

Inside the World of Pokémon: a Machine Learning Project

Dario Bertazioli¹, Alessandro Borroni¹, Fabrizio D'Intinosante¹, Mirko Giugliano^{1*}, Massimiliano Perletti¹

Sommario

I Pokémon rappresentano ad oggi il secondo franchise videoludico per volume di vendite al mondo, che si è evoluto dagli anni 90 fino ai giorni nostri, attraversando numerosi cambi di piattaforma fino ad approdare nel 2018, oltre che sulle classiche console portatili marchiate Nintendo, anche sull'ibrido Nintendo Switch. Non stupisce quindi la forte componente competitiva venutasi a creare in questi ultimi anni, basata sulla possibilità per i giocatori di affrontarsi dapprima gli uni contro gli altri in locale, aumentando quindi la longevità dei titoli che in precedenza erano limitati allo svolgimento della storia in single player, ed oggi anche con la possibilità di giocare in remoto, contro avversari provenienti da tutto il pianeta. Quello degli e-sports è un tema profondamente fluido e dinamico, in pieno fermento negli ultimi anni, che sta riscuotendo da un lato sempre più successo a livello di partecipanti e di pubblico e dall'altro, di conseguenza, l'avvento di sponsor e grandi capitali. In quest'ottica risulta quindi stimolante poter applicare l'occhio e gli strumenti di un data scientist, con la massima cautela, ad un campo così variegato e complesso, partendo da quello che è stato ed è tutt'oggi un brand cult della cultura videoludica come quello dei Pokémon, una vera e propria pietra miliare per un'intera generazione di videogiocatori.

¹ Università degli Studi di Milano Bicocca, CdLM in Data Science.

*Corresponding author: m.giugliano@campus.unimib.it

Indice

1	Introduzione	1
2	Classificazione di <i>isLegendary</i>	2
2.1	Preprocessing	2
	Partizionamento • Approccio Preliminare	
2.2	Feature Selection	3
2.3	Classificazione	4
2.4	ROC Curves	4
3	Classificazione di "<i>rarity_class</i>"	5
3.1	Preprocessing	5
	Discretizzazione • Missing Replacement • Approccio Preliminare	
3.2	Feature Selection	6
3.3	Classificazione	6
3.4	ROC Curves	7
4	Conclusioni	7
	Riferimenti bibliografici	7

1. Introduzione

Quello dei Pokémon è un franchise giapponese di grande successo di proprietà di The Pokémon Company che fonda il suo successo, oltre che sulla serie animata, anche su tutta una serie di videogiochi realizzati in forma Role Playing Game (RPG) a partire dal 1996 e pubblicati da Nintendo. Nello specifico

con Pokémon ci si riferisce a creature di fantasia che vivono nell'immaginario "mondo Pokémon". La componente competitiva dei videogiochi, diventata sempre più preponderante andando avanti con il tempo, rende quella dell'analisi delle statistiche di questi "animali virtuali" un aspetto molto interessante da approfondire con il fine di classificarli proprio sulla base delle loro caratteristiche peculiari. Mossi dalla curiosità di esplorare a fondo questo mondo ci siamo proposti diverse domande di ricerca, tra cui:

- È possibile classificare i Pokémon cosiddetti "leggendari" (ovvero le creature simbolo di ogni generazione che dovrebbero rappresentare delle specie di divinità all'interno dei videogames) sulla base delle statistiche con le quali sono caratterizzati? Oppure la loro importanza è dovuta solo alla difficoltà che si ha nella loro cattura?
- Dall'altro lato, invece, è possibile ricondurre la difficoltà in cui ci si può imbattere nel trovare e catturare un Pokémon alle sue statistiche oppure è un parametro scelto a priori dagli sviluppatori, completamente legato da quelle che sono le effettive capacità delle creature?

Allo scopo di rispondere a queste domande abbiamo utilizzato il dataset "*pokemon_alopez247.csv*"[1] composto dai 721 Pokémon realizzati fino alla sesta generazione, caratterizzati da 22 attributi:

- *Name*: Nominale. Il nome di ogni Pokémon.

- *Type_1*: Categorica. Il tipo del Pokémon. Rappresenta la sua natura, il suo stile di vita e le abilità che è in grado di apprendere per i combattimenti. Può assumere 18 diversi valori.
- *Type_2*: Categorica. I Pokémon possono avere una doppia natura, nativamente o in seguito a una evoluzione. Non tutti possiedono una seconda natura.
- *Total*: Numerica. È il risultato della somma di tutte le caratteristiche di combattimento di un Pokémon. Rappresenta un buon indicatore della forza complessiva.
- *HP*: Numerica. Rappresenta i punti salute del Pokémon, più sono numerosi maggiore è il danno che può sostenere durante un combattimento.
- *Attack*: Numerica. Valore di attacco base, più è elevata maggiore è il danno che il Pokémon è in grado di infliggere.
- *Defense*: Numerica. Valore di difesa di base, più è elevata minore è il danno che il Pokémon riceverà dagli attacchi avversari.
- *Sp_Atk*: Numerica. Ha la stessa caratterizzazione di “Attack” ma vale per gli attacchi speciali.
- *Sp_Def*: Numerica. Ha la stessa caratterizzazione di “Defense” ma vale per gli attacchi speciali avversari.
- *Speed*: Numerica. Più è elevata maggiore sarà il numero di attacchi che il Pokémon potrà sferrare durante un combattimento.
- *Generation*: Numerica. La generazione in cui i Pokémon sono stati realizzati.
- *isLegendary*: Booleana. Indica se il Pokémon è leggendario o meno.
- *Color*: Nominale. Colore del Pokémon in accordo con il Pokédex.
- *hasGender*: Booleana. Indica se il Pokémon può essere classificato come maschio o femmina.
- *Pr_Male*: Numerica. Nel caso in cui il Pokémon abbia genere, la probabilità che sia maschio.
- *Egg_Group_1*: Nominale. Indica il gruppo uovo del Pokémon.
- *Egg_Group_2*: Nominale. Indica un eventuale secondo gruppo uovo del Pokémon. Non è presente per tutti i Pokémon.
- *hasMegaEvolution*: Booleana. Indica se il Pokémon dispone o no di mega evoluzioni.
- *Height_m*: Numerica. Indica l’altezza in metri del Pokémon in accordo con il Pokédex.

- *Weight_kg*: Numerica. Indica il peso in chilogrammi del Pokémon in accordo con il Pokédex.
- *Catch_Rate*: Numerica. È un coefficiente moltiplicativo della probabilità di esito positivo del processo di cattura di un Pokémon. Assume valori tra 3 e 255.
- *Body_Style*: Nominale. Indica la fisionomia del Pokémon in accordo con il Pokédex.

2. Classificazione di *isLegendary*

Per rispondere alla prima domanda di ricerca, ovvero se sia possibile classificare efficacemente i Pokémon “legendari” sulla base delle loro statistiche di combattimento, abbiamo seguito il seguente schema.

2.1 Preprocessing

Per la classificazione di “isLegendary”, basata sulle statistiche dei singoli Pokémon dei 22 attributi iniziali, ne escluderemo 13 a priori:

Body_Style, Color, Name: poiché non hanno alcuna utilità per la classificazione.

Generation: i nostri obiettivi di ricerca non includono una ricerca basata sulla generazione dei Pokémon.

Egg_Group_1, Egg_Group_2: dato che alcuni Pokémon non possiedono sesso e non si riproducono ma esistono in quanto esemplari unici, questi attributi sarebbero di difficile interpretazione e tenderebbero a distorcere la classificazione.

Type_1, Type_2, hasGender, Pr_Male, hasMegaEvolution: non hanno nulla a che fare con le pure statistiche combattive dei Pokémon ma rappresentano solo delle caratteristiche peculiari che non influenzano le prestazioni in combattimento.

Catch_Rate: non è legata a caratteristiche fisiche di un Pokémon, ma solo alla difficoltà di cattura, decisa a priori dai programmatori del gioco.

Total: Essendo la somma totale delle caratteristiche fisiche, è per noi di scarso interesse non distinguendo tra queste¹.

Nei rimanenti attributi abbiamo verificato l’assenza di *missing values*.

2.1.1 Partizionamento

Concluso questo procedimento, abbiamo proseguito estraendo dal dataset prima di tutto il validation set (20%) per effettuare efficacemente la procedura di *feature selection* e, successivamente, dal restante 80% del dataset abbiamo estratto il *training set* (67%) ed il *test set* (33%), eseguendo un campionamento stratificato rispetto a *isLegendary*. Prima di procedere alla classificazione però, basandoci sui risultati del-

¹Inoltre, avendo eseguito delle classificazioni includendo tale attributo, ci siamo resi conto che in ambito di *Feature Selection* tutti i wrapper avrebbero selezionato solamente la variabile *Total* (o quasi), ottenendo poi delle performance finali di accuracy tendente all’unità. Per questo motivo, non soddisfatti dall’eccessivo peso riassuntivo di tale variabile, abbiamo deciso di escluderla.

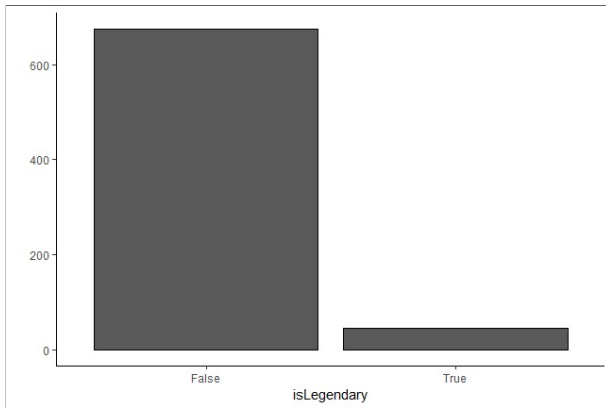


Figura 1. Distribuzione dell'attributo "isLegendary"

l'esplorazione preliminare del dataset² (visualizzabili nel metanodo denominato "Some_Vizs"), abbiamo deciso di eseguire un ricampionamento attraverso la tecnica dell'*oversampling* (implementata nel nodo Smote³) [2] sul *training set* e sul *validation set*. Questa ci ha permesso di campionare casualmente (con reinserimento) dalla classe minoritaria un numero di osservazioni tale da pareggiare la cardinalità della classe maggioritaria così da sopperire al forte sbilanciamento presente nella variabile "isLegendary", caratterizzata da una presenza del valore "True" del 6,8%, come mostrato in Fig. 1.

2.1.2 Approccio Preliminare

Con lo scopo di restringere la rosa di modelli per la classificazione, abbiamo inizialmente affrontato il problema testando 7 classificatori di diversa natura:

DecisionTree, *J-48* e *RandomForest* tra gli euristici,
Logistic e *SimpleLogistic* tra i regression based,
NaiveBayes e *NBTree* tra i probabilistici.

Al fine di eseguire la classificazione con algoritmi differenti tra loro, abbiamo deciso di selezionare i 4 classificatori che si sono rivelati i migliori tra quelli testati, scegliendoli in base al criterio dell'accuracy (vedasi Fig. 2) e dell'interpretabilità:

J-48, *RandomForest*
SimpleLogistic
NBTree

Notiamo anche che i valori di *F-measure*, riportati in Fig. 3, sono sistematicamente più bassi rispetto alle relative accuracy, ma ne seguono mediamente l'andamento: questo fatto non stupisce, in quanto l'attributo è estremamente sbilanciato nelle sue modalità e sul *test set* non viene effettuato *oversampling*.

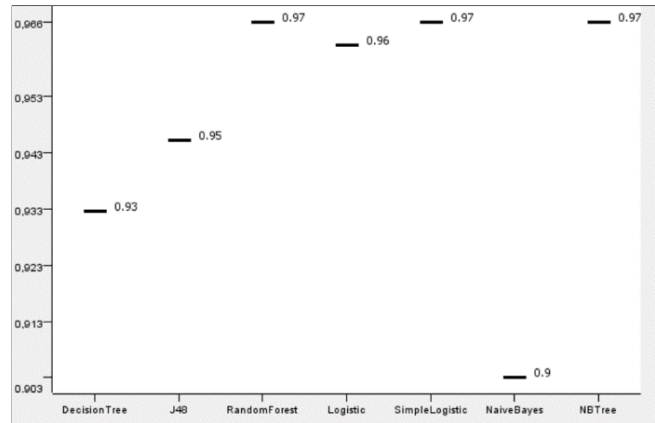


Figura 2. Confronto preliminare tra le accuracy di vari modelli di classificatori.

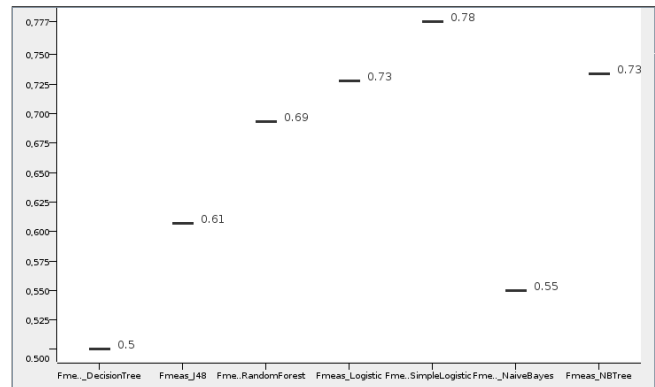


Figura 3. Performance preliminare: *F-measure* relativa ai modelli di classificazione testati.

2.2 Feature Selection

Al fine di ottimizzare le prestazioni dei classificatori abbiamo deciso di selezionare le variabili rilevanti per ogni modello che ci appresteremo a utilizzare per eliminare, così, attributi ridondanti o irrilevanti. L'approccio da noi scelto è quello del "wrapper" per cui è il classificatore stesso ad essere utilizzato per trovare il subset ottimale tra gli attributi disponibili, attraverso il nodo KNIME AttributeSelectedClassifier. Il classificatore viene istruito con il validation set con lo scopo di determinare, per ogni modello, il sottoinsieme ottimale di attributi. Come effetto collaterale di questa procedura, essendo stato ricavato il subset ottimale di attributi per ogni modello dall'utilizzo di questa porzione di dati, da qui in avanti quest'ultima non verrà utilizzata laddove si sfrutti la conoscenza ottenuta. In prima istanza esaminiamo i risultati della feature selection: per ogni classificatore sono stati scelti quasi sempre gli stessi attributi, segno che questi sembrerebbero essere piuttosto caratterizzanti rispetto agli altri. Per i vari modelli le feature selezionate sono:

- *J-48*: *Attack*, *Defense*, *Sp_Atk*, *Sp_Def*, *Speed*, *Height_m*.

²Abbiamo anche effettuato un'esplorazione di tipo grafico del dataset, mediante l'utilizzo del software Tableau, il cui risultato è accessibile al seguente link: https://public.tableau.com/views/Pokmon-MLproject/Introduction?:embed=y&:display_count=yes. Per una visualizzazione ottimizzata consigliamo di scaricare il font a tema Pokémon al link diretto riportato di seguito: <https://www.fontspace.com/jackster-productions/pokemon-gb>.

³<https://nodepit.com/node/org.knime.base.node.mine.smote.SmoteNodeFactory>

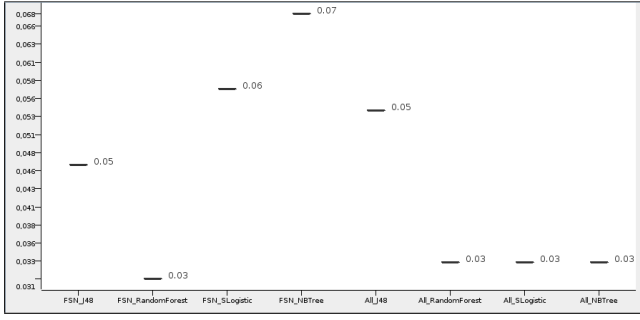


Figura 4. Errore ($1 - acc$) relativo ai modelli di classificazione adottati.

- *RandomForest*: *HP, Attack, Defense, Sp_Atk, Sp_Def, Speed, Height_m*.
- *SimpleLogistic*: *HP, Defense, Sp_Atk, Sp_Def, Speed, Height_m, Weight_kg*.
- *NBTtree*: *Defense, Sp_Atk, Sp_Def, Speed, Height_m, Weight_kg*.

2.3 Classificazione

Una volta selezionati gli attributi più appropriati per la classificazione abbiamo effettuato un confronto tra le performance dei diversi classificatori ottenute applicando la feature selection (utilizzando sia il metodo dell'*hold out* che quello del *K-Folds Cross Validation*) e quelle ottenute sfruttando tutti gli attributi.

Come riportato in Fig. 4, i risultati sono tra loro molto simili, anche considerando il fatto che la feature selection ha mediamente rimosso 1 o 2 attributi da quelli disponibili. L'utilizzo della *K-Folds Cross Validation* non mostra differenze significative rispetto alla stima con l'*holdout*. L'elemento più significativo risultante da questa analisi (implementando la *Feature Selection*) lo si ottiene osservando gli intervalli di confidenza relativi all'accuracy dei modelli adottati: nello specifico si può apprezzare un calo di prestazioni del *SimpleLogistic* e del *NBTtree* in favore degli algoritmi *J-48* e *RandomForest*, come riporta la Fig. 5.

Questo trend non si limita solo alla stima dell'accuratezza ma anche alla misura della *F-measure*; come mostrato in Fig. 6, in generale i classificatori regression based e probabilistici sembrano soffrire maggiormente la riduzione degli attributi disponibili rispetto ai classificatori euristici.

2.4 ROC Curves

Per ultimo abbiamo esaminato l'andamento dei classificatori tramite le *ROC curves* per avere un confronto tra le diverse probabilità di corretta classificazione degli eventi (*isLegendary = True*) al variare dell'errata classificazione dei non-eventi (*isLegendary = False*). La *ROC curve* infatti rappresenta sull'asse delle ordinate il numero dei record positivi di un certo subset dei dati espresso come percentuale del numero totale dei record positivi, mentre sull'asse delle ascisse il numero di record negativi del subset come percentuale di tutti i record

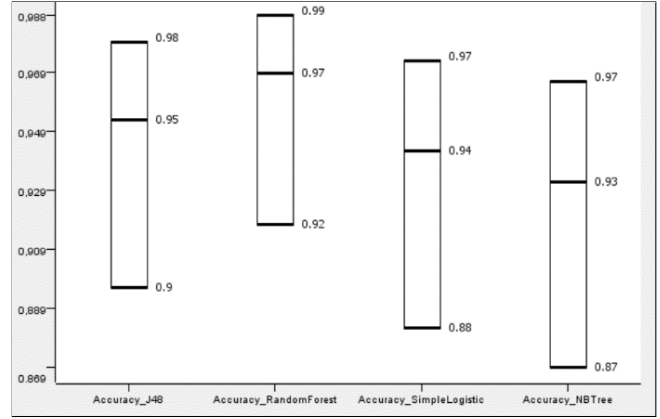


Figura 5. Intervallo di confidenza per l'accuracy relativo ai modelli di classificazione adottati (ottenuti con semplice *hold out*). L'intervallo in questione è ricavato con soglia di confidenza $1 - \alpha = 0.99$.

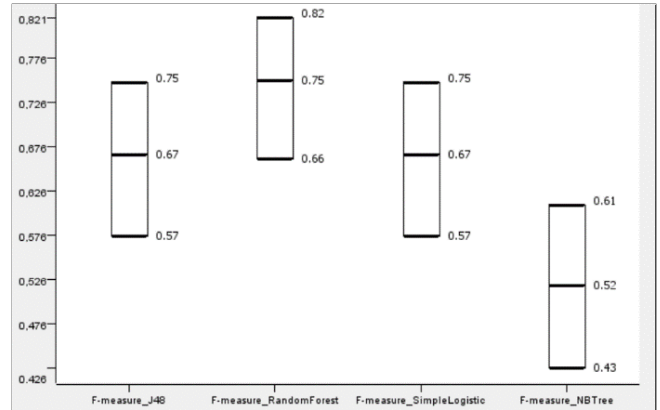


Figura 6. Come in Fig. 4 ma per le misure di *F-Measure*.

negativi. Tra i modelli testati, il modello con la probabilità più alta di identificare bene i Pokémon "legendari", basandosi sull'area sottostante la *ROC curve* riportata in Tab. 1, è il *SimpleLogistic*, mentre facendo riferimento esclusivamente ai valori raggiunti da misure di performance come accuratezza ed *F-measure* il migliore sembrerebbe essere il *RandomForest*. Tuttavia, osservando le stime degli intervalli di confidenza, la differenza tra i modelli non risulterebbe significativa.

P(isLegendary=True)	area sotto la curva
<i>J-48</i>	0,9299
<i>RandomForest</i>	0,8943
<i>SimpleLogistic</i>	0,9823
<i>NBTtree</i>	0,8568

Tabella 1. Area sottesa alle *ROC curves* per *isLegendary = True*.

3. Classificazione di “*rarity_class*”

3.1 Preprocessing

Per la classificazione di “*rarity_class*”, che si pone l’obiettivo di individuare quali aspetti concorrano a determinare la difficoltà di cattura di un Pokémon, escludiamo a priori 4 attributi dei 22 iniziali:

Body_Style, *Color*, *Name*: poiché non hanno alcuna utilità per la classificazione.

Catch_Rate, essendo la feature dalla quale è stata ricavata la variabile obiettivo “*rarity_class*”.

3.1.1 Discretizzazione

Nel videogioco in questione, quando un allenatore incontra un Pokémon selvatico interessante (ad esempio piuttosto raro, o piuttosto forte dal punto di vista delle statistiche), il giocatore può tentare di catturarlo: l’esito di tale processo è sempre incerto, e tale “suspense” è caratteristica di questa fase del gioco. L’esito di questo processo di cattura è influenzato da due fattori: in primo luogo dal Pokémon in questione (l’informazione della facilità o meno di cattura di uno specifico Pokémon è appunto codificata nella variabile *Catch_Rate*), mentre il secondo fattore avente un ruolo importante nella fase di cattura è un moltiplicatore random, che rende più casuale e quindi incerto il risultato, aumentando il divertimento (nonché gli scherzosi sfoghi d’ira da parte del giocatore, qualora il tanto desiderato Pokémon fugga evitando la cattura). Con lo scopo di svolgere un lavoro di classificazione riguardante *Catch_Rate* ma non volendosi rivolgere alla variabile codificata così come presente nel dataset, si è proceduto alla sua discretizzazione codificando una nuova variabile denominata “*rarity_class*”, categorica, con lo scopo di catalogare ogni Pokémon in una delle tre classi: “Common”, “Uncommon”, “Rare”. Stante quanto riportato sopra, questo processo di discretizzazione ha anche lo scopo di rendere più verosimile l’informazione ottenuta riguardo alla facilità di cattura di un Pokémon particolare: infatti, riducendo la variabilità iniziale di *Catch_Rate* (nonché la dimensionalità della stessa), si può considerare in un certo senso meno importante la randomicità del processo di cattura, giacché si ottiene una informazione meno predittiva nello specifico, ma “mediata” (più generale) e quindi più affidabile. La discretizzazione è avvenuta basandosi sui quantili visualizzati osservando la particolare funzione di ripartizione della variabile *Catch_Rate*, riportata in Fig. 7. Eseguendo dei binning manuali, abbiamo cercato di rispettare un criterio di “rarietà”:

- per la classe “Rare” sono stati scelti i Pokémon con valore di *Catch_Rate* < 45 , ovvero circa l’11% del dataset;
- per “Uncommon” sono stati selezionati quelli aventi valore di $45 \leq \text{Catch_Rate} < 90$ così da selezionare il successivo 47%;
- per i “Common” sono stati selezionati tutti quelli con *Catch_Rate* ≥ 90 , comprendendo così il rimanente 42% dei record.

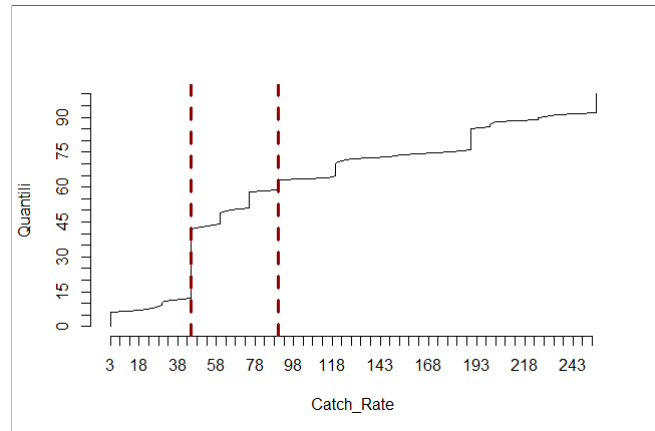


Figura 7. Funzione di ripartizione della feature *Catch_Rate*. Le linee verticali tratteggiate rappresentano la divisione in intervalli discreti associati ai valori della nuova variabile *rarity_class*.

Vale la pena di notare che la nuova variabile così ottenuta è stata utilizzata solo per questa seconda classificazione.

3.1.2 Missing Replacement

Una volta effettuato il *preprocessing*, procediamo all’imputazione dei dati mancanti presenti nelle variabili *Type_2*, *Pr_Male* e *Egg_Group_2*. Avendo una conoscenza del dominio si è deciso di procedere per imputazione manuale laddove vi sia la presenza di missing value, trattandosi di poche osservazioni. Per i Pokémon sprovvisti di *Type_2*, dato che sono molti, si è creata una nuova classe denominata “NoOne” proprio perché il fatto stesso che un Pokémon non possieda un secondo tipo rappresenta di per sé un’informazione. In modo analogo si è proceduto per *Egg_Group_2*. Per *Pr_Male*, in corrispondenza dei Pokémon per cui non era specificato il sesso, ovvero per quelli con *hasGender* uguale a “false” ci si è limitati ad inserire *Pr_Male* = 0, poiché non vi è alcuna possibilità di rintracciarli in versione maschile se questi ultimi non possiedono una distinzione sessuale.

3.1.3 Approccio Preliminare

Una prima fase di approccio alla classificazione consiste in una valutazione del comportamento di diversi classificatori, che abbiamo testato in modo preliminare utilizzando tutti i records (divisi tra training e test set seguendo la consueta proporzione del 67% e 33%).

Abbiamo quindi selezionato quattro modelli di classificazione, scelti in base alle migliori performance iniziali, riportate in Fig. 8: *J-48*, *RandomForest*, *SimpleLogistic*, *NBTree*. In particolare, a un primo approccio tali classificatori sembrano equivalentemente performanti: ci proponiamo quindi di investigare ulteriormente, con metodi di apprendimento più sofisticati e affidabili, per tentare di stabilire se uno dei quattro modelli possa risultare migliore degli altri.

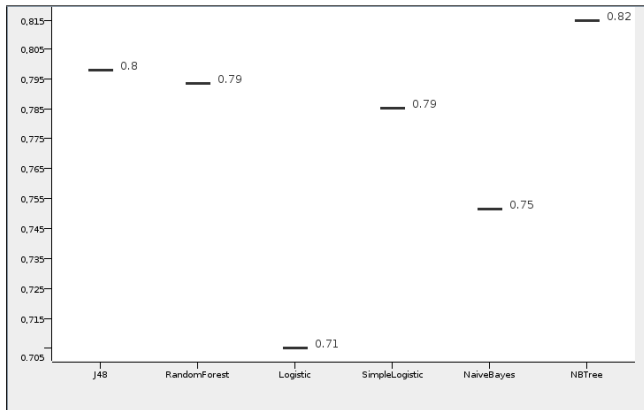


Figura 8. Confronto preliminare tra le accuracy di vari modelli di classificazione inizialmente testati.

3.2 Feature Selection

Partizioniamo inizialmente il dataset estraendo tramite un campionamento stratificato (rispetto a *rarity_class*) un *Validation Set* composto dal 15% dei records. Utilizzando questo sottoinsieme, procediamo quindi con la *Feature Selection*, sfruttando dei wrappers che restituiscono per ciascun classificatore le seguenti variabili valutate più rilevanti:

- *J-48*: *Total*, *hasGender*, *Pr_Male*, *hasMegaEvolution*;
- *RandomForest*: *Total*, *HP*, *Sp_Atk*, *hasGender*, *Pr_Male*, *hasMegaEvolution*, *Height_m*;
- *SimpleLogistic*: *Total*, *Defense*, *Sp_Atk*, *Pr_Male*;
- *NBTree*: *Type_2 Total*, *Pr_Male*, *Egg_Group_1*, *hasMegaEvolution*.

Possiamo notare come alcuni attributi vengano selezionati in tutti i classificatori: in particolare, risultano comunemente rilevanti *Total*, come da aspettative essendo una variabile riassuntiva delle statistiche di ciascun Pokémon, e dunque molto descrittiva, e *Pr_Male*. La selezione di quest'ultimo attributo si spiega considerando che i Pokémon Leggendari, ovvero molti tra i più rari, non hanno un genere definito, per cui per costoro la probabilità di avere sesso maschile risulta caratteristicamente $Pr_Male = 0$, come accennato nella Sez. 3.1.2.

In seguito compareremo le performance ottenute utilizzando questi attributi selezionati con quelle ricavate sfruttando tutte le feature disponibili.

3.3 Classificazione

Partizioniamo ulteriormente (sempre con *stratified sampling*) i restanti records (scartando quindi il *validation set* usato per la *Feature Selection*) in un *training set* (67% dei records) e un *test set* (33%). Procediamo quindi con l'addestramento dei classificatori, utilizzando sostanzialmente l'approccio della *K-FOLDS Cross Validation* con $k = 10$ iterazioni. Come misure di performance abbiamo analizzato l'errore risultante dalla

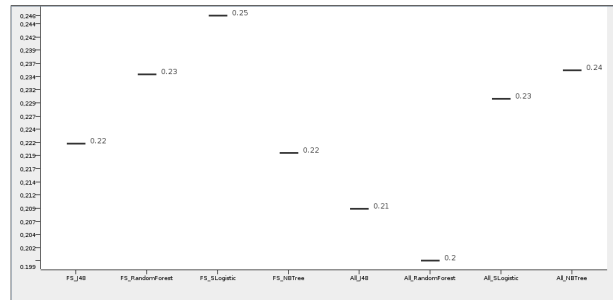


Figura 9. Performance: errore ($1 - acc$) per ogni classificatore, dopo l'apprendimento tramite *Cross Validation*. Si possono confrontare i risultati ottenuti con (voci con prefisso *FS*) e senza (voci con prefisso *All*) Feature Selection.

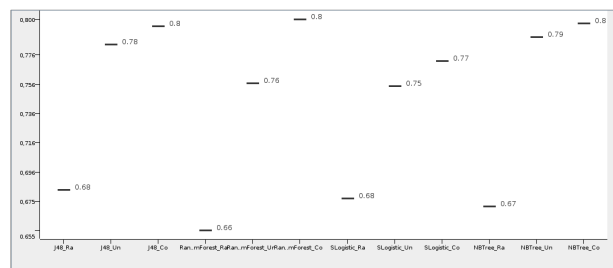


Figura 10. Performance: *F-measure* per ogni classificatore, dopo l'apprendimento tramite *Cross Validation*.

classificazione (inteso come $1 - acc$ in %) e la *F - Measure*, poiché il *test set* risultante dalle partizioni è composto da pochi records e la classe *Rare* risulta leggermente sbilanciata. Otteniamo le performance riportate in Fig. 9.

Possiamo notare che le performance relative alla classificazione dei records appartenenti alla classe *Rare* sono sistematicamente peggiori rispetto a quelle relative alle altre classi: questo risultato è spiegato con lo sbilanciamento del dataset relativamente a questa classe, che risulta leggermente sotto-rappresentata, fatto che provoca un peggior apprendimento del classificatore. Nonostante ciò, per tutti e quattro i classificatori le performance sono paragonabili: in prima analisi non si distingue un classificatore nettamente migliore degli altri. Per approfondire ulteriormente la comparazione tra classificatori, riportiamo in Fig. 11 un'analisi dei *confidence interval* relativamente all'accuracy dei modelli [3]. Notiamo che la performance media dei classificatori si assesta in tutti i casi intorno a $acc_{mean} \approx 0.8$, e possiamo notare un leggero ma generale aumento delle performance laddove sia stata applicata la *Feature Selection*, segno che la riduzione della complessità del sistema non ha portato a una perdita di informazione, ma ha anzi permesso di mantenere performance simili (addirittura leggermente migliori) e di ottenere un risultato decisamente più interpretabile.

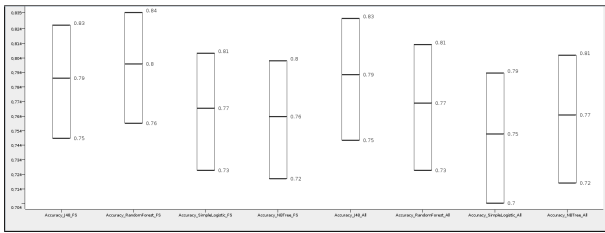


Figura 11. Confronto dei confidence interval (accuracy) per ogni classificatore, sia nel caso di *FeatureSelection* (FS) sia nel caso di inclusione di tutti gli attributi (All). Si può notare che gli intervalli di confidenza si sovrappongono reciprocamente, segno che i risultati dei vari classificatori sono compatibili, e i modelli stessi risultano equivalenti per questo tipo di problema. L'apprendimento avviene tramite *Cross Validation*.

Classificatore	$P(Uncommon)$	$P(Common)$	$P(Rare)$
J-48	0.855	0.903	0.843
RandomForest	0.874	0.908	0.920
SimpleLogistic	0.825	0.896	0.891
NBTree	0.856	0.896	0.890

Tabella 2. Area sottesa alle *ROC curves* per ogni modalità di *rarity_class*.

3.4 ROC Curves

Analizzando le *ROC Curves*, si osserva una differenza piuttosto significativa tra le aree sottese alle *ROC Curves* corrispondenti alle diverse modalità di *rarity_class*, come mostrato in Tab. 2. Sembrerebbe infatti che la classe *Uncommon* risulti di più difficile classificazione per via della sua posizione intermedia tra i due intervalli estremi *Common* e *Rare*. Infatti, questi ultimi sono meglio caratterizzati proprio a causa della loro natura estrema. Emergono quindi i limiti di una discretizzazione non supervisionata ma effettuata manualmente, seppur guidata da osservazioni qualitative (come riportato in Fig. 7).

4. Conclusioni

In conclusione riportiamo sinteticamente le risposte alle principali domande di ricerca.

- È possibile individuare i Pokémon leggendari solamente in base alle loro statistiche “fisiche”?

La risposta è sì, e con risultati piuttosto consistenti: infatti, l'accuracy (nonché la misura di *F-Measure*) raggiunta dai classificatori per tale compito è piuttosto alta, e tutti i modelli di classificazione usati per l'analisi si sono rilevati sostanzialmente equivalenti e ben performanti. In particolare abbiamo scoperto che bastano pochi attributi comuni selezionati attraverso *Feature Selection* per ottenere ottimi risultati di classificazione, segno che il risultato è chiaramente interpretabile tramite il significato che questi attributi assumono nel mondo dei Pokémon. In linea di massima possiamo affermare

che i Pokémon etichettati -dai creatori del gioco- come leggendari non posseggono tale titolo solo per ragioni di trama o convenienza, ma perché dotati a tutti gli effetti di statistiche di combattimento generalmente piuttosto elevate.

- È possibile predire in base alle statistiche generali la facilità di cattura di un Pokémon?

La risposta è (parzialmente) sì: con la classificazione multibinaria che abbiamo effettuato, avente come target la variabile *rarity_class* da noi creata, abbiamo ottenuto un'accuracy media dell'ordine di $acc_{mean} \approx 0.8$, comune a tutti i classificatori adottati. Tale risultato indica che i relativamente pochi attributi selezionati (dalla *Feature Selection*) influenzano in modo significativo il processo di cattura.

I risultati sono probabilmente incrementabili applicando direttamente una regressione alla variabile continua *catch_rate*, come eseguito ad esempio in [4], tuttavia la risposta alla domanda principale sarebbe in questo frangente offuscata (se non contraddetta, in alcuni casi) dal peso del fattore random che gioca un ruolo importante durante il processo di cattura. Con la nostra discretizzazione pensiamo di ottenere un risultato più affidabile in termini di riscontro nel gioco, benché meno predittivo nello specifico.

Riferimenti bibliografici

- [1] <https://www.kaggle.com/alopez247/pokemon>
- [2] Brownlee J. 8 tactics to combat imbalanced classes in your machine learning dataset. <https://machinelearningmastery.com>, 2015.
- [3] Tan P.-N. Steinbach M. Kumar V. Introduction to data mining. <http://www.uokufa.edu.iq/staff/ehsanali/Tan.pdf>, 2006.
- [4] Statistical analysis of Pokémon, Asier López Zorrilla, <https://www.kaggle.com/alopez247/pokemon>, 2017.