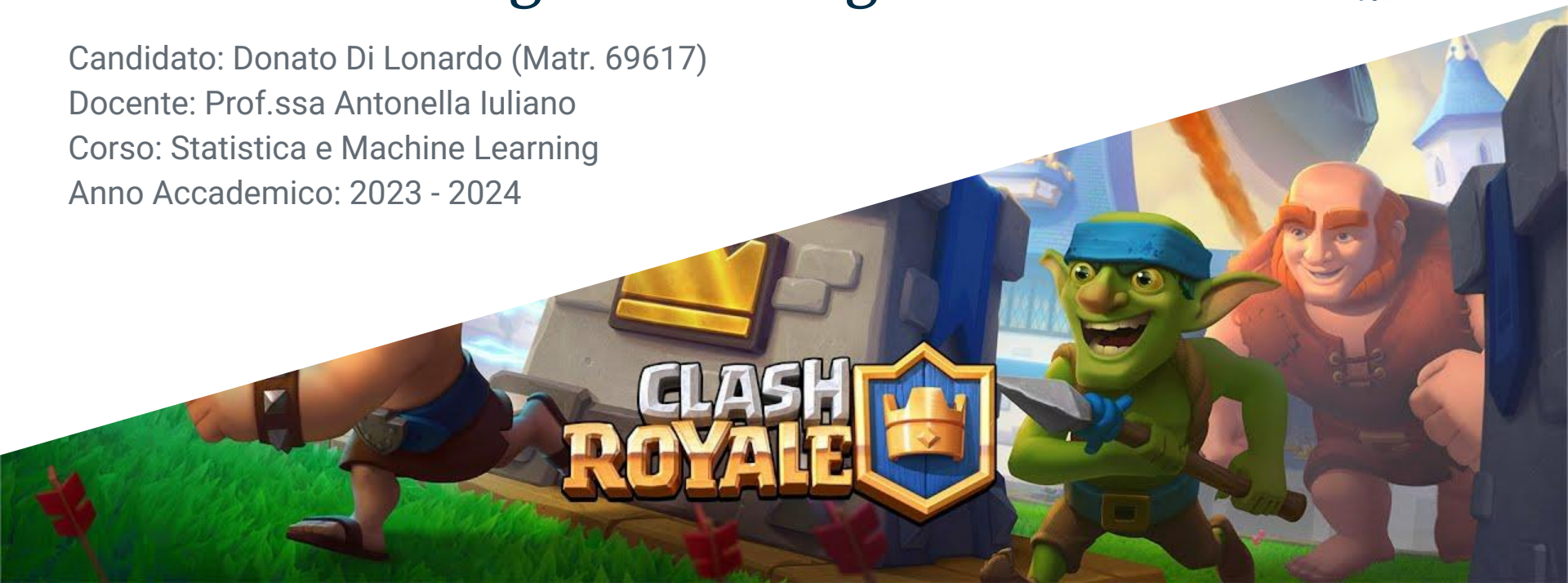
Analisi Predittiva delle Performance dei Giocatori in Clash Royale: Un'applicazione dei Modelli di Regressione Regolarizzata.

Candidato: Donato Di Lonardo (Matr. 69617)
Docente: Prof.ssa Antonella Iuliano
Corso: Statistica e Machine Learning
Anno Accademico: 2023 - 2024

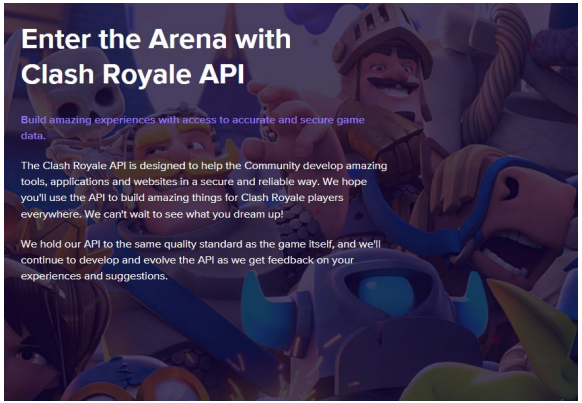


Introduzione e Obiettivo del Progetto

- **Clash Royale**
  - Gioco di strategia mobile con milioni di giocatori.
  - Basato su duelli PvP con mazzi di 8 carte giocabili ed una di supporto.
  - La performance dei giocatori genera un'enorme quantità di dati (Big Data).
- **Obiettivo del Progetto**
  - **Prevedere la performance** di un giocatore, misurata dal suo numero di **trofei**.
  - Identificare le **variabili più influenti** sul successo di un giocatore.
  - Confrontare diverse famiglie di modelli di regressione per trovare il miglior compromesso tra **accuratezza predittiva**, **parsimonia** e **interpretabilità**.

Il Percorso dei Dati: Dall'API al Dataset

La casa produttrice del gioco (Supercell) mette a disposizione una **API** (Application Programming Interface) pubblica dedicata a Clash Royale. Interrogando l'API ho ottenuto i dati di **12812** giocatori distinti.



I dati forniti dall'API si presentano in una **struttura** gerarchica **complessa** di liste e sotto-liste che richiede una trasformazione significativa per una efficacemente esplorazione e manipolazione.

Gestione di **valori mancanti** (NA) o anomali attraverso strategie mirate a preservare la qualità del dataset senza compromettere troppo l'ammontare del dataset stesso.

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

Dataset piatto

Il dataset piatto ottenuto al termine di questo processo è composto da circa 12.525 osservazioni (giocatori) e 173 variabili. Di seguito, una descrizione dettagliata di alcune colonne:

Variabile Risposta (Target):

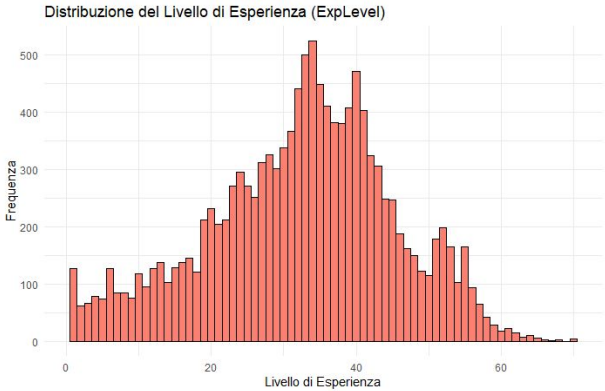
- `trophies` **(Numerica):** Il numero di trofei totalizzati dal giocatore in una modalità di gioco. Il nostro obiettivo è predire questo valore.

Predittori Chiave (Esempi):

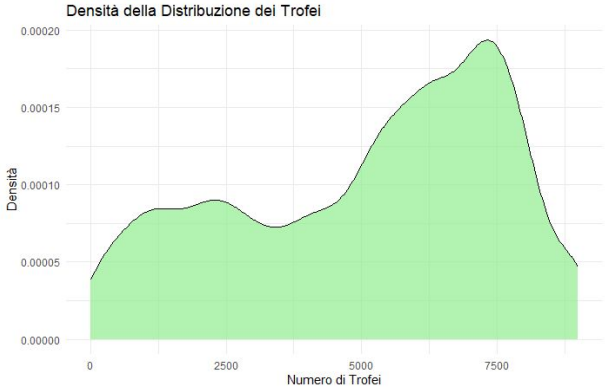
- `cardsOwned` **(Numerica):** Numero totale di carte sbloccate.
- `wins` **(Numerica):** Numero totale di vittorie che del giocatore.
- `totalExpPoints` **(Numerica):** Punti esperienza totali accumulati.
- `role` **(Fattore)**: Il ruolo del giocatore all'interno del proprio clan.
- `[NomeCarta]` **(Fattore binaria):** Presenza di una carta specifica nel mazzo, indicatore dello stile di gioco.

Per garantire una valutazione imparziale e robusta del modello di regressione il dataset pulito è stato suddiviso in due porzioni distinte:

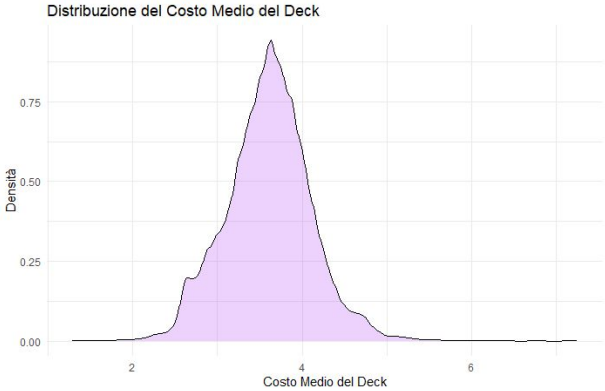
- **Training Set:** 80% delle osservazioni del dataset e utilizzato per l'addestramento e la scelta del modello.
- **Test Set:** 20% delle osservazioni, sarà “blindato” e verrà utilizzato solo alla fine del processo di modellazione per una valutazione finale e non distorta delle prestazioni del modello.



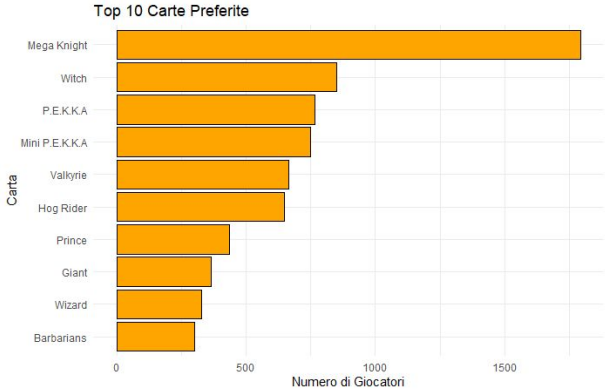
Distribuzione del Livello di Esperienza



Densità della Distribuzione dei Trofei dei Giocatori.



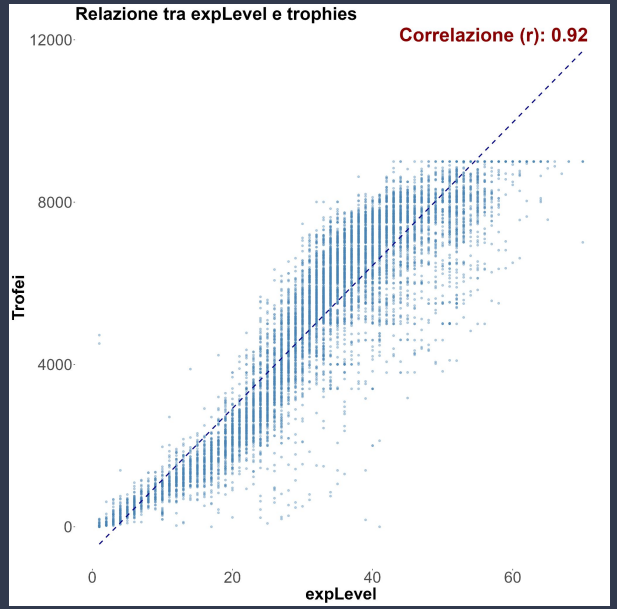
Densità della distribuzione del Costo Medio del Deck



Top 10 Carte Preferite dai Giocatori

Uno Sguardo ai Dati (Analisi Esplorativa)

Strategie predittive analizzate



Per identificare il miglior modello predittivo, sono stati esplorati e confrontati tre diverse metodologie di regressione:

1. **Regressione Lineare Classica (OLS)**
    - Obiettivo: Stabilire una base di partenza e verificare le assunzioni statistiche.
2. **Selezione del Miglior Sottoinsieme (Penalizzazione L^0)**
    - Obiettivo: Trovare il modello più parsimonioso identificando solo le variabili essenziali.
3. **Regressione Penalizzata (Ridge, Lasso, Elastic Net)**
    - Obiettivo: Gestire la complessità e la multicollinearità per migliorare la capacità predittiva.

Approccio 1 Regressione Lineare Classica (OLS)

Le fondamenta del nostro modello

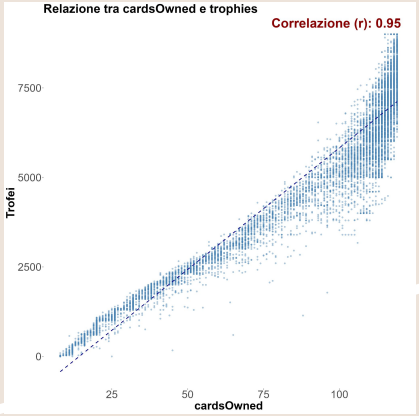
$$Y = X\beta + \varepsilon$$

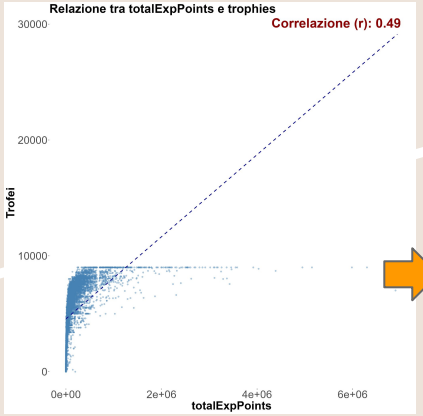
L'obiettivo della OLS è minimizzare la somma dei quadrati dei residui. È un modello potente e altamente interpretabile, ma si basa su assunzioni stringenti:

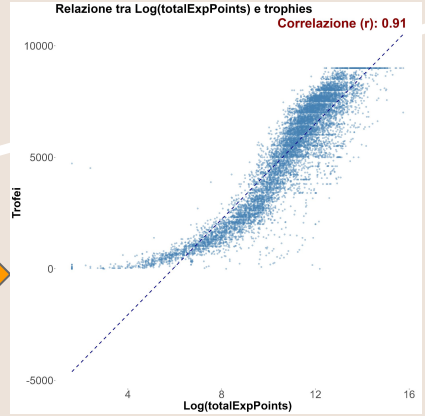
- **Linearità dei dati:** si assume che la relazione tra predittori (X) e risultato (Y) sia lineare.
- **Normalità dei residui:** si assume che gli errori residui siano distribuiti normalmente.
- **Omogeneità della varianza dei residui:** si assume che i residui abbiano una varianza costante (omoschedasticità).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

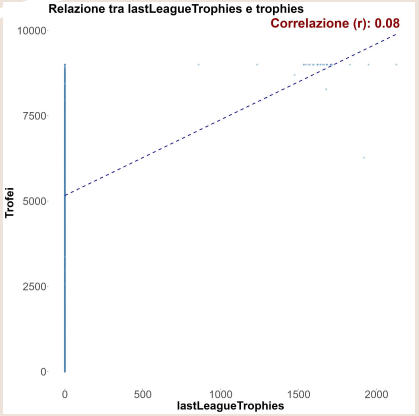
Verifica della Dipendenza Lineare e Ottimizzazione delle Variabili

- Studiare la relazione tra le variabili predittive (X) e la variabile target `trophies` (Y).
- Migliorare la linearità di queste relazioni.









Relazioni Già Lineari o Sufficientemente Lineari (9 variabili)

Relazioni Non Lineari che Necessitano di Trasformazioni matematiche (18 variabili)

Relazioni Insoddisfacenti che Portano alla Rimozione (9 variabili)

Gestione della Multicollinearità Attraverso l'Analisi VIF

La presenza di multicollinearità, ovvero correlazione tra le variabili predittive, 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)**. L'obiettivo è stato quello di rimuovere le variabili con valori superiori a 5, un indicatore riconosciuto di multicollinearità problematica.

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

Rimozione 8 variabili:

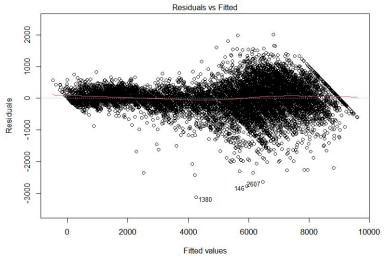
`meanCostDeck`, `battleCount`, `expLevel`, `yearsSinceRegistration`, `losses`, `threeCrownWins`, `meanLevelCards` e `sq_meanLevelSupportCards`

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

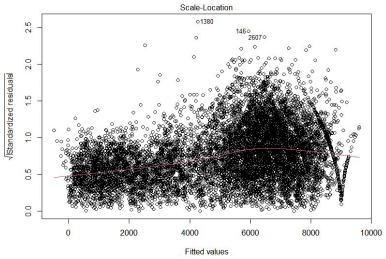
Verifica delle Assunzioni legate ai Residui

1. Linearità e Omoschedasticità (Analisi "Residui vs. Valori Fittati" e "Scala-Posizione").
2. Normalità dei Residui (Analisi Q-Q plot e istogramma dei residui).
3. Outlier e High-leverage points (Individuati con "Residui vs. Leverage").

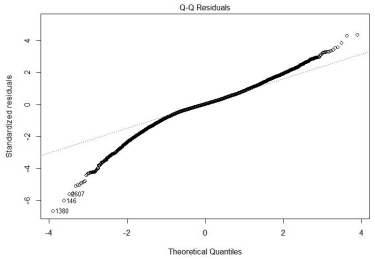
Leggere violazioni migliorate trasformando la variabile obiettivo con il logaritmo



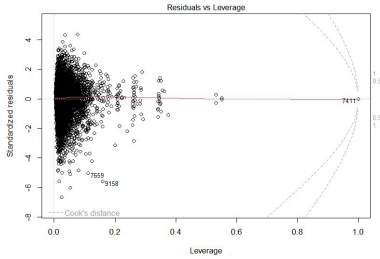
Residui vs. Valori Fittati



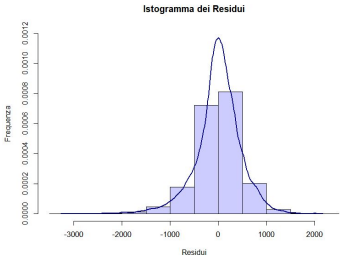
Scala-Posizione



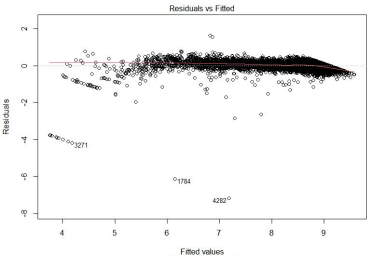
Q-Q plot dei residui

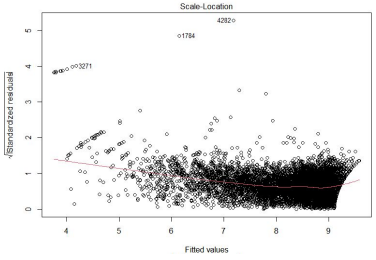


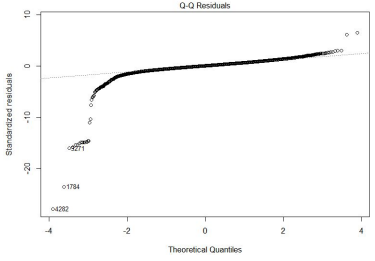
Residui vs. Leverage

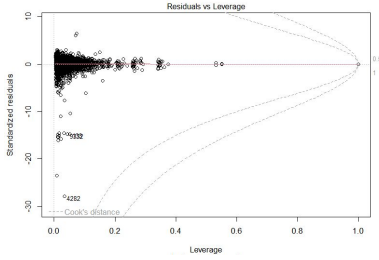


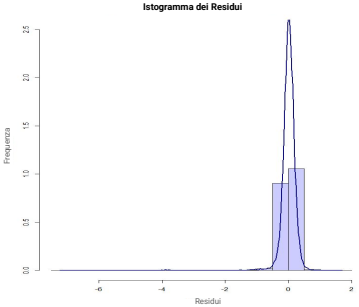
Istogramma dei residui











Valutazione del Modello

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

Variabilità spiegata dal modello e significatività generale.
Tabella ANOVA generale

R^2	0.9146		R^2 aggiustato	0.9119
Natura della variabilità	Somma dei quadrati	df	Media dei quadrati	F-statistic
Modello	SS_R = 7147.086	p = 302	SSRp = 23.666	344.428
Errore	SS_E = 667.660	n-p-1 = 9717	SSEn-p-1 = 0.0687	p-value
Totale	SS_T = 7814.746	n-1 = 10019		$<2.2\times10^{-16}$

Valore dei coefficienti e verifica della significatività attraverso t-test.

	Estimate	t value	Pr(>|t|)
log_totalExpPoints	2.491e-01	36,11	<2e-16
log_wins	2.707e-01	36,00	<2e-16
cardsOwned	1.009e-02	31,30	<2e-16
CardsLevel13	-6.195e-03	-8,23	<2e-16
log_CardsLevel15	-8.106e-02	-8,35	<2e-16
fourth_root_expPoints	-1.427e-02	-9,68	<2e-16
log_CardsLevel14	-5.333e-02	-9,95	<2e-16
sqrt_CardsEvo	-5.881e-02	-11,31	<2e-16
log_tournamentBattleCount	-2.951e-02	-11,42	<2e-16
CardsLevel10	-2.941e-03	-11,73	<2e-16
log_totalDonations	-2.812e-02	-13,26	<2e-16

10-fold Cross-Validation:

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

Errore di training in scala logaritmica:

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

MSE su dati di training in scala originale: 807877.8.

Approccio 2 Selezione L^0 del miglior sottoinsieme di predittori

L'obiettivo è trovare il sottoinsieme di predittori che massimizza la performance con il minor numero di variabili. Questo metodo, a differenza delle penalizzazioni, **elimina invece di restringere**.

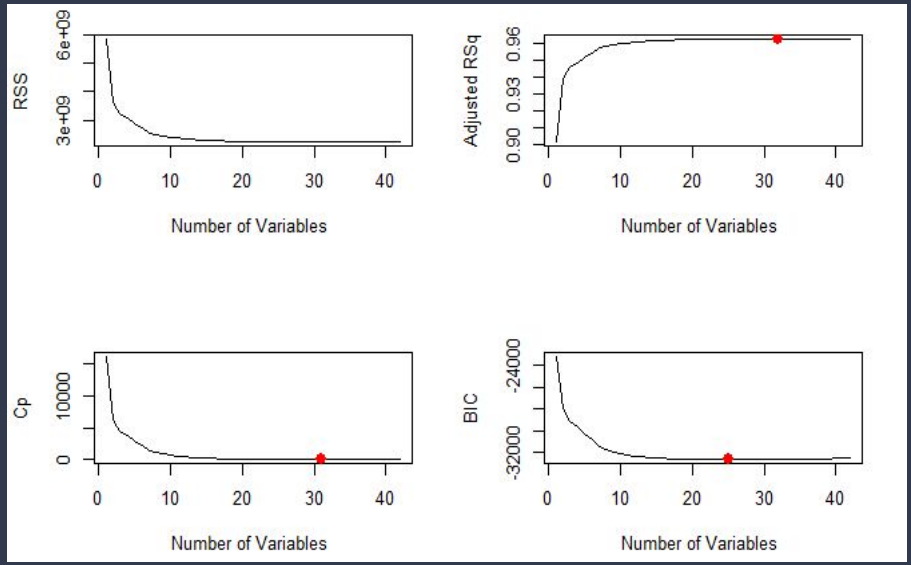
Metodi analizzati:

- **Best Subset Selection (`regsubsets`) :** Approccio esaustivo che garantisce di trovare il miglior modello per ogni dimensione.
- **Step Selection (`step`) :** Approccio euristico (greedy) più veloce.

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

- Variabile Risposta: **`trophies`**.
- Variabili Predittive (34 predittori totali gestiti nei modelli come 42):
  - 30 variabili numeriche.
  - 4 variabili fattoriali:
    - **`bestLeagueNumber`**: variabile ordinale a 10 livelli gestita attraverso 9 contrasti polinomiali ortogonali.
    - **`Hog Rider`**, **`Elixir Golem`**, **`Mega Knight`**: Tre variabili binarie (TRUE/FALSE).

Risultati della Best Subset Selection



Trovare il sottoinsieme di variabili più performante per una data dimensione del modello, basandosi su criteri come l'RSS (Residual Sum Squares), R^2 (Coefficiente di Determinazione), R^2 aggiustato, il Cp di Mallows e il BIC (Bayesian Information Criterion).

Eseguendo `regsubsets()` con la modalità `exhaustive` addestra tutti i 2^{42} modelli possibili e restituisce il migliore per ogni dimensione.

- **Coefficiente di Determinazione**: Il valore di R^2 aumenta monotonicamente 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.
- **Residual Sum Squares:** Coerentemente con l'R^2 diminuisce monotonicamente all'aumentare delle variabili.
- **Adjusted R^2:** Questo criterio penalizza l'aggiunta di predittori non significativi. L'R^2 aggiustato raggiunge un picco circa pari a **0.9626** e lo raggiunge con il miglior modello con **32** variabili (MSE=222061.3).
- **Cp di Mallows:** Il Cp di Mallows è un criterio che cerca di bilanciare il bias e la varianza del modello. Il valore minimo di Cp osservato è di circa **26.6348**, ottenuto con un modello che include **31** variabili (MSE=222102).
- **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 (MSE=222780.1).

	25	26	27	28	29	30	31	32
(Intercept)	*	*	*	*	*	*	*	*
challengeMaxWins	*	*	*	*	*	*	*	*
donationsReceived	*	*	*	*	*	*	*	*
warDayWins	*	*	*	*	*	*	*	*
meanCostDeck								*
daysSinceRegistration								
cardsOwned	*	*	*	*	*	*	*	*
CardsLevel13								
CardsLevel12	*	*	*	*	*	*	*	*
CardsLevel11	*	*	*	*	*	*	*	*
CardsLevel10	*	*	*	*	*	*	*	*
lastLeagueTrophies							*	*
bestLeagueTrophies					*	*	*	*
log_CardsLevel14	*	*	*	*	*	*	*	*
log_CardsLevel15	*	*	*	*	*	*	*	*
log_challengeCardsWon								
log_clanCardsCollected								
log_totalDonations	*	*	*	*	*	*	*	*
log_donations	*	*	*	*	*	*	*	*
log_threeCrownWins	*	*	*	*	*	*	*	*
log_starPoints	*	*	*	*	*	*	*	*
log_totalExpPoints					*	*	*	*

	25	26	27	28	29	30	31	32
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	*	*	*	*	*	*	*	*

Risultati del processo step

La selezione step **costruisce** il modello **passo dopo passo**, aggiungendo o rimuovendo predittori in base all'AIC (Akaike Information Criterion). Questo metodo "**greedy**" non garantisce di trovare il modello globalmente migliore, ma è **computazionalmente** più **efficiente**.

La funzione ha tre modalità di ricerca:

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

- **Selezione all'indietro e Selezione Bidirezionale:** portano allo stesso modello con 36 variabili predittive, escludono alcune variabili indicando che la loro rimozione ha portato a una riduzione dell'AIC. L'R^2 e R^2 aggiustato sono 0.9627 e 0.9626.
- **Selezione in avanti:** ha identificato il modello completo come il migliore in questa direzione di ricerca. L'R^2 e l'R^2 aggiustato sono 0.9627 e 0.9626.

Modello	gradi di libertà	numero di predittori	AIC
Selezione all'indietro	38	36	151724
Selezione in avanti	44	42	151732
Selezione Bidirezionale	38	36	151724

Approccio 3 Regressione Penalizzata

$$P_{\lambda,\alpha}(\beta) = \lambda \left[(1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right]$$

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 (**Filosofia "Shrinkage"**). Questi metodi sono implementati in R con la funzione **`glmnet()`**.

Metodi analizzati:

- **Regressione Ridge (α=0):** Aggiunge una penalità L^2. L'effetto è di "**restringere**" i coefficienti verso zero riducendone la varianza e gestendo la multicollinearità. I coefficienti non vengono mai azzerati completamente. Coefficienti correlati vanno a 0 insieme.
- **Regressione Lasso (α=1):** Aggiunge una penalità L^1 . Ha la caratteristica unica di **azzerare** completamente i coefficienti di predittori meno rilevanti (selezione automatica delle variabili).
- **Elastic Net (0<α<1):** Combina le penalità L^1 e L^2. Offre i vantaggi di entrambi i metodi: capacità di selezione del Lasso e la stabilità e l'effetto di raggruppamento (grouping effect) del Ridge. Per questa analisi, è stato utilizzato **α**=0.5.

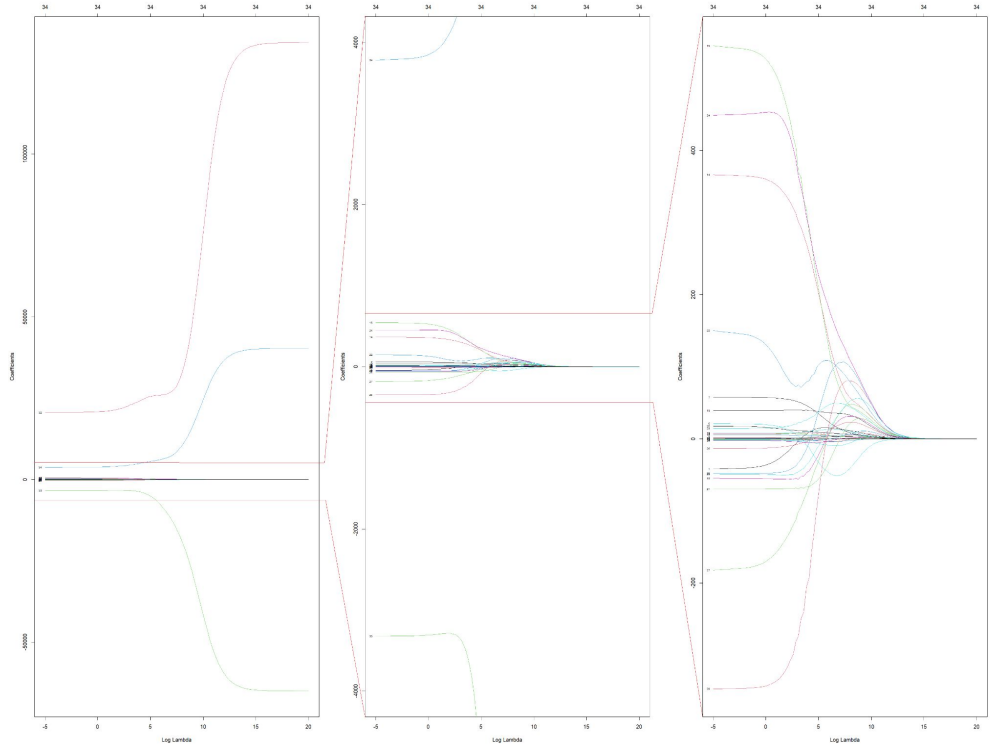
Il dataset è stato accuratamente preparato per ottimizzare la modellazione e include:

- Variabile Risposta: **`trophies`**.
- Variabili Predittive (32 predittori totali gestiti nei modelli come 34):
  - 31 variabili numeriche.
  - la variabile fattoriale **`bestLeagueNumber`**: variabile ordinata a 10 livelli gestita attraverso i soli primi 3 contrasti polinomiali (associati ad un **`penalty.factor`** nullo).

Oltre ad **`alpha`** ed alla **`penalty`** la funzione **`glmnet`** prende in input anche una griglia di valori di **λ**.

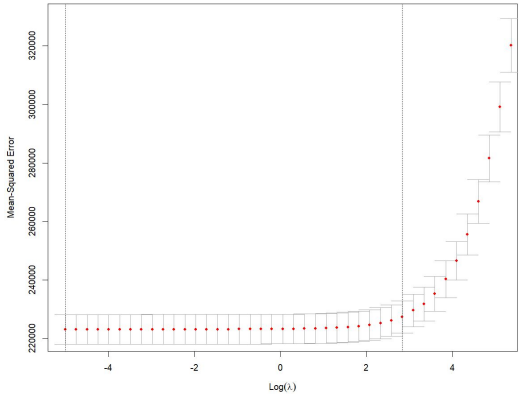
Risultati Ridge

Evoluzione dei coefficienti della Regressione Ridge al variare di λ



È stata eseguita cross-validazione a 10 fold utilizzando **`cv.glmnet`** e con output:

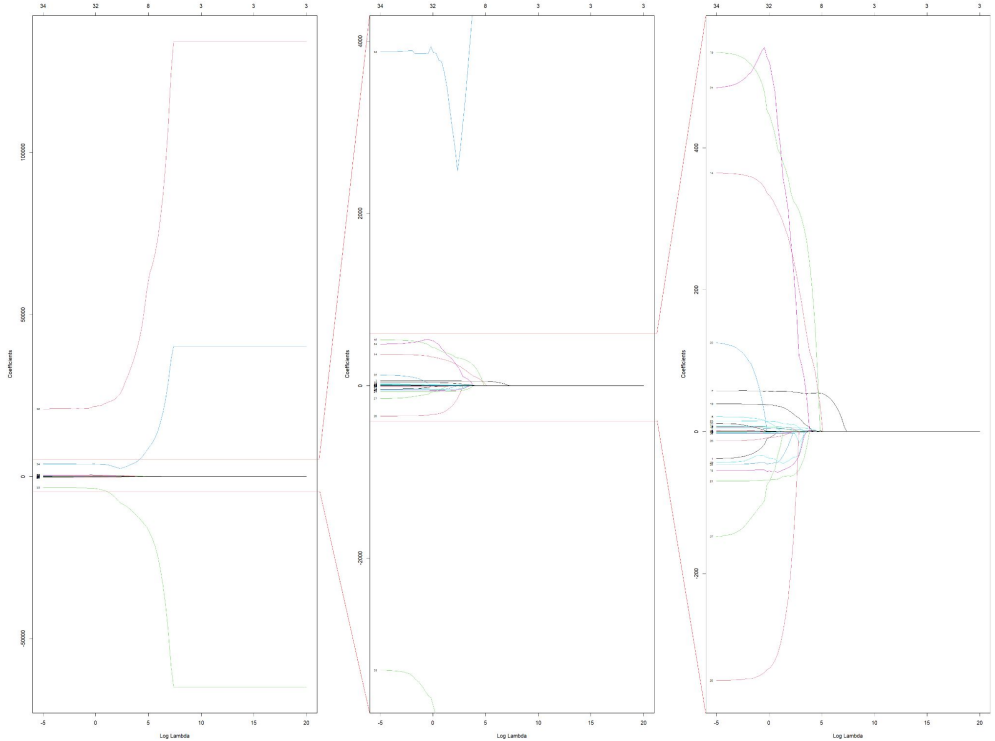
	Lambda	Index	MSE	SE	Nonzero
min	0.0067	100	223119	5062	34
1se	16.91639	69	227418	5494	34



- **`lambda.min = 0.0067`**: valore di lambda che minimizza l'errore quadratico medio (MSE) di cross-validazione;
- **`lambda.1se = 16.916`**: 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, spesso preferito per bilanciare l'accuratezza predittiva con la parsimonia del modello.

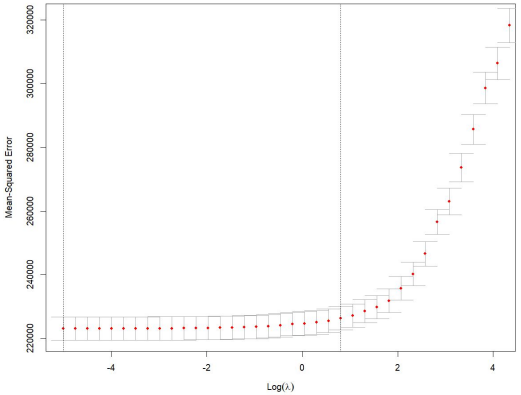
Risultati LASSO

Evoluzione dei coefficienti della Regressione LASSO al variare di λ



È stata eseguita cross-validazione a 10 fold utilizzando **`cv.glmnet`** e con output:

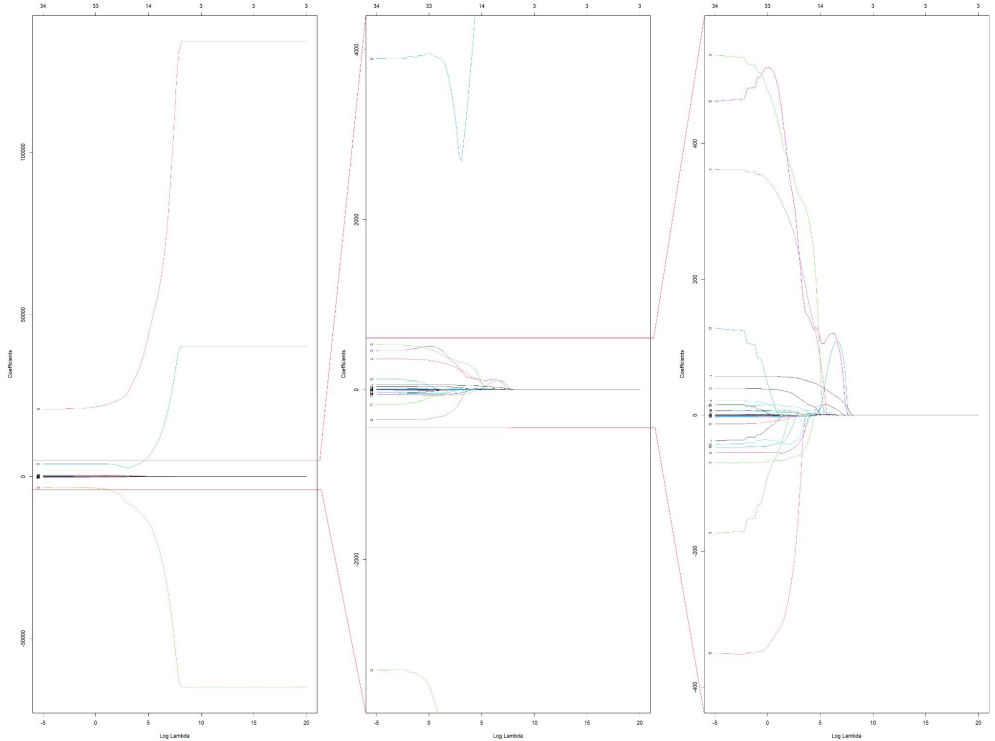
	Lambda	Index	MSE	SE	Nonzero
min	0.0067	100	223092	3680	34
1se	2.2436	77	226368	3634	29



- **`lambda.min = 0.0067`**: valore di lambda che minimizza l'MSE (ma che non fa selezione perchè troppo piccolo);
- **`lambda.1se = 2.2436`**: identifica il modello più semplice il cui MSE rientra in una deviazione standard dall'MSE minimo. Con questo lambda il numero di coefficienti non nulli (intercetta esclusa) è 29, indicando una selezione di variabili.

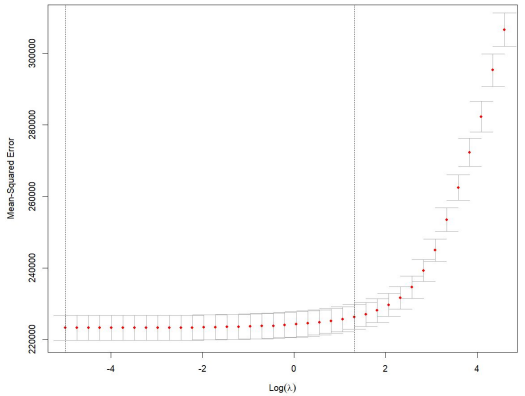
Risultati Elastic Net

Evoluzione dei coefficienti della Regressione Elastic Net al variare di λ



È stata eseguita cross-validazione a 10 fold utilizzando **`cv.glmnet`** e con output:

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



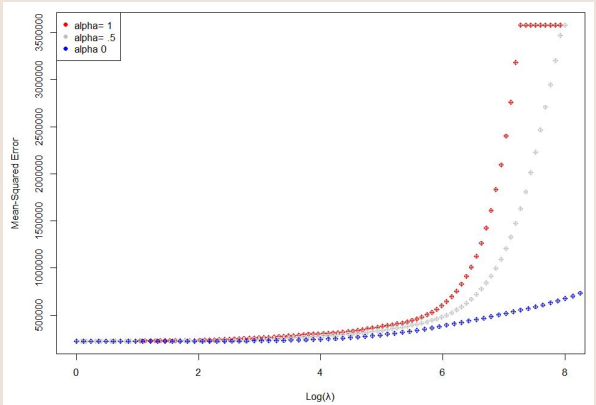
- **`lambda.min = 0.0067`**: valore di lambda che minimizza l'MSE (ma che non fa selezione perchè troppo piccolo);
- **`lambda.1se = 3.7178`**: identifica il modello più semplice il cui MSE rientra in una deviazione standard dall'MSE minimo. Con questo lambda il numero di coefficienti non nulli (intercetta esclusa) è 30, indicando una selezione di variabili.

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

Confronto tra i modelli penalizzati

Modello	MSE (CV λ = lambda_min)	MSE (Validation Set)	Coefficienti non nulli (λ = lambda_1se)
Ridge	223119	225852.7	34
Lasso	223092	226091.3	29
Elastic Net	223267	225973.7	30

Cross-validation con un set di `foldid` comune ed in corrispondenza di `lambda.min`:



- **LASSO (α=1):** MSE = `224561.8`
- **Elastic Net (α=0.5):** MSE = `223834.5`
- **Ridge (α=0):** MSE = `222823.5`

Confronto finale

ELASTIC NET

Il modello Ridge ha mostrato l'Errore Quadratico Medio (MSE) di cross-validazione 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), ha mostrato un MSE leggermente superiore.

L'Elastic Net si posiziona molto vicino al Ridge in termini di accuratezza predittiva, ma offre un vantaggio significativo nella parsimonia, selezionando 30 predittori non nulli (con λ1se). Questo lo rende un eccellente compromesso tra la minimizzazione dell'errore (performance predittiva) e la semplicità del modello (interpretabilità).

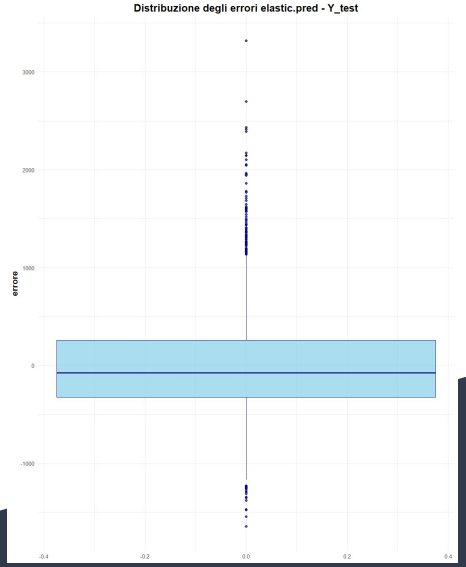
Ulteriore conferma è la somiglianza tra le variabili che l'Elastic Net azzera e quelle che sono state escluse dai modelli di **`regsubsets`**.

Le variabili escluse dall'Elastic Net riflettono una selezione coerente con la ricerca di un modello parsimonioso e ben bilanciato, indicando che l'Elastic Net non solo gestisce efficacemente il compromesso tra le penalità L^1 ed L^2, ma identifica anche un sottoinsieme di variabili fondamentali che altri metodi parsimoniosi tenderebbero a favorire.

Valutazione del Modello Finale sul Set di Test Indipendente

Il modello Elastic Net, adattato con il parametro di regolarizzazione **`lambda.1se`**, 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`.



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

- **Minimo:** `-1646.83` ,
- **1° Quartile:** `-324.60`,
- **Mediana:** `-76.94`,
- **Media:** `-6.64` (la media vicina a zero suggerisce che il modello non ha un bias sistematico),
- **3° Quartile:** `259.39`,
- **Massimo:** `3317.73` (il modello ha sovrastimato di oltre 3300 trofei in alcuni casi).

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.

GRAZIE PER L'ATTENZIONE