017 per il modello con una variabile e raggiunge oltre 0.9627 quando tutte le 42 variabili sono incluse. Sebbene un  $R^2$  più alto indichi una maggiore varianza spiegata, questo criterio da solo non è sufficiente per la selezione del modello, poiché tende a favorire modelli più complessi anche se le variabili aggiunte non apportano un miglioramento sostanziale.
- **Residual Sum Squares:** Coerentemente con l' $R^2$ , l'RSS diminuisce monotonicamente all'aumentare delle variabili. Parte da circa 5.86 miliardi per il modello con una variabile e scende a circa 2.22 miliardi per il modello completo. Anche l'RSS, da solo, non penalizza la complessità e quindi non è ideale per selezionare modelli parsimoniosi.
- **Adjusted R<sup>2</sup>:** Questo criterio penalizza l'aggiunta di predittori non significativi. L'R<sup>2</sup> aggiustato aumenta rapidamente, raggiunge un picco e poi tende a stabilizzarsi o a diminuire leggermente. Il valore massimo di  $R^2$  aggiustato osservato è di circa 0.9625793, raggiunto con un modello che include 32 variabili. Questo suggerisce un buon equilibrio tra capacità esplicativa e parsimonia.
- **Cp di Mallows:** Il Cp di Mallows è un criterio che cerca di bilanciare il bias e la varianza del modello. L'obiettivo è trovare un modello in cui il Cp sia circa uguale al

numero di parametri ( $p+1$ , dove  $p$  è il numero di predittori). Il valore minimo di  $C_p$  osservato è di circa 26.63481, ottenuto con un modello che include 31 variabili. Questo modello è un forte candidato, indicando un buon compromesso tra aderenza ai dati e semplicità.

- **Bayesian Information Criterion:** Il BIC impone una penalità maggiore per la complessità del modello, favorendo soluzioni più parsimoniose. Il valore minimo di BIC osservato è di circa -32648.37, raggiunto con un modello che include 25 variabili. Questo criterio indica un modello significativamente più snello rispetto a quelli suggeriti da  $R^2$  aggiustato e C\_p.

### **Coefficienti di Esempio (Modello con 10 Variabili):**

Per illustrare la composizione di uno dei modelli selezionati, di seguito sono riportati i coefficienti stimati per il modello con 10 variabili:

(Intercept)	cardsOwned	CardsLevel10	log_CardsLevel14
460.872752	58.472605	6.585795	320.623604
log_CardsLevel15	log_totalDonations	log_threeCrownWins	log_starPoints
347.939144	-55.828422	-386.984182	-81.988862
log_wins	estLeagueNumber.L	bestLeagueNumber.Q	
426.565230	624.829411	-442.240555	

Questi coefficienti indicano l'impatto stimato di ciascun predittore sui trofei, mantenendo costanti le altre variabili del modello. Ad esempio, `cardsOwned` e `log_wins` mostrano un effetto positivo significativo sui trofei, mentre `log_totalDonations` e `log_threeCrownWins` mostrano un effetto negativo in questo specifico sottoinsieme.

In sintesi, la selezione del miglior sottoinsieme offre una gamma di modelli "ottimali" a seconda del criterio scelto, variando da 25 a 32 variabili.

CardsLevel12	*	*	*	*	*	*	*	*
CardsLevel11	*	*	*	*	*	*	*	*
CardsLevel10	*	*	*	*	*	*	*	*
lastLeagueTrophies							*	*
bestLeagueTrophies					*	*	*	*
log_CardsLevel14	*	*	*	*	*	*	*	*
log_CardsLevel15	*	*	*	*	*	*	*	*
log_challengeCardsWon								
log_clanCardsCollected								
log_totalDonations	*	*	*	*	*	*	*	*
log_donations	*	*	*	*	*	*	*	*
log_threeCrownWins	*	*	*	*	*	*	*	*
log_starPoints	*	*	*	*	*	*	*	*
log_totalExpPoints					*	*	*	*
log_tournamentBattleCount	*	*	*	*	*	*	*	*
log_wins	*	*	*	*	*	*	*	*
sq_log_battleCount				*	*	*	*	*
fourth_root_expPoints	*	*	*	*	*	*	*	*
fourth_root_losses	*	*	*	*	*	*	*	*
sqrt_CardsEvo	*	*	*	*	*	*	*	*
sqrt_yearsSinceRegistration	*	*	*	*	*	*	*	*
sq_meanLevelCards	*	*	*	*	*	*	*	*
sq_meanLevelSupportCards		*	*	*	*	*	*	*
bestLeagueNumber.L	*	*	*	*	*	*	*	*
bestLeagueNumber.Q	*	*	*	*				
bestLeagueNumber.C					*	*	*	*
bestLeagueNumber^4	*	*	*	*				
bestLeagueNumber^5						*	*	*
bestLeagueNumber^6								
bestLeagueNumber^7	*	*	*	*	*	*	*	*
bestLeagueNumber^8								
bestLeagueNumber^9								
`Hog Rider`TRUE			*	*	*	*	*	*
`Elixir Golem`TRUE								
`Mega Knight`TRUE	*	*	*	*	*	*	*	*

## 5.2 Selezione Step (Basata su AIC)

La selezione step è un approccio che costruisce il modello passo dopo passo, aggiungendo o rimuovendo predittori in base a un criterio di informazione, tipicamente l'AIC (Akaike Information Criterion). Questo metodo "greedy" non garantisce di trovare il modello globalmente migliore, ma è computazionalmente più efficiente rispetto alla selezione del miglior sottoinsieme esaustiva, specialmente con un gran numero di predittori.

### 5.2.1 Metodologia (step)

La funzione `step()` è stata applicata al modello lineare completo (`model.m`) utilizzando diverse direzioni di ricerca:

- **direction = "backward"**: Inizia dal modello completo e rimuove iterativamente le variabili che aumentano maggiormente l'AIC.
- **direction = "forward"**: Inizia dal modello nullo (solo intercetta) e aggiunge iterativamente le variabili che diminuiscono maggiormente l'AIC.
- **direction = "both" (stepwise)**: Combina entrambi gli approcci, aggiungendo o rimuovendo variabili in ogni passaggio per ottimizzare l'AIC.

La funzione `step()` restituisce l'ultimo modello addestrato, cioè il modello a partire dal quale non è stato possibile ottenere un miglioramento ulteriore dell'AIC attraverso l'aggiunta o la rimozione di variabili.

### 5.2.2 Risultati e Interpretazione

#### Modelli Selezionati:

- **Selezione all'indietro**: Questo processo ha portato a un modello finale con 36 variabili predittive (37 parametri contanto l'intercetta). Il modello esclude alcune delle variabili presenti nel modello completo originale, indicando che la loro rimozione ha portato a una riduzione dell'AIC. L'R<sup>2</sup> per questo modello è di 0.9627, e l'R<sup>2</sup> aggiustato è di 0.9626.
- **Selezione in avanti**: Sorprendentemente, la selezione in avanti ha identificato il modello completo con tutte le 42 variabili come il migliore secondo il criterio AIC in questa direzione di ricerca. Ciò suggerisce che ogni variabile ha contribuito a ridurre l'AIC quando aggiunta, e che il modello completo era il punto di convergenza ottimale per questo percorso greedy. L'R<sup>2</sup> e l'R<sup>2</sup> aggiustato sono rispettivamente 0.9627 e 0.9626.
- **Selezione Bidirezionale**: Il modello risultante dalla selezione bidirezionale è identico a quello di selezione all'indietro, anch'esso con 36 variabili. Le metriche di performance (R<sup>2</sup>=0.9627, R<sup>2</sup> aggiustato =0.9626) sono le stesse del modello di selezione all'indietro.

#### Confronto AIC:

Il confronto diretto dei valori AIC per i tre modelli stepwise è il criterio principale per valutarli:

Modello	gradi di libertà	numero di predittori	AIC
Selezione all'indietro	38	36	151723.7
Selezione in avanti	44	42	151731.5
Selezione Bidirezionale	38	36	151723.7

- **backward e stepwise:** Entrambi i modelli hanno il valore AIC più basso, indicando che sono i preferiti dall'algoritmo `step()` tra le opzioni esplorate. Questi modelli contengono 36 variabili (corrispondenti a 37 parametri inclusa l'intercetta ed a 38 gradi di libertà come riportato dalla tabella).
- **forward:** Nonostante sia il modello più complesso (42 variabili corrispondenti a 43 parametri intercetta inclusa ed a 44 gradi di libertà), il suo AIC è leggermente superiore. Questo suggerisce che la maggiore complessità del modello completo non è giustificata, secondo il criterio AIC.

In conclusione, l'algoritmo `step()` basato su AIC privilegia i modelli più parsimoniosi con 36 variabili, ritenendoli migliori rispetto al modello completo.

### 5.3 Considerazioni di selezione

L'analisi condotta attraverso `regsubsets` e `step` ha fornito una visione completa delle strategie per l'identificazione di modelli ottimali nella previsione dei trofei dei giocatori.

- **Diversità dei Modelli Ottimali:** È emerso chiaramente che la scelta del "miglior" modello può variare significativamente a seconda del criterio di selezione adottato:
  - `regsubsets()` con R<sup>2</sup> aggiustato ha indicato un modello con 32 variabili.
  - `regsubsets()` con Cp di Mallows ha suggerito un modello con 31 variabili.
  - `regsubsets()` con BIC ha favorito un modello più parsimonioso con 25 variabili.
  - `step()` basato su AIC ha identificato un modello con 36 variabili (sia in direzione backward che both), mentre la selezione forward ha mantenuto tutte le 42 variabili.
- **Compromesso tra Complessità e Performance:** Ogni criterio bilancia in modo diverso la capacità del modello di adattarsi ai dati e la sua complessità. Il BIC, ad esempio, penalizza maggiormente i modelli complessi, portando a scelte più conservative in termini di numero di predittori. Al contrario, l'AIC e l'R<sup>2</sup> aggiustato tendono a selezionare modelli leggermente più ricchi di variabili, pur mantenendo un occhio alla parsimonia.
- **Algoritmi Greedy vs. Esaustivi:** È importante ricordare che `step()` è un algoritmo "greedy" che non garantisce di trovare il modello globalmente migliore, ma piuttosto il migliore all'interno del percorso esplorato; `regsubsets()`, d'altra parte, è più esaustivo (entro i limiti di `nvmax`) e tende a identificare i veri migliori sottoinsiemi per ogni dimensione.

La scelta definitiva del modello dipenderà dagli obiettivi specifici dell'analisi. Se l'interpretabilità e la parsimonia sono prioritarie, un modello con meno variabili (come quello suggerito dal BIC o dai metodi stepwise) potrebbe essere preferibile. Se l'obiettivo è massimizzare la capacità predittiva e la dimensione del modello non è una restrizione stringente, si potrebbe optare per un modello più completo o quello che massimizza l'R<sup>2</sup> aggiustato.

## Capitolo 6: Regressione Penalizzata (Ridge, Lasso, Elastic Net)

Questo capitolo approfondisce l'applicazione di modelli di regressione penalizzata. A differenza della selezione di sottoinsiemi (subset selection) o della regressione OLS, i metodi penalizzati introducono un termine di regolarizzazione nella funzione di perdita, che permette di controllare la complessità del modello, prevenire l'overfitting e gestire la multicollinearità tra i predittori. Questo approccio è spesso definito come "penalizzazione" proprio perché introduce questo termine aggiuntivo nella funzione di costo per limitare la dimensione dei coefficienti del modello.

Esploreremo le metodologie di regressione penalizzata concentrandoci su tre approcci principali: Ridge (con penalizzazione  $L^2$ ), Lasso (con penalizzazione  $L^1$ ) ed Elastic Net (una combinazione delle due). Ognuno di questi metodi offre strategie distinte per la gestione dei predittori, influenzando la stima e la selezione dei coefficienti.

Per le nostre analisi, utilizzeremo il dataset `new_train_data_processed` (utilizzato anche nel capitolo precedente ed a cui sono state rimosse 3 ulteriori variabili dummy ed è stata aggiunta una variabile numerica interpretabile come contrasto lineare di una variabile fattoriale). In questo dataset, per migliorare la robustezza e l'affidabilità dei modelli, abbiamo rimosso 10 osservazioni problematiche identificate in fasi precedenti come outlier o punti ad alta influenza. Il dataset finale per questa analisi consiste quindi in 10010 osservazioni e 33 variabili.

Questo dataset è stato accuratamente preparato per ottimizzare la modellazione e include:

- Variabile Risposta: `trophies`.
- Variabili Predittive (32 predittori totali gestiti nei modelli come 34):
  - 31 variabili numeriche: predittori numerici, risultato delle trasformazioni matematiche e di alcune delle rimozioni effettuate nel Capitolo 5.
  - `bestLeagueNumber`: Una variabile fattoriale ordinata a 10 livelli, che cattura la lega più alta mai raggiunta dal giocatore. Una differenza sostanziale introdotta in questo capitolo è che per ottimizzare l'interpretazione e la stabilità dei modelli, questa variabile verrà gestita attraverso soli 3 contrasti polinomiali ortogonali (lineare, quadratico, cubico), permettendo ai modelli di catturare i trend più significativi attraverso i livelli della lega senza introdurre eccessiva complessità. Per quanto possa sembrare riduttivo non lo è in quanto abbiamo già verificato nei modelli precedenti che gli altri livelli di contrasto non sono risultati significativi.

È importante notare che i termini espansi per la variabile fattoriale (i 3 contrasti polinomiali per `bestLeagueNumber`) non verranno soggetti a penalizzazione nei modelli Ridge, Lasso ed Elastic Net, impostando il loro `penalty.factor` a `0`. Questa scelta mira a preservare la loro influenza e la coerenza della loro interpretazione all'interno del modello, evitando che i loro coefficienti vengano ristretti verso lo zero dalla penalizzazione.

Rispetto alle analisi condotte nel Capitolo 5, dal dataset sono state rimosse le tre variabili dummy `Hog Rider`, `Elixir Golem` e `Mega Knight`. Questa scelta è stata motivata da due ragioni principali:

1. **Scarsa utilità:** Nei modelli precedenti queste variabili dummy non sono risultate molto significative, quindi le sacrificiamo in favore di una complessità inferiore.
2. **Potenziale Disturbo nell'Interpretazione:** Data la decisione di non penalizzare le variabili fattoriali queste hanno dei coefficienti che tendono a crescere con il crescere della penalizzazione stessa, comportamento innaturale limitato rimuovendo quante più variabili fattoriali possibile.

L'obiettivo di questo capitolo sarà duplice:

- applicare e comprendere le differenze tra le penalizzazioni  $L^1$ ,  $L^2$  ed Elastic Net nella stima dei coefficienti e nella selezione delle variabili;
- confrontare l'efficacia predittiva e la parsimonia dei modelli ottenuti, ponendo le basi per la scelta del modello finale e la sua valutazione nel Capitolo 7.

## 6.1 La routine `glmnet`

Il cuore di questa analisi risiede nell'utilizzo della funzione `glmnet()`, che permette di adattare modelli di regressione lineare con diverse forme di penalizzazione. Il comportamento specifico del modello è governato dal parametro alpha:

- **Regressione Ridge ( $\alpha = 0$ ):** Aggiunge una penalità  $L^2$  (somma dei quadrati dei coefficienti) alla funzione di perdita. L'effetto principale è quello di "restringere" i coefficienti verso zero, riducendone la varianza e gestendo efficacemente la multicollinearità. I coefficienti non vengono mai azzerati completamente.
- **Regressione Lasso ( $\alpha = 1$ ):** Aggiunge una penalità  $L^1$  (somma dei valori assoluti dei coefficienti). La penalità  $L^1$  ha la caratteristica unica di azzerare completamente i coefficienti di predittori meno rilevanti, realizzando così una selezione automatica delle variabili.
- **Elastic Net ( $0 < \alpha < 1$ ):** Combina le penalità  $L^1$  e  $L^2$ . Questa combinazione offre i vantaggi di entrambi i metodi: la capacità di selezione delle variabili del Lasso e la stabilità e l'effetto di raggruppamento (grouping effect) dei coefficienti del Ridge. Per questa analisi, è stato utilizzato un valore di alpha=0.5.

Per la costruzione dei modelli, è stata creata una matrice di disegno `X` dalla quale è stata rimossa l'intercetta, in quanto `glmnet` gestisce autonomamente l'intercetta. La matrice di disegno è stata creata dal dataset `new_train_data_processed`. Per l'analisi, `X` include:

- Le 31 variabili numeriche pre-processate.
- La variabile fattoriale ordinata `bestLeagueNumber`, espansa tramite `poly(bestLeagueNumber, 3)` per includere i suoi primi tre contrasti polinomiali ortogonali (lineare, quadratico, cubico).

La formula `trophies ~ . - bestLeagueNumber + poly(bestLeagueNumber, 3)`, data in input a `model.matrix()`, ha generato una matrice X con 34 predittori (escludendo l'intercetta). Il vettore Y rappresenta la variabile risposta `trophies`.

Un aspetto cruciale di questa implementazione è l'utilizzo del parametro `penalty.factor` nella funzione `glmnet()`. Le variabili derivate da `bestLeagueNumber` sono state escluse dalla penalizzazione impostando il loro `penalty.factor` a `0`, tutti gli altri 31 predittori numerici sono stati invece soggetti a penalizzazione (`penalty.factor = 1`).

È stata generata una griglia di valori per il parametro di regolarizzazione  $\lambda$  (composta da 100 valori da  $e^{-5}$  a  $e^{20}$ ) per esplorare un ampio spettro di intensità di penalizzazione. La selezione del lambda ottimale per ciascun modello è stata effettuata tramite cross-validation, come descritto nelle sezioni seguenti.

## 6.2 Regressione Ridge ( $\alpha=0$ )

La regressione Ridge introduce una penalizzazione  $L^2$  (proporzionale alla somma dei quadrati dei coefficienti) alla funzione di perdita dei minimi quadrati, con l'obiettivo di ridurre la varianza del modello e gestire la multicollinearità.

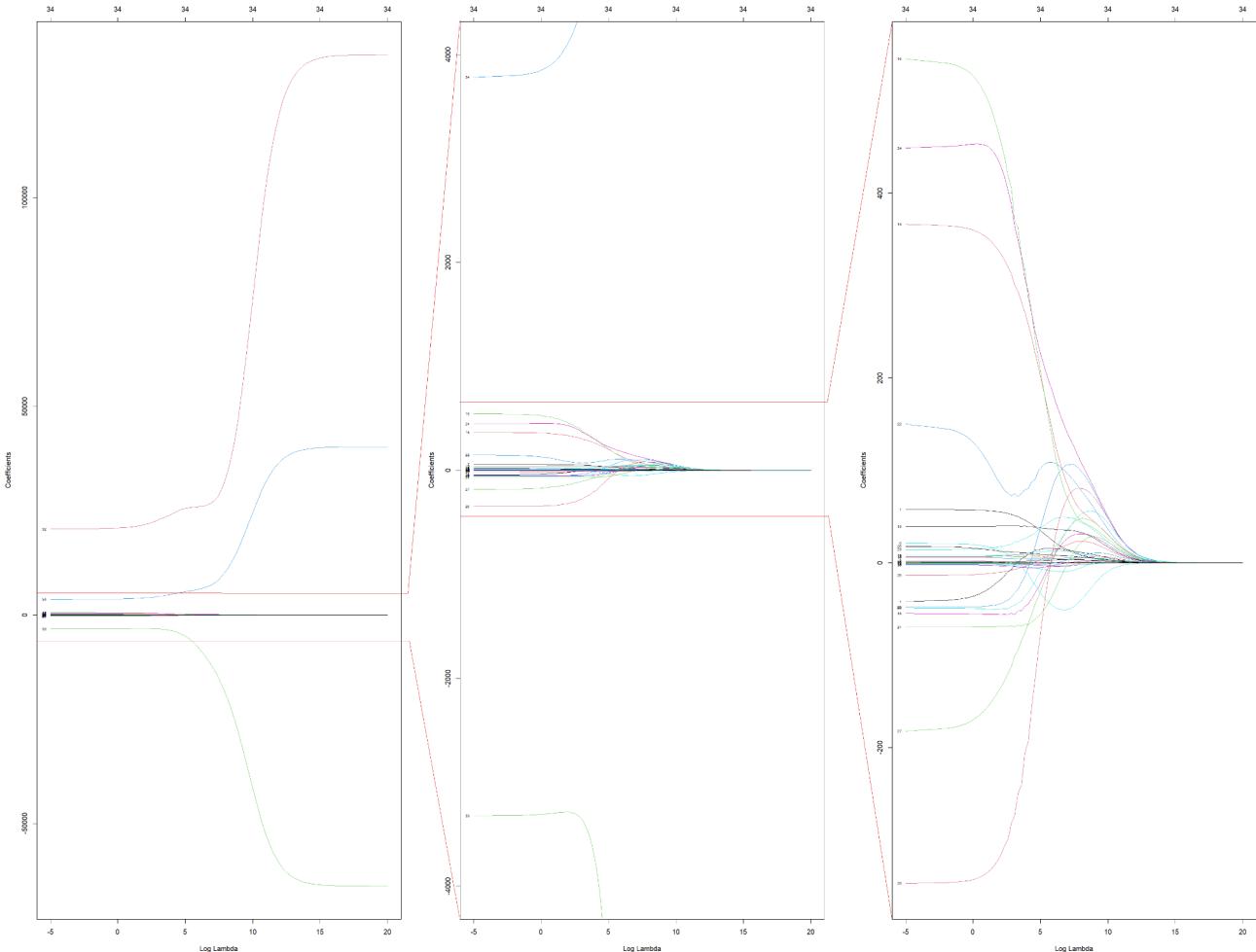


Figura 19: Evoluzione dei coefficienti della Regressione Ridge al variare di  $\lambda$

Il modello Ridge è stato adattato utilizzando la griglia di  $\lambda$  e il `penalty.factor` specificati. L'analisi del percorso dei coefficienti (Figura 19) mostra come i coefficienti dei predittori penalizzati si riducano asintoticamente verso zero all'aumentare di  $\lambda$  (spostandosi verso destra nel grafico `log(lambda)`). Nessuno di essi raggiunge mai esattamente zero, che è la caratteristica distintiva della regressione Ridge: esegue uno "shrinkage" (contrazione) ma non una vera e propria selezione delle variabili.

Nel grafico che illustra l'evoluzione dei coefficienti si può osservare un'azione di gruppo, dove variabili correlate tra loro (ad esempio `log_CardsLevel14`, `log_CardsLevel15`, `log_wins`) vengono rimpicciolate insieme, mantenendo le loro relazioni relative.

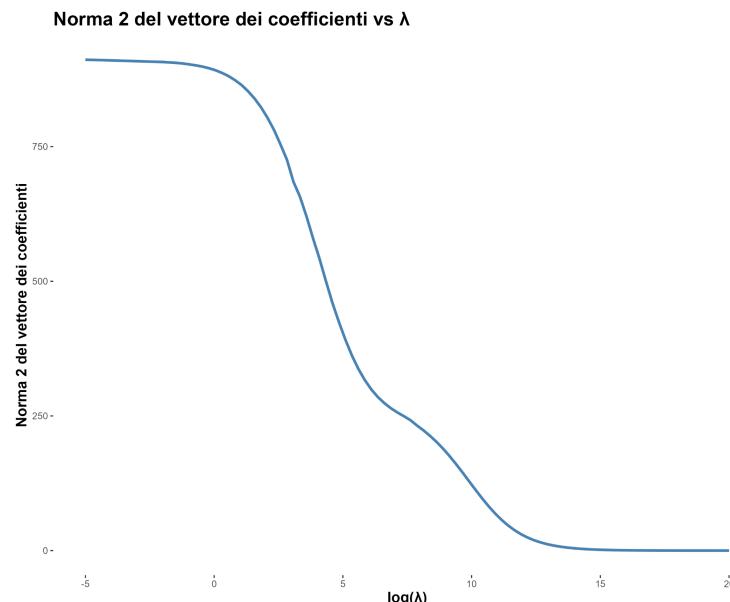


Figura 20: Norma  $L^2$  del vettore dei coefficienti penalizzati vs  $\log(\lambda)$  per la Regressione Ridge

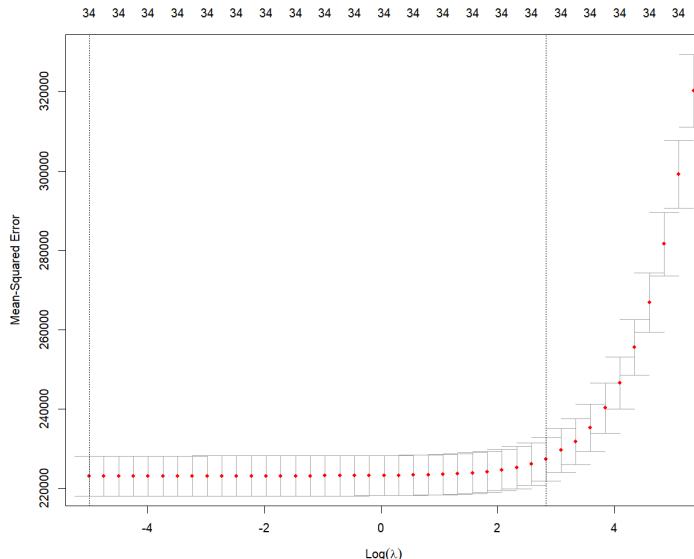
Il comportamento di shrinkage è confermato dall'andamento della norma L<sup>2</sup> dei coefficienti. Come mostrato nel grafico della Norma 2 vs lambda (Figura 20), all'aumentare di lambda (maggiore regolarizzazione), la norma 2 dei coefficienti diminuisce, indicando una minore penalizzazione e coefficienti più grandi.

Per valutare le prestazioni del modello Ridge e per identificare il valore ottimale di lambda, è stata eseguita una cross-validation a 10 fold utilizzando (`cv.glmnet`). L'output della cross-validation fornisce informazioni chiave:

	Lambda	Index	Measure	SE	Nonzero
min	0.0067	100	223119	5062	34
1se	16.91639	69	227418	5494	34

- **`lambda.min = 0.0067`**: Questo è il valore di lambda che minimizza l'errore quadratico medio (MSE) di cross-validation. Corrisponde al modello che fornisce la migliore performance predittiva sulla griglia testata, il modello in corrispondenza di questo  $\lambda$  ha mostrato un MSE di 223119.

- **lambda.1se = 16.916**: Questo valore di lambda identifica il modello più semplice (con maggiore regolarizzazione, quindi coefficienti più vicini allo zero) il cui MSE rientra in una deviazione standard dall'MSE minimo. Viene spesso preferito per bilanciare l'accuratezza predittiva con la parsimonia del modello.



*Figura 21: MSE di Cross-Validazione per la Regressione Ridge*

Il grafico di cross-validazione (Figura 21) mostra l'andamento dell'MSE medio di cross-validazione in funzione di `log(lambda)` con le barre di errore che indicano l'incertezza intorno alla stima dell'MSE (un intervallo centrato in MSE e di raggio l'errore standard). Le linee verticali tratteggiate indicano `lambda.min` e `lambda.1se`.

Per valutare ulteriormente le prestazioni del modello Ridge, è stato utilizzato un set di validazione separato (operando una divisione randomica di `X` ed `Y` con uno stesso sistema di indici di riga, così il 50% dei dati va in `X.train` ed `Y.train`, mentre il restante 50% va in `X.validation` ed `Y.validation`). Il modello è stato addestrato sul set di training (`X.train`, `Y.train`). L'MSE sul set di validazione, utilizzando `lambda.min` ottenuto dalla cross-validation, è risultato essere: 225852.7. Questo valore è molto vicino all'MSE minimo stimato dalla cross-validation (223119), confermando la robustezza della cross-validation nello stimare l'errore di predizione. Confrontando con un lambda arbitrario ( $s=4$ , MSE = 226041), la scelta basata su CV ha prodotto un leggero miglioramento.

I coefficienti finali del modello Ridge, calcolati sull'intero dataset utilizzando `lambda.1se` (per un modello più robusto, sebbene meno performante predittivamente di `lambda.min`), mostrano che nessun predittore è stato escluso (tutti i 34 coefficienti sono non nulli). I coefficienti sono stati ridotti rispetto a un modello OLS standard, ma mantengono la loro rilevanza. Ad esempio, `log_CardsLevel15` e `log_CardsLevel14` mostrano coefficienti positivi significativi, mentre `log_threeCrownWins` e `log_starPoints` presentano coefficienti negativi.

### 6.3 Regressione Lasso ( $\alpha=1$ )

La regressione LASSO (Least Absolute Shrinkage and Selection Operator) introduce una penalizzazione  $L^1$  (proporzionale alla somma dei valori assoluti dei coefficienti). A differenza della Ridge, il LASSO ha la capacità di azzerare completamente i coefficienti dei predittori meno rilevanti, realizzando così una selezione automatica delle variabili.

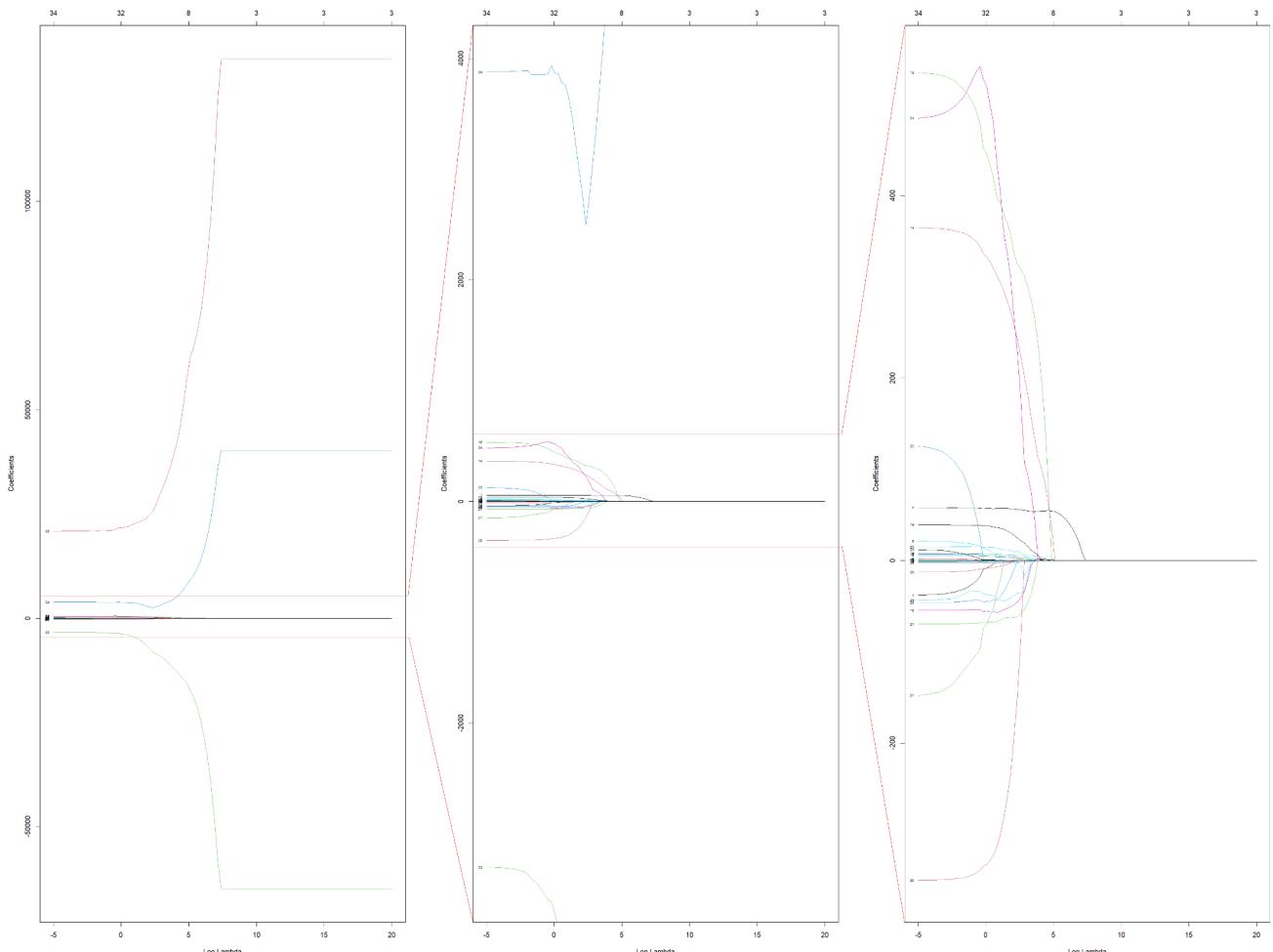
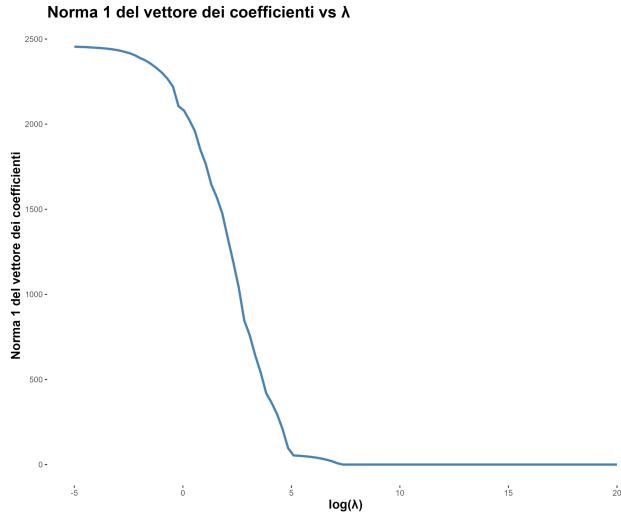


Figura 22: Percorso dei Coefficienti per la Regressione Lasso

La regressione Lasso è stata implementata con `alpha = 1` in `glmnet()`, mantenendo lo stesso `penalty.factor` per le variabili fattoriali. L'analisi del percorso dei coefficienti (Figura 22) mostra che, all'aumentare di  $\lambda$ , alcuni coefficienti vengono spinti esattamente a zero, indicando che le corrispondenti variabili vengono escluse dal modello. Ciò conferisce al LASSO un vantaggio rispetto alla Ridge, infatti facendo selezione delle variabili restituisce un modello più parsimonioso. Con il LASSO, però, si perde l'azione di gruppo osservata nella Ridge; infatti, se due variabili sono correlate, il Lasso tende a selezionarne solo una e ad azzerare l'altra.

L'andamento della norma  $L^1$  dei coefficienti, mostrato nella Figura 23, conferma come la somma dei valori assoluti dei coefficienti diminuisca all'aumentare di lambda, fino a quando la maggior parte dei coefficienti non diventa zero.



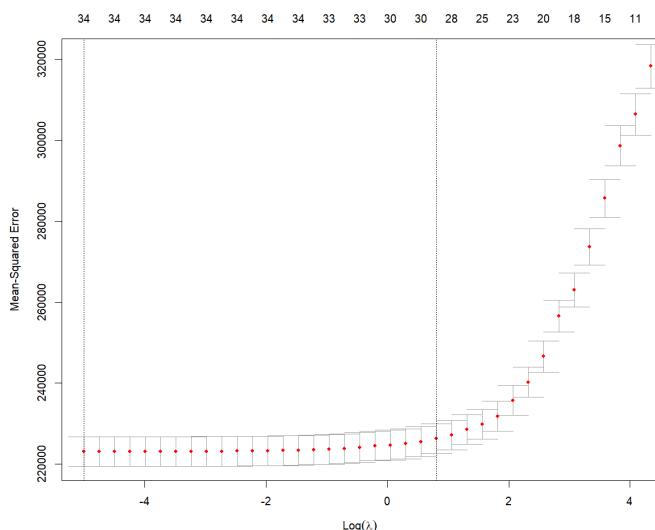
*Figura 23: Norma  $L^1$  del vettore dei coefficienti penalizzati vs  $\log(\lambda)$  per la Regressione Lasso*

Un'ispezione dei coefficienti per un lambda specifico (es. `coef(lasso.mod, s = 10)`) conferma che diverse variabili sono state azzerate, tra cui `expLevel`, `CardsLevel13`, `log_totalExpPoints`, `sq_log_battleCount` e altre.

Anche per il modello Lasso, per valutare le prestazioni e per identificare il valore ottimale di lambda è stata eseguita una cross-validation a 10 fold utilizzando (`cv.glmnet`). L'output della cross-validation fornisce informazioni chiave:

	Lambda	Index	Measure	SE	Nonzero
min	0.0067	100	223092	3680	34
1se	2.2436	77	226368	3634	29

- **`lambda.min = 0.0067`**: Valore di lambda che minimizza l'MSE (ma che non fa selezione perché troppo piccolo).
- **`lambda.1se = 2.2436`**: Valore di lambda che produce un modello più semplice con un MSE entro una deviazione standard dal minimo. Con questo lambda il numero di coefficienti non nulli (intercetta esclusa) è 29, indicando una selezione di variabili.



*Figura 24: MSE di Cross-Validazione per la Regressione Lasso*

Il grafico di cross-validation (Figura 24), analogo a quello del modello Ridge, mostra l'andamento dell'MSE medio di cross-validation in funzione di `log(lambda)`.

Anche per il modello Lasso è stata fatta validazione addestrando sul set di training e calcolando l'errore sul set di validazione usando `lambda.min`. L'MSE sul set di validazione è 226091, vicino al valore di MSE della cross-validation (223092). Inoltre l'MSE di validazione è molto simile a quello ottenuto con la Ridge ma col vantaggio del Lasso di selezione delle variabili.

Estraendo i coefficienti del Lasso addestrato con `lambda.1se` si nota che 5 coefficienti sono nulli. Le variabili che sono state azzerate sono `daysSinceRegistration`, `CardsLevel13`, `log_challengeCardsWon`, `log_totalExpPoints`, e `sq_log_battleCount`. Questo dimostra l'efficacia del LASSO nella selezione automatica delle variabili, producendo un modello più interpretabile e potenzialmente meno incline all'overfitting.

## 6.4 Regressione Elastic Net ( $\alpha=0.5$ )

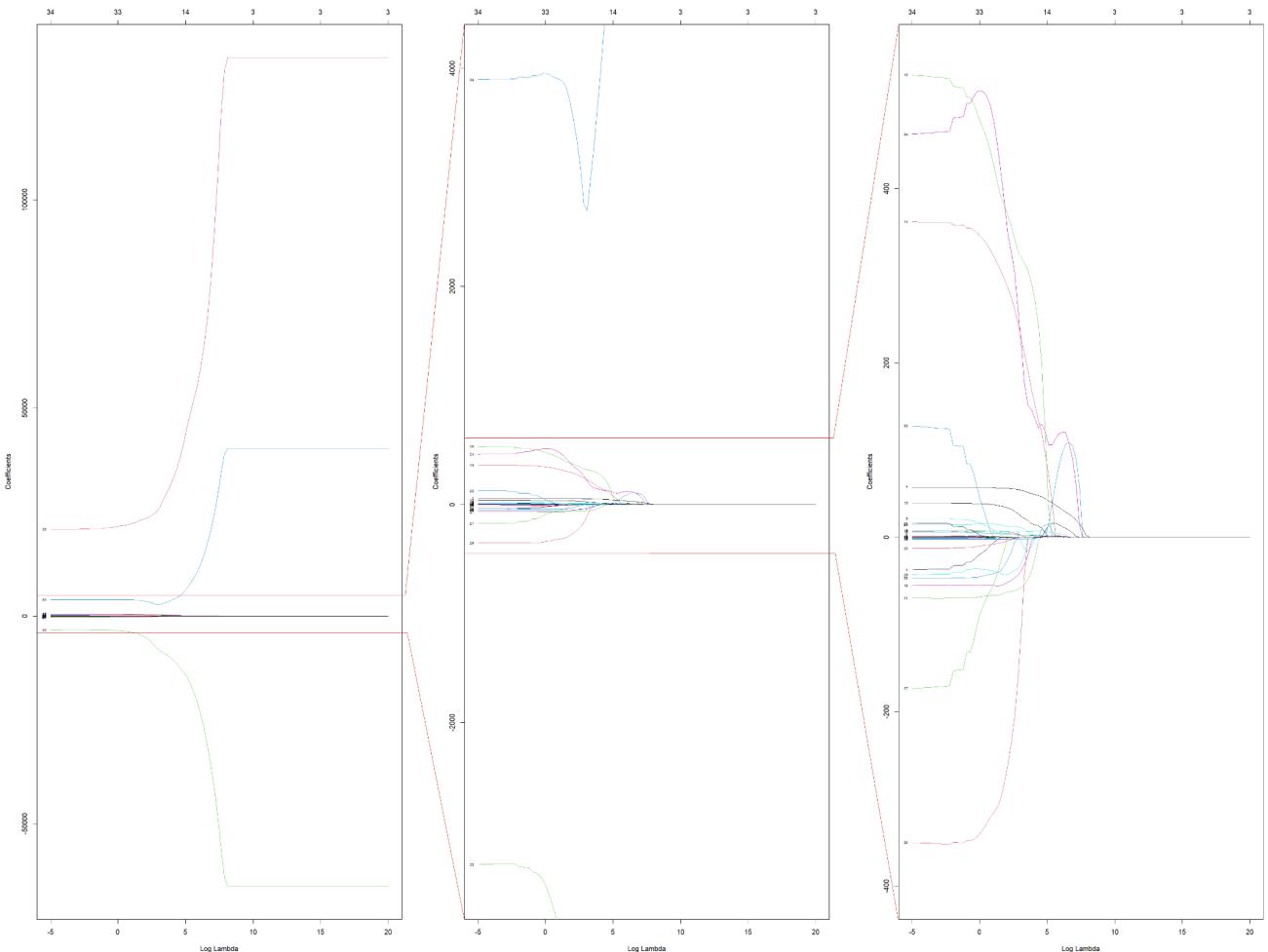


Figura 25: Percorso dei Coefficienti per la Regressione Elastic Net

La regressione Elastic Net combina le penalizzazioni  $L^1$  del LASSO e  $L^2$  della Ridge ( $\alpha=0.5$  per un mix bilanciato). Questa combinazione permette di beneficiare sia dello shrinkage dei

coefficienti (come Ridge) sia della selezione delle variabili (come Lasso), ed è particolarmente utile in presenza di gruppi di variabili correlate.

La regressione Elastic Net è stata implementata con `alpha = 0.5` in `glmnet()`, mantenendo lo stesso `penalty.factor` per le variabili fattoriali. L'analisi del percorso dei coefficienti (Figura 25) mostra un comportamento intermedio: i coefficienti vengono ridotti verso lo zero, ma alcuni di essi possono essere azzerati esattamente pur mantenendo una certa tendenza all'azione di gruppo tipica della Ridge. Grazie a questo comportamento ibrido Elastic Net riesce a effettuare una selezione delle variabili, pur mantenendo alcune delle relazioni di raggruppamento tra coefficienti correlati.

Anche per il modello Elastic Net, al fine di valutare le prestazioni e per identificare il valore ottimale di lambda è stata eseguita una cross-validazione a 10 fold utilizzando (`cv.glmnet`). L'output della cross-validazione fornisce informazioni chiave:

	Lambda	Index	Measure	SE	Nonzero
min	0.0067	100	223267	3507	34
1se	3.7178	75	226303	3417	30

- **lambda.min = 0.0067**: Valore di lambda che minimizza l'MSE (troppo piccolo per fare selezione).
- **lambda.1se = 3.7178**: Valore di lambda più regolarizzato entro una deviazione standard dal minimo. A questo `lambda.1se`, il numero di coefficienti non nulli (intercetta esclusa) è 30, il che indica un'efficace selezione delle variabili.

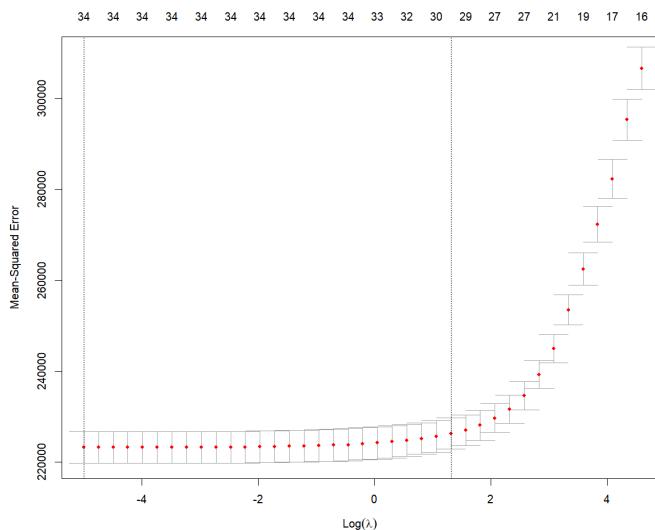


Figura 26: MSE di Cross-Validazione per la Regressione Elastic Net

Il grafico di cross-validazione (Figura 26), analogo ai precedenti, mostra l'andamento dell'MSE medio di cross-validazione in funzione di `log(lambda)`.

Anche per il modello Elastic Net è stata fatta validazione addestrando sul set di training e calcolando l'errore sul set di validazione usando `lambda.min`. L'MSE sul set di validazione è 225974, vicino al valore di MSE della cross-validazione (223267).

Analizzando i coefficienti del modello Elastic Net in corrispondenza di  $\lambda=\text{lambda.1se}$ , abbiamo riscontrato 30 variabili non trascurate (incluse quelle non penalizzate), con l'azzeramento dei coefficienti di 4 variabili: `daysSinceRegistration`, `log_challengeCardsWon`, `log_totalExpPoints`, e `sq_log_battleCount`. Rispetto al Lasso, l'Elastic Net ha azzerato una variabile in meno: `CardsLevel13`. Questo dimostra la capacità di Elastic Net di gestire la selezione delle variabili pur mitigando l'effetto di azzeramento eccessivo che talvolta si verifica con il solo LASSO in presenza di forte correlazione tra i predittori.

## 6.5 Confronto tra i Modelli Penalizzati

Per un confronto diretto, esaminiamo i coefficienti e le prestazioni dei modelli Ridge, Lasso ed Elastic Net, insieme al modello OLS completo. Tutti i coefficienti per i modelli penalizzati sono stati estratti utilizzando il `lambda.1se` dalla rispettiva cross-validation, al fine di favorire la parsimonia e la robustezza.

La tabella seguente riassume i coefficienti stimati per ciascun modello evidenziando le differenze dovute ai diversi tipi di penalizzazione.

Variable	OLS	Ridge	Lasso	Elastic_net
(Intercept)	-803,366	-362,628564	-118,701357	-149,5011625
expLevel	-52,827	-7,165929858	-0,1018691926	-2,357437369
challengeMaxWins	6,193	7,39083611	7,173667876	7,181782922
donationsReceived	-0,712	-0,7521628263	-0,6628234266	-0,6748163203
warDayWins	-1,785	-1,964813977	-1,906066659	-1,916151342
meanCostDeck	22,673	5,630682658	4,728798867	5,422999397
daysSinceRegistration	0,0093	0,00816102706	0	0
cardsOwned	57,292	52,49808818	56,9511201	56,61725388
CardsLevel13	2,792	0,08514290951	0	-0,01721366415
CardsLevel12	8,243	7,045604466	5,362599281	5,641466589
CardsLevel11	5,934	4,758581689	4,270510981	4,359706149
CardsLevel10	7,785	8,661271075	6,918962325	7,118571499
lastLeagueTrophies	-0,099	-0,1103435585	-0,07398065547	-0,07888221409
bestLeagueTrophies	-0,216	-0,2200498898	-0,1674885692	-0,1791780158
log_CardsLevel14	371,218	312,8487116	311,932183	313,6427263
log_CardsLevel15	566,947	405,561193	398,5977248	406,7164574
log_challengeCardsWon	0,658	0,01148558356	0	0
log_clanCardsCollected	-1,063	-2,937368953	-0,6586897173	-0,9295258158
log_totalDonations	-54,570	-54,65779386	-57,06477856	-55,646586
log_donations	39,472	39,98044903	36,27844556	36,79876799
log_threeCrownWins	-347,625	-281,6614811	-313,3803156	-308,4909411
log_starPoints	-70,083	-68,28571866	-66,64118164	-66,25223282
log_totalExpPoints	209,095	72,17973502	0	0
log_tournamentBattleCount	13,347	19,9792977	13,51843955	14,82137616
log_wins	380,763	387,7588354	434,2973314	448,2848044
sq_log_battleCount	28,198	11,56951705	0	0
fourth_root_expPoints	-14,703	-8,118350129	-7,001203578	-7,073418502
fourth_root_losses	-234,220	-112,3590991	-33,59527674	-40,84691887

sqrt_CardsEvo	-46,137	-37,40856446	-43,65438544	-40,25094205
sqrt_yearsSinceRegistration	-42,065	-47,24819269	-41,5477078	-40,22709003
sq_meanLevelCards	-2,208	-3,980319283	-2,491931588	-2,528957493
sq_meanLevelSupportCards	0,863	1,145981373	0,4655258527	0,674036962
poly(bestLeagueNumber, 3)1	20486,7	22532,3261	22541,3982	22530,44227
poly(bestLeagueNumber, 3)2	-3362,6	-3335,426777	-4237,058337	-4102,795477
poly(bestLeagueNumber, 3)3	3745,7	4370,961671	3768,259295	3870,706835

La tabella evidenzia le differenze tra i coefficienti stimati dai vari modelli. Si può notare come i coefficienti OLS siano generalmente più grandi in valore assoluto. Ridge li riduce tutti, mentre Lasso ed Elastic Net azzerano specifici coefficienti, realizzando la selezione delle variabili. In particolare, le variabili azzerate dal Lasso e dall'Elastic Net sono quelle che il modello ritiene meno influenti per la predizione del numero di trofei, tenendo conto della penalizzazione.

I termini polinomiali di `bestLeagueNumber` rimangono sempre presenti e con coefficienti significativi in tutti i modelli penalizzati grazie all'impostazione di `penalty.factor = 0`.

Il confronto più diretto delle prestazioni predittive avviene attraverso l'MSE stimato dalla cross-validation e l'MSE calcolato sul set di validazione.

Modello	MSE (CV $\lambda = \text{lambda\_min}$ )	MSE (Validation Set)	Coefficienti non nulli ( $\lambda = \text{lambda\_1se}$ )
Ridge	223119	225852.7	34
Lasso	223092	226091.3	29
Elastic Net	223267	225973.7	30

Le differenze negli MSE sui set di validazione sono minime tra i tre modelli penalizzati, il che suggerisce che tutti e tre offrono prestazioni predittive comparabili per questo dataset.

Per un ultimo confronto più robusto tra i tre tipi di regolarizzazione, è stata eseguita una cross-validation con lo stesso set di `foldid` per ciascun modello (Ridge, Lasso ed Elastic Net) e con una nuova griglia di lambda, al fine di garantire una comparabilità diretta.

I risultati degli MSE (corrispondenti a `lambda.min` per ciascun alpha) sono:

- **LASSO ( $\alpha=1$ ):** MSE = 224561.8
- **Elastic Net ( $\alpha=0.5$ ):** MSE = 223834.5
- **Ridge ( $\alpha=0$ ):** MSE = 222823.5

Questo confronto diretto, basato su un partizionamento identico dei dati per la cross-validation, indica che il Ridge ha ottenuto MSE minimo tra le diverse configurazioni di alpha testate.

Il grafico di confronto in Figura 27 visualizza le curve di cross-validation per i tre valori di alpha.

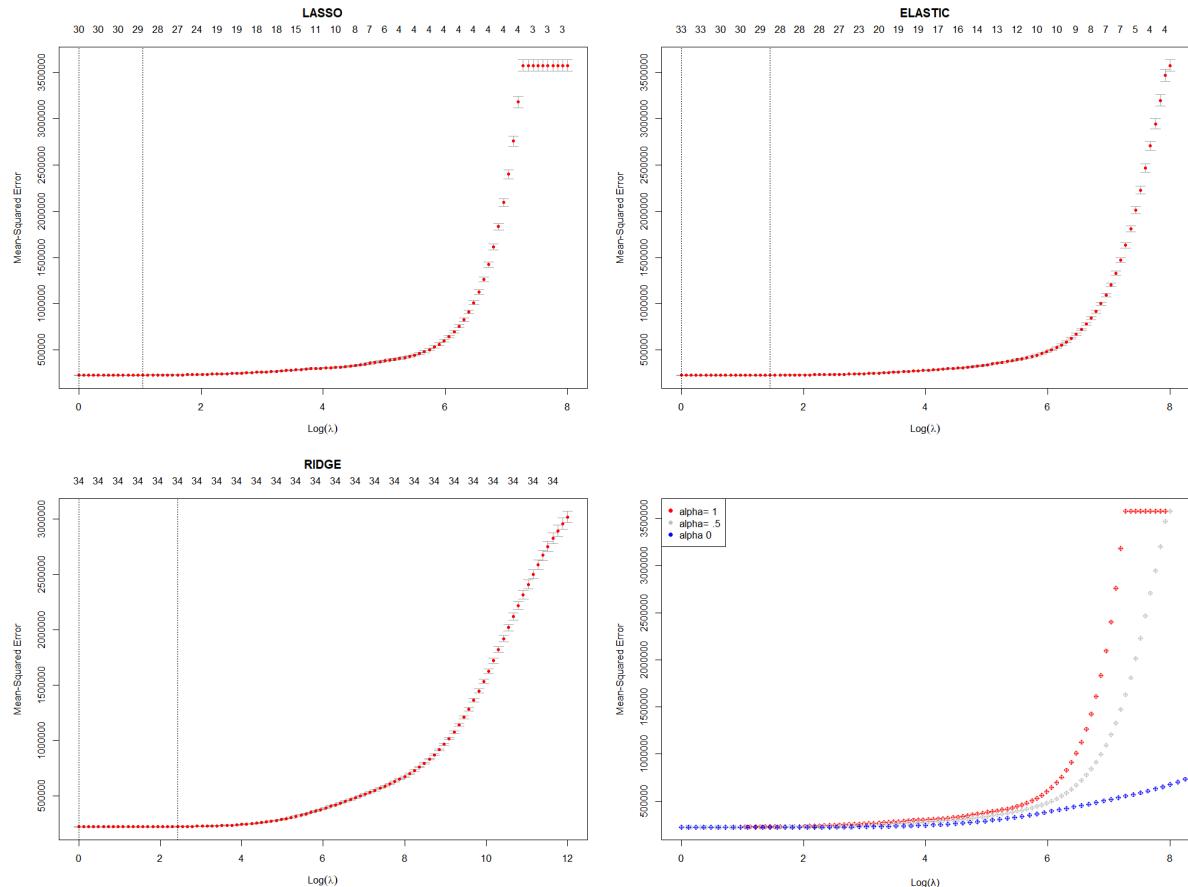


Figura 27: Confronto delle Curve di Cross-Validazione per Ridge, Lasso ed Elastic Net

L'ultimo grafico aggrega le curve MSE per i diversi alpha. Come ci si aspettava anche dai risultati di validazione la cross-validation ha rivelato che il Ridge offre il miglior comportamento predittivo. Invece Elastic Net offre il miglior compromesso tra MSE e parsimonia in questo specifico confronto. La Lasso (`alpha = 1`, linea rossa) mostra un MSE più elevato, soprattutto per valori di  $\log(\lambda)$  più grandi (minore penalizzazione).

## 6.6 Conclusioni sui metodi penalizzati

L'analisi dei modelli di regressione regolarizzata ha dimostrato la loro efficacia nel gestire i problemi di multicollinearità e overfitting, tipici dei modelli lineari con un elevato numero di predittori.

Tra i tre modelli regolarizzati, il **Ridge è emerso come il più performante** in termini di MSE di cross-validation, per quanto comunque realizzi un modello pieno.

Dal punto di vista della selezione delle variabili è risultato significativo il Lasso che ha offerto una precisione predittiva eccellente (simile alla Ridge e all'Elastic Net), ma ha anche svolto selezione automatica delle variabili, identificando un sottoinsieme di predittori più influenti. Le 29 variabili mantenute dal modello LASSO a  $\lambda_{1se}$  rappresentano un insieme più snello e

focalizzato per la previsione dei trofei, facilitando l'interpretabilità e la comprensione delle feature principali.

L'Elastic Net ha offerto un equilibrio tra i due, fornendo anch'esso una selezione di variabili e mantenendo la flessibilità di adattamento a diverse strutture di correlazione del Ridge.

In sintesi, l'approccio con i modelli regolarizzati, e in particolare l'Elastic Net, ha permesso di costruire un modello robusto e interpretabile per la previsione dei trofei dei giocatori, gestendo efficacemente la complessità del dataset e identificando i predittori più rilevanti.

# Capitolo 7: Selezione del Modello Finale e Applicazione

Questo capitolo conclusivo ha l'obiettivo di sintetizzare i risultati ottenuti nei capitoli precedenti per identificare il modello di regressione ottimale per la previsione dei trofei dei giocatori. Infine tale modello è applicato a dati non visti per quantificare in modo rigoroso le sue capacità predittive.

## 7.1 Confronto e Valutazione dei Modelli Esplorati

La selezione del "miglior" modello non si basa unicamente sulla minimizzazione dell'errore di previsione, ma considera un equilibrio tra accuratezza predittiva, parsimonia, interpretabilità e robustezza. Abbiamo esplorato diverse famiglie di modelli, ciascuna con i propri vantaggi e svantaggi, che verranno qui riassunti e confrontati.

### 7.1.1 Modello OLS con Variabile Risposta Trasformata (Capitolo 4)

Nel Capitolo 4 è stato sviluppato un modello di regressione lineare ordinaria (OLS) utilizzando `log(trophies+1)` come variabile risposta. Questo modello è stato costruito su un dataset sottoposto a un'accurata fase di preparazione e preprocessing:

- **Gestione dei Valori Mancanti (NA) e dei Valori Anomali (< 0):** rimozione dei valori assenti gestiti per righe o per colonne.
- **Trasformazione delle Variabili Predittive:** Un'ampia fase di analisi visiva e trasformazioni matematiche è stata applicata per linearizzare le relazioni con `trophies`. Conversione in fattoriali di 7 variabili. Rimozione di 9 variabili.
- **Gestione della Multicollinearità (Analisi VIF):** È stata condotta un'analisi iterativa del Variance Inflation Factor (VIF), rimuovendo variabili con valori superiori a 5.

**Analisi delle Assunzioni del Modello OLS (`log(trophies+1)`):** La trasformazione `log(trophies+1)` della variabile risposta ha portato a un apprezzabile miglioramento nella conformità alle assunzioni del modello lineare rispetto a un modello con `trophies` non trasformati.

#### Punti di Forza:

- La trasformazione logaritmica ha linearizzato efficacemente le relazioni e stabilizzato la varianza, rendendo il modello più aderente alle assunzioni OLS.
- Coefficienti chiari e interpretabili (sebbene sulla scala logaritmica).
- **Alti valori di R<sup>2</sup> e R<sup>2</sup> Aggiustato:** Il modello ha mostrato un R<sup>2</sup> di `0.9146` e un R<sup>2</sup> aggiustato di `0.9119`, indicando un'ottima capacità esplicativa.
- **Significatività Complessiva:** L'F-statistic pari a `344.4` con p-value `<2.2e-16` ha confermato la significatività globale del modello.
- **Cross-Validation:** Una 10-fold Cross-Validation ha prodotto un RMSE medio di `0.269` (sulla scala logaritmica) e un R<sup>2</sup> medio di `0.9049`, indicando una buona capacità di generalizzazione e un overfitting non eccessivo.

### Punti di Debolezza:

- La selezione delle variabili è stata un processo manuale e iterativo che pur riducendo la multicollinearità, ha sollevato un importante compromesso con le capacità predittive.
- Trade-off con `model_lm_full`: Confrontando `model_lm_log_trophies` (302 predittori, AIC=1903.77, BIC=4096.32, Cp=14223.6) con un `model_lm_full`, i criteri di informazione hanno sorprendentemente indicato che il modello completo fosse superiore. Un Cp così elevato suggerisce che la rimozione di variabili (anche se altamente collineari) potrebbe aver introdotto un bias significativo, sacrificando la capacità predittiva. Questo implica che il `model_lm_full`, pur essendo più complesso, potrebbe aver catturato meglio la varianza totale.
- L'errore è sulla scala logaritmica, rendendo il confronto diretto con gli MSE dei modelli del Capitolo 6 (su scala originale dei trofei) meno immediato. La ri-trasformazione delle previsioni per ottenere l'MSE sulla scala originale è necessaria per un confronto equo. Dopo aver addestrato il modello sul training set possiamo ri-trasformare i valori fittati in scala originale e calcolare l'MSE sui dati di training in scala originale. Così facendo il metodo diventa confrontabile ma per nulla performante dato che l'errore di training risulta 807877.8.

### 7.1.2 Modelli con Penalizzazione L0 (Capitolo 5)

Nel Capitolo 5, abbiamo esaminato due approcci basati sulla penalizzazione L<sup>0</sup> che mirano a identificare il sottoinsieme ottimale di predittori minimizzando il numero di coefficienti non nulli: la selezione del miglior sottoinsieme (`regsubsets`) e la selezione iterativa a passi (`step()`). Questi modelli lavorano sulla variabile risposta `trophies` su scala originale.

- **Punti di Forza:**
  - `regsubsets`: Offre i migliori sottoinsiemi per ogni numero di variabili, permettendo di bilanciare accuratezza e parsimonia in base a criteri come BIC (più parsimonioso), Cp o R<sup>2</sup> aggiustato.
  - `step()`: Algoritmo efficiente per la selezione del modello basata sull'AIC.
  - Entrambi producono modelli intrinsecamente parsimoniosi con coefficienti esattamente a zero per le variabili non selezionate, migliorando l'interpretabilità.
- **Punti di Debolezza:**
  - `regsubsets`: Computazionalmente intenso.
  - `step()`: Non garantisce l'ottimo globale (essendo un algoritmo greedy).
  - Nessuno dei due affronta direttamente la multicollinearità tramite la contrazione dei coefficienti, a differenza dei metodi penalizzati L<sup>1</sup>/L<sup>2</sup>.
- **Metriche Chiave:**
  - `regsubsets`: Ha identificato modelli ottimali con **25 variabili** (minimo BIC), **31 variabili** (minimo Cp), e **32 variabili** (massimo R<sup>2</sup> aggiustato).
  - `step()`: I modelli finali hanno incluso **36 variabili** (backward e stepwise) e **42 variabili** (forward), con R<sup>2</sup> aggiustato attorno a 0.963 e valori di AIC tra **151723.7** e **151731.5**.

### 7.1.3 Modelli con Penalizzazione L<sup>1</sup>/L<sup>2</sup>/Elastic Net (Capitolo 6)

Nel Capitolo 6, abbiamo applicato tecniche di regressione penalizzata (Ridge, Lasso ed Elastic Net) utilizzando la variabile risposta `trophies` su scala originale. Come discusso, le variabili dummy `Hog Rider`, `Elixir Golem` e `Mega Knight` sono state rimosse dal dataset per un modello più pulito, portando la matrice di disegno X a includere 34 predittori. I termini polinomiali di `bestLeagueNumber` sono stati ridotti a tre e sono stati esclusi dalla penalizzazione (`penalty.factor = 0`).

- **Punti di Forza:**

- **Gestione della Multicollinearità:** Tutte e tre le tecniche regolarizzate sono più o meno efficaci nel gestire predittori altamente correlati attraverso la contrazione dei coefficienti.
- **Prevenzione dell'Overfitting:** La penalizzazione riduce la complessità e migliora la generalizzabilità.
- **Selezione Automatica delle Variabili (Lasso ed Elastic Net):** Lasso ed Elastic Net azzerano i coefficienti, realizzando una selezione automatica delle variabili e producendo modelli parsimoniosi.
- **Robustezza:** La cross-validation intrinseca di `glmnet` permette una selezione robusta del parametro  $\lambda$ .

- **Punti di Debolezza:** L'interpretazione diretta dei singoli coefficienti può essere meno intuitiva rispetto all'OLS a causa dello shrinkage.

- **Metriche Chiave:**

- **Ridge ( $\alpha=0$ ):** Ha mantenuto tutti i **34 predittori**.
  - **MSE (CV  $\lambda_{\min}$ ):** `222823.5` (Questo è l'MSE più basso tra tutti i modelli regolarizzati testati).
- **Elastic Net ( $\alpha=0.5$ ):** Ha selezionato **30 predittori** (a  $\lambda_{1se}$ ).
  - **MSE (CV  $\lambda_{\min}$ ):** `223834.5`
- **Lasso ( $\alpha=1$ ):** Ha selezionato **29 predittori** (a  $\lambda_{1se}$ , il modello più parsimonioso).
  - **MSE (CV  $\lambda_{\min}$ ):** `224561.8`

In questo confronto finale dei modelli regolarizzati (testati con la stessa griglia di CV su  $\alpha$ ), il **modello Ridge ha mostrato l'MSE di cross-validation più basso**, indicando la migliore accuratezza predittiva tra i modelli regolarizzati. Elastic Net e Lasso hanno MSE molto simili al Ridge, ma con il vantaggio di produrre modelli più parsimoniosi.

## 7.2 Scelta del Modello Finale: Elastic Net

Alla luce dell'analisi approfondita dei diversi approcci di modellazione presentati nei Capitoli 4, 5 e 6, si è giunti alla decisione di selezionare il modello **Elastic Net** come modello finale per la previsione dei trofei dei giocatori. Questa scelta è basata su un'attenta valutazione dei compromessi tra accuratezza predittiva, parsimonia del modello e interpretabilità, oltre alla sua robustezza e capacità di gestione della multicollinearità.

Come emerso dal confronto finale del Capitolo 6, il modello Ridge ha mostrato l'Errore Quadratico Medio (MSE) di cross-validation più basso. Tuttavia, il Ridge mantiene tutti i 34

predittori attivi, non eseguendo alcuna selezione delle variabili. Il Lasso, pur essendo il più parsimonioso (29 predittori non nulli), ha mostrato un MSE leggermente superiore.

L'Elastic Net, con un MSE di cross-validation di `223834.5`, si posiziona molto vicino al Ridge in termini di accuratezza predittiva, ma offre un vantaggio significativo nella parsimonia, selezionando 30 predittori non nulli a  $\lambda 1se$ . Questo lo rende un eccellente compromesso tra la minimizzazione dell'errore (performance predittiva) e la semplicità del modello (interpretabilità).

Un ulteriore elemento chiave a supporto di questa scelta è la notevole somiglianza tra le variabili che l'Elastic Net azzerà e quelle che sono state escluse dal modello `regsubsets` quando si mirava a minimizzare il criterio Cp. Le variabili escluse dall'Elastic Net (`daysSinceRegistration`, `log_challengeCardsWon`, `log_totalExpPoints`, `sq_log_battleCount`) riflettono una selezione coerente con la ricerca di un modello parsimonioso e ben bilanciato, indicando che l'Elastic Net non solo gestisce efficacemente le penalità  $L^1$  ed  $L^2$ , ma identifica anche un sottoinsieme di variabili fondamentali che altri metodi parsimoniosi tenderebbero a favorire.

Pertanto, l'Elastic Net è stato scelto per la sua capacità di offrire un modello robusto, predittivamente competitivo e al contempo sufficientemente parsimonioso da facilitare l'interpretazione, rappresentando il "vincitore" ideale per gli obiettivi di questo studio.

### 7.3 Valutazione del Modello Finale sul Set di Test Indipendente

Avendo scelto il modello Elastic Net come il più equilibrato tra accuratezza e parsimonia, è cruciale valutarne la performance su un set di dati completamente indipendente e non visto durante alcuna fase di addestramento o selezione del modello. Questo set di test, rappresentante il 20% delle osservazioni originali, fornisce una stima imparziale della capacità di generalizzazione del modello.

Il modello Elastic Net, adattato con il parametro di regolarizzazione `lambda.1se` (scelto per il suo equilibrio tra performance e parsimonia), è stato utilizzato per generare previsioni sul set di test. Le metriche di errore calcolate sono le seguenti:

- **Errore Quadratico Medio (MSE):** `277649.5`,
- **Radice dell'Errore Quadratico Medio (RMSE):** `526.9246`,
- **Errore Assoluto Medio (MAE):** `391.1744`.

Questi valori forniscono un'indicazione chiara dell'accuratezza predittiva del modello sulla scala originale dei trofei. L'RMSE di circa 527 trofei significa che, in media, le previsioni del modello si discostano dai valori reali per circa 527 trofei, mentre l'MAE di circa 391 trofei indica l'errore assoluto medio. Confrontando questo MSE sul set di test (`277649.5`) con l'MSE di cross-validation stimato nel Capitolo 6 (`223834.5`), si osserva un aumento dell'errore sul set di test. Questa è una situazione attesa, poiché l'errore di cross-validation fornisce una stima interna della performance, mentre il test set è una valutazione esterna su dati completamente nuovi, spesso più conservativa. Tuttavia, l'ordine di grandezza rimane coerente, indicando una buona capacità di generalizzazione del modello Elastic Net.

L'analisi degli errori (la differenza tra i trofei previsti e quelli reali) sul set di test rivela la distribuzione seguente:

- **Minimo:** **-1646.83** (il modello ha sottostimato di oltre 1600 trofei in alcuni casi),
- **1° Quartile:** **-324.60**,
- **Mediana:** **-76.94**,
- **Media:** **-6.64** (la media degli errori è molto vicina a zero, suggerendo che il modello non ha un bias sistematico di sovrastima o sottostima complessiva),
- **3° Quartile:** **259.39**,
- **Massimo:** **3317.73** (il modello ha sovrastimato di oltre 3300 trofei in alcuni casi).

Questa distribuzione mostra che, sebbene la media sia vicina a zero, esistono casi di errori significativi sia in sottostima che in sovrastima, in particolare nelle code della distribuzione. Ciò è comune nei modelli che prevedono variabili con un'ampia gamma di valori come i trofei dei giocatori.

## 7.4 Conclusioni Finali

Questo studio ha esplorato in profondità diverse metodologie di regressione per prevedere il numero di trofei dei giocatori di Clash Royale, partendo da un'estesa raccolta e pulizia dei dati fino alla costruzione e valutazione di modelli complessi.

Il percorso analitico ha evidenziato come le fasi di pre-processing (gestione NA, trasformazioni delle variabili predittive e della variabile risposta, analisi VIF e gestione della multicollinearità) siano state cruciali per preparare un dataset robusto e ottimale per la modellazione.

Il confronto tra i modelli **1m** (con variabile risposta trasformata), i metodi di selezione  $L^0$  (Best Subset Selection, Stepwise) e i modelli di regressione regolarizzata ( $L^1$ /Lasso,  $L^2$ /Ridge, Elastic Net) ha permesso di comprendere i loro rispettivi punti di forza e debolezza.

La scelta finale è ricaduta sul modello **Elastic Net**. Questa decisione è stata guidata dalla sua capacità di raggiungere un eccellente equilibrio tra accuratezza predittiva e parsimonia, grazie alla sua intrinseca capacità di selezione delle variabili. La coerenza tra le variabili selezionate da Elastic Net e quelle identificate come ottimali dai criteri di informazione (come il Cp) in altri modelli ha ulteriormente rafforzato questa scelta.

Il modello Elastic Net finale ha dimostrato una buona capacità di generalizzazione sul set di test indipendente, con un RMSE di circa 527 trofei.

In conclusione, questo studio fornisce una metodologia robusta per la previsione delle prestazioni dei giocatori in Clash Royale, evidenziando l'importanza di un preprocessing accurato e di una selezione informata tra diverse tecniche di modellazione. Il modello Elastic Net si configura come uno strumento prezioso per comprendere i fattori che influenzano il successo dei giocatori e per generare previsioni affidabili.