Verità o pregiudizi del mondo Pokémon

Francesca Galletti[1], Christian Internò[1], Andrea Marinoni[1], Allegra Sotgiu[1]
[1] Università degli studi di Milano Bicocca, CdLM Data Science

Abstract
La domanda a cui si è voluto rispondere riguarda la possibilità di capire dal primo approccio con un Pokémon, esclusivamente dalle sue caratteristiche estetiche, se questo risulterà essere forte o addirittura leggendario.
Dopo aver osservato le variabili di interesse all'interno del dataset, paragonabile al Pokédex, si è deciso di selezionare le variabili che possono essere lette nella schermata iniziale di "lotta" nei videogiochi Pokémon. Si sono in seguito applicate tecniche di Machine Learning con l'obiettivo di prevedere la forza della creatura, calcolata come la somma delle statistiche di lotta nella variabile '*Total'* e se sia leggendario o meno, basandosi sulla variabile booleana '*isLegendary',* riuscendo in entrambi casi a ottenere buoni risultati.

KeyWords
Machine Learning – Classificazione – Pokémon – Pregiudizi

INDICE

1. INTRODUZIONE

Nel 1996 la Nintendo pubblicò due videogiochi del creatore Satoshi Tajiri per la console Game Boy, Pokémon versione rossa e Pokémon versione blu[1]. Questi due videogame vedevano come protagonisti delle creature immaginarie che il giocatore poteva catturare e far combattere. Nelle prime versioni del gioco erano disponibili 150 creature. Dal 1996 a oggi il mondo dei Pokémon si è sviluppato anche in altri mercati, come quello degli anime, del cinema, delle carte di gioco collezionabili e diverse collaborazioni. Nintendo ha sviluppato successive versioni del gioco implementando nuove creature. Nel 2016 è stato rilasciato Pokémon Go[2] che sfruttando la realtà aumentata e il GPS, portava il giocatore a vivere un'esperienza unica trasformandolo in un vero e proprio cacciatore di Pokémon con l'ausilio del proprio smartphone.
I Pokémon sono creature dalle caratteristiche e poteri differenti, ve ne sono di più forti e tra questi ci sono quelli denominati leggendari[3].
Vista la varietà di colori e la simpatia che suscitano le forme di queste creature, viene spontaneo chiedersi se ci sia un modo per riconoscere da un primo approccio la pericolosità di un Pokémon o alternativamente

se risulta valido anche nel loro mondo il detto 'l'abito non fa il monaco'.
Quindi, analizzando le caratteristiche osservabili dal primo incontro, come ad esempio altezza, peso e colore, è possibile predire se si tratta di un Pokémon particolarmente forte? Inoltre, utilizzando le stesse variabili, è possibile predire se un Pokémon è leggendario?

Lo scopo principale di questo studio è quindi quello di cercare di capire quanto l'aspetto estetico o di primo impatto di un Pokémon sia indicativo della sua forza.

2. PRESENTAZIONE DATA SET

Il dataset[4] preso in considerazione presenta 21 variabili per 721 Pokémon. Descriviamo di seguito le variabili:

- *Number.* ID associato al Pokémon
- *Name*. Nome del Pokémon (Nominale)
- *Type_1*. Indica il tipo di Pokémon e le sue abilità. Assume 18 valori (Categorica)
- *Type_2*. Alcuni Pokèmon possono assumere una seconda natura (Categorica)
- *Total*. E' la somma di tutte le caratteristiche di un Pokémon. Il range da 180 a 720 (Numerica)
- *HP*. E' il valore dei punti salute. Più è elevato più il Pokémon subirà meno danni durante un attacco. (Numerica)
- *Attack*. E' il valore associato alla potenza di attacco di un Pokémon. (Numerica)
- *Defense*. E' il valore associato alla capacità di difesa di un Pokémon. Più il valore è alto più la creatura subirà meno danno durante gli attacchi avversari (Numerica)
- *Sp_Atk*. E' relativa alla potenza di attacchi speciali (Numerica)
- *Sp_Def*. E' relativa alla difesa associata agli attacchi speciali degli avversari (Numerica)
- *Speed*. Questo valore è correlato a numero di attacchi che un Pokémon può sferrare nei combattimenti. Più è alta meno saranno i numero di attacchi (Numerica)
- *Generation*. E' la generazione di appartenenza del Pokémon (Numerica)
- *isLegendary*. Indica se la creatura è leggendaria oppure no (Booleana)
- *Color*. E' il colore del Pokémon (Nominale)
- *hasGender*. Indica il genere del Pokémon (Booleana)
- *Pr_male*. Correlata con la precedente indica la probabilità che il genere sia maschile (Numerica)
- *EggGroup1*. Indica il gruppo dell'uovo a cui appartiene il Pokémon (Nominale)
- *EggGroup2*. Alcuni Pokémon hanno due uova (Nominale)
- *hasMegaEvolution*. Alcuni Pokémon hanno mega evoluzioni, questa variabile indica se il Pokémon le ha oppure no (Booleana)
- *Height_m*. Altezza in metri (Numerica)
- *Weight_kg*. Peso in chilogrammi (Numerica)
- *Catch_Rate*. Possibilità di cattura di un Pokémon (Numerica)
- *Body_Style*. Indica la forma l'aspetto che il Pokémon può assumere (Nominale)

3. PRIMA DOMANDA

La prima domanda riguarda la possibilità di capire dal primo approccio con un Pokémon, se questo risulterà essere forte.
Sebbene la forza e l'abilità in combattimento siano fortemente legate al Pokémon specifico che si incontra, si può considerare come indicatore generale di forza la somma delle caratteristiche di lotta della creatura, riportata nel dataset preso in considerazione nella colonna *Total*.

3.1 PREPROCESSING

Per rispondere alla domanda di ricerca si è deciso di procedere con la binarizzazione della variabile *Total* considerando il valore medio del range pari a 450. I Pokémon con valore inferiore sono stati categorizzati con 'debole' e

gli altri con ‘forte’. Osservando che quelli che risultano forti sono il 46% e rispettivamente quelli deboli il 54% si può quindi considerare il data set come bilanciato.
Il passo successivo è stato quello di effettuare una selezione delle variabili di interesse: sono state considerate solo quelle relative alle caratteristiche che si possono osservare nelle varie versioni del videogioco nella schermata una volta avviato un incontro/combattimento con un Pokémon qualsiasi.

Figura 1. Schermata pokémon nel videogioco Pokémon Go.

Le variabili del dataset corrispondenti alle caratteristiche che si vogliono considerare sono: *Height_m, Weight_kg, Body_Style, hasGender, Type_1, Type_2, Pr_male, Color.*

Il Dataset ottenuto presentava diversi Missing value nelle colonne relative a *Type_2* (Categorica) la quale indica se alcuni Pokèmon assumano una seconda natura e *Pr_Male* (Numerica) indicante la probabilità che il genere sia maschile.

Per quanto riguarda la variabile *Type_2* i missing value sono stati sostituiti attraverso un valore fisso: ‘no’.
Questo è stato fatto perché la mancanza di una seconda tipologia risulta essere un’informazione rilevante ai fini dell’analisi.

Considerando la colonna *Pr_Male* i missing value erano associati al valore ‘false’ nella colonna *HasGender* ,ovvero quella che indica se quel pokémon ha un genere. In questo caso si è optato per aggiungere una nuova colonna al dataset, la quale indica la probabilità che il Pokémon sia femmina, creata come 1-*Pr_Male* e denominata *Pr_Female*.
In questo modo i missing value presenti nell’attributo *Pr_male* sono riportati anche nella colonna *Pr_Female*, ma sono in seguito stati sostituiti con 0 per indicare un pokemon senza genere ed evitare perdità di informazione.

3.2 MODELLI

Sono stati utilizzati per la nostra indagine i seguenti modelli di classificazione:

- Metodi Euristici: Non garantiscono di raggiungere risultati ottimali, permettono di ottenere soluzioni approssimate e comunque ragionevoli. Questi metodi fanno riferimento in particolar modo agli alberi decisionali (Random Forest) che vengono sviluppati a partire da un sottoinsieme di dati iniziali (training set) per il quale è nota la classe output. Tra questi modelli si e scelto di utilizzare il classificatore Random Forest e J48.
- Metodi basati sulla regressione: Simple Logistic
  Utilizzano la probabilità condizionata parametrica, servono per risolvere problemi di regressione binaria a diversi livelli. Sono applicabili ad attributi continui e con certe accuratezze anche ad attributi nominali.
- Metodi di Separazione: I metodi di separazione dividono lo spazio di un attributo in H regioni disgiunte, separando le osservazioni secondo la classe target. Ogni regione può comprendere un set ottenuto da operazioni di unione e intersezione di set teorici applicati a regioni di forma elementare. Viene definita una loss function per dare valore anche ai punti misclassificati; dopodiché viene risolto un problema di ottimizzazione ai fini di

determinare la suddivisione ottima che minimizzi la loss function globale. I metodi scelti sono SMO-poly, SMO-puk e MultiLayer Perceptron.

- Metodi Probabilistici, basati sul formula di Bayes: Tra i metodi più utilizzati vi sono quelli basati su ipotesi probabilistiche come il classificatore Naive Bayes e NBtree che permettono, a partire dall'elevata quantità di dati, di ottenere risultati accurati sotto ipotesi di ` indipendenza tra gli attributi.

3.3 LE MISURE DI PERFORMANCE

Per calcolare le performance dei classificatori si è optato di calcolare i seguenti indicatori: Recall, Precision, F1 Measure e Accuracy.

- Accuracy: è la proporzione tra il numero delle predizioni corrette e il totale delle predizioni:
  $Accuracy = \frac{TN + TP}{TN + FN + TP + FP}$

  Per TN e TP si indicano i valori che sono stati correttamente classificati come Negative e Positive. Al contrario con FN e FP si indicano quelli predetti erroneamente negativi o positivi

- Recall: misura la frazione dei record positivi correttamente predetti dal modello preso in considerazione. La formula per calcolare questo indicatore è:
  $r = \frac{TP}{TP+FN}$
  Un valore elevato indicherà che pochi record positivi sono stati erroneamente classificati

- Precision: misura la frazioni dei record realmente positivi che il modello ha inserito nel gruppo dei positivi. La formula per calcolare questo indicatore è:
  $p = \frac{TP}{TP+FP}$
  Più è alto il valore assunto da questo indicatore più sarà basso il numero di Falsi Negativi prodotti dal modello

- F1-Measure: questo indicatore è la media armonica dei due precedenti, Recall e Precision.
  $F1 = \frac{2rp}{r+p}$
  Un elevato valore dell'indicatore F1 Measure indica un elevato valore degli indicatori Recall e Precision

3.4 HOLDOUT

Il primo approccio utilizzato per valutare i modelli di classificazione utilizzati è il metodo Holdout. Questo prevede di partizionare il dataset in due sottoinsiemi disgiunti con rispettivamente il 67% dei dati (training set) e 33% (test set). Il metodo utilizzato per questo procedimento è lo stratified sampling, il quale considera i record in modo che all'interno del campione vengano rispettate le proporzioni tra gli attributi presenti nel dataset di partenza.
La variabile usata per la stratificazione è *Total*.
Nonostante la presenza di variabili nominali, è stato possibile utilizzare i metodi SMO, MultiLayer Perceptron e SimpleLogistic, che appartengono ai modelli di separazione o basati su regressione, in quanto i nodi Weka utilizzati gestiscono autonomamente i dati nominali o includono l'opzione Nominal to Binary che rende possibile il loro utilizzo[5].
I risultati ottenuti sono riportati in tabella.

Classificatore	Recall	Precision	F1-measure	Accuracy	AUC
Random Forest	0.7	0.729	0.714	0.765	0.876
J48	0.7	0,761	0.729	0.782	0.812
NBTree	0.75	0.743	0.746	0.786	0.8499
Naive Bayes	0.48	0.762	0.589	0.718	0.815
Simple Logistic	0.66	0.795	0.721	0.786	0.878
SMO poly	0.46	0.59	0.517	0.639	0.666
SMO-puk	0.47	0.443	0.456	0.529	0.523
MultiLayer Perceptron	0.51	0.548	0.528	0.618	0.769

Figura 2. Tabella risultati prestazioni dei modelli utilizzati.

In generale osservando i risultati sopra riportati è possibile apprendere come i modelli utilizzati siano accettabili.

Per avere ulteriore conferma della validità dei metodi utilizzati, si è deciso di effettuare un nuovo partizionamento dividendo il dataset iniziale in due insiemi disgiunti A e B, con A contenente 80% dei record e B il 20%. In questo modo A è stato utilizzato per ripetere la procedura Holdout e in particolare per creare l'intervallo di confidenza al 90% per l'Accuracy, usando il 67% dei dati come training set e il 33% come test set, e l'insieme B come set di validazione. I risultati sono riportati nel grafico.

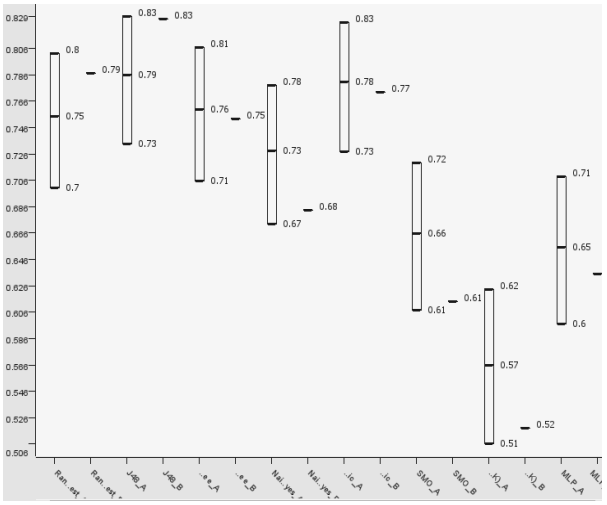

Figura 3. Intervalli di confidenza per l'Accuracy.

Come si può vedere dal grafico e dalla tabella il modello SMO-puk risulta avere un'accuratezza più bassa rispetto agli altri modelli utilizzati. I modelli migliori risultano RandomForest, J48, NBTree e Simple Logistic.

3.5 CROSS VALIDATION

Si è deciso di valutare ogni modello anche attraverso il metodo della cross validation con 10 fogli, ovvero dividendo il dataset in 10 parti di uguale cardinalità e ripetendo il procedimento utilizzando ogni volta uno di questi come test set e tutti gli altri come training set. In questo modo si ottengono 10 valori di accuracy che sono stati confrontati con un boxplot.

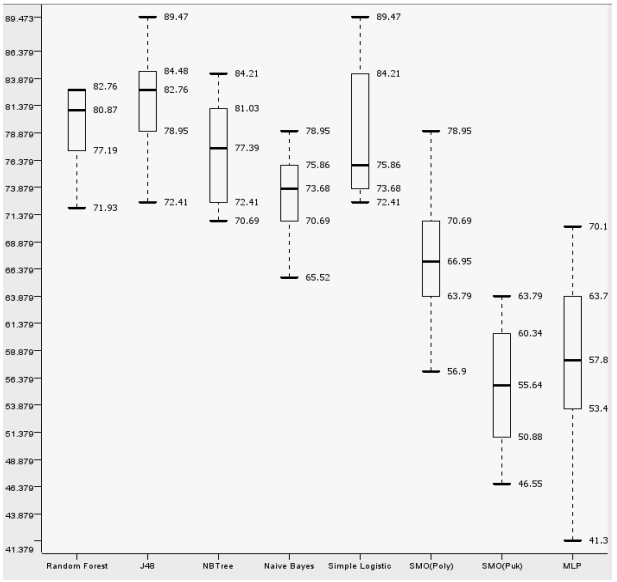

Figura 4. Boxplot valori accuracy per modello.

Osservando i risultati di entrambe le tecniche si può osservare che gli ultimi 3 modelli utilizzati (SMO-poly, SMO-puk e MultiLayer Perceptron) sono quelli che risultano avere accuratezza più bassa pertanto, per le analisi seguenti si terranno in considerazione solo i primi 5.

3.6 FEATURE SELECTION

Per aumentare l' interpretabilità dei dati e quindi capire quali siano le variabili, tra le 8 selezionate inizialmente, più significative per stabilire se un Pokémon sia forte, si è deciso di usare un filtro multivariato e in particolare il nodo Weka AttributeSelectedClassifier con il metodo CfsSubsetEval e BestFirst.
Si è scelto di utilizzare questo tipo di filtro in quanto riconosce anche gli attributi ridondanti, infatti tra le otto variabili ne sono presenti tre, *HasGender*, *Pr_male*, *Pr_female*, che sono evidentemente legate tra loro.

Utilizzando i 5 metodi ritenuti migliori in precedenza, le variabili selezionate dal filtro risultano essere *Height_m*, *Weight_kg* e *Pr_Female*.

Si osserva in tabella il miglioramento delle misure di performance ottenute dalla classificazione utilizzando solo le tre variabili più rilevanti.

Classificatore	Recall	Precision	F1-measure	Accuracy	AUC
Random Forest	0.789	0.796	0.793	0.811	0.885
J48	0.716	0.857	0.78	0.815	0.9
NBTree	0.697	0.854	0.768	0.807	0.892
Naive Bayes	0.523	0.919	0.667	0.761	0.866
Simple Logistic	0.716	0.867	0.784	0.819	0.914

Figura 5. Tabella risultati prestazioni dei modelli utilizzati con filtro multivariato.

Tutte le misure di performance subiscono dei cambiamenti migliorando leggermente rispetto ai modelli nei quali venivano considerate tutte le variabili.

4. SECONDA DOMANDA

Dopo aver stabilito che utilizzando le variabili selezionate è possibile capire con buona accuratezza se il Pokémon può essere considerato forte, viene automatico domandarsi se sia anche possibile stabilire, con le medesime variabili, l'appartenenza di un Pokémon alla classe dei leggendari, noti per essere tra i più forti in assoluto.

4.1 PREPROCESSING

Per rispondere a questa seconda domanda sono stati usati gli stessi attributi considerati nel rispondere alla prima, ovvero: *Height_m, Weight_kg, Body_Style, hasGender, Type_1, Type_2, Pr_male, Color*

Questa volta la classe da prevedere è *isLegendary* indicante se la creatura è leggendaria oppure no (Booleana). Le classi risultano sbilanciate, infatti solo il 6,4% dei Pokémon risultano leggendari.

4.2 HOLDOUT DATASET SBILANCIATO

Per valutare i modelli di classificazione è stato utilizzato il metodo Holdout.
Anche in questo caso la partizione del dataset è stata fatta attraverso il metodo di stratified sampling con il training set formato con il 67% dei record e il test set con il 33%. La variabile usata per la stratificazione è *isLegendary*.
La seguente tabella riporta i risultati:

Classificatore	Recall	Precision	F1-measure	Accuracy	AUC
Random Forest	0.733	0.917	0.815	0.979	0.911
J48	0.733	1	0.846	0.983	0.926
NBTree	0.8	0.857	0.828	0.979	0.973
Naive Bayes	0.6	0.474	0.529	0.933	0.939
Simple Logistic	0.8	0.706	0.75	0.966	0.892
SMO-poly	0.733	0.846	0.786	0.975	0.862
SMO-puk	0	0	?	0.933	0.498
MultiLayer Perceptron	0.733	0.917	0.815	0.979	0.943

Figura 6. Tabella risultati prestazioni dei modelli utilizzati Holdout.

Poichè il dataset è sbilanciato non è sufficiente fare un'analisi dell'accuracy per capire il buon funzionamento di un modello di classificazione: per questo motivo si è scelto di mostrare gli intervalli di confidenza al 95% relativi alla misura F1 in modo da tener conto di Recall e Precision.
Per fare questo, il dataset iniziale è stato diviso tramite campionamento stratificato in Partizione A, 80%, e Partizione B, 20%.
Alla partizione A è stato applicato il metodo Holdout con un training set formato dal 67% dei record e un test set dal 33% per creare l'intervallo di confidenza, la partizione B è stata usata per validare i metodi.

Nel grafico non sono stati riportati i valori relativi a SMO-puk poiché il metodo non riesce a classificare correttamente nessun Pokémon leggendario, ovvero TP=0.

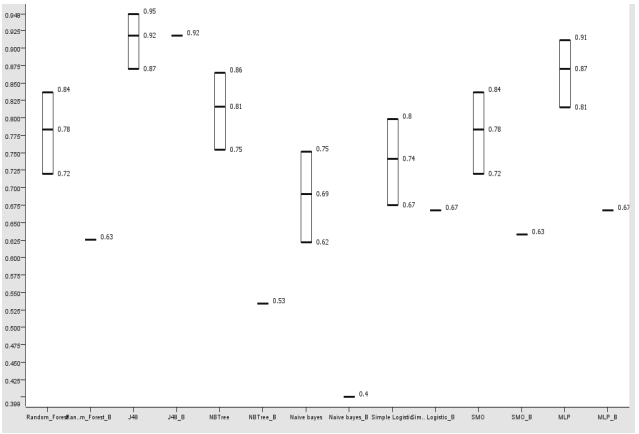

Figura 7. Intervalli di confidenza per F1-measure.

Quasi per ogni metodo il valore di F1-measure per la partizione B non ricade nell'intervallo di confidenza creato tramite la partizione A. La causa di questo potrebbe essere dovuta al forte sbilanciamento della classe *isLegendary* che rende i valori ottenuti estremamente dipendenti dalla partizione utilizzata.
Per ovviare al problema si è deciso di bilanciare l'attributo *isLegendary* attraverso il nodo Smote[6]: sono stati quindi duplicati i valori della classe minoritaria fino a renderli equiparabili a quella in origine maggioritaria.

4.3 HOLDOUT DATASET BILANCIATO

Con il nuovo dataset bilanciato si è proceduto con la riclassificazione attraverso gli stessi modelli usati precedentemente.
È possibile osservare i risultati ottenuti nella seguente tabella.

Classificatore	Recall	Precision	F1-measure	Accuracy	AUC
Random Forest	1	0.965	0.982	0.982	0.997
J48	1	0.929	0.963	0962	0.973
NBTree	0.991	0.969	0.98	0.98	0.998
Naive Bayes	0.91	0.914	0.912	0.913	0.958
Simple Logistic	0.996	0.937	0.965	0.964	0.981
SMO-poly	1	0.937	0.967	0.966	0.966
SMO-puk	1	0.996	0.998	0.998	0.998
MultiLayer Perceptron	1	0.937	0.967	0.966	0.98

Figura 8. Tabella risultati prestazioni dei modelli utilizzati con Dataset bilanciato.

Riutilizzando il procedimento già visto in precedenza, si osserva che gli intervalli di confidenza contengono i valori calcolati con la partizione B e quindi i metodi performano nettamente meglio con l'attributo *isLeggendary*.

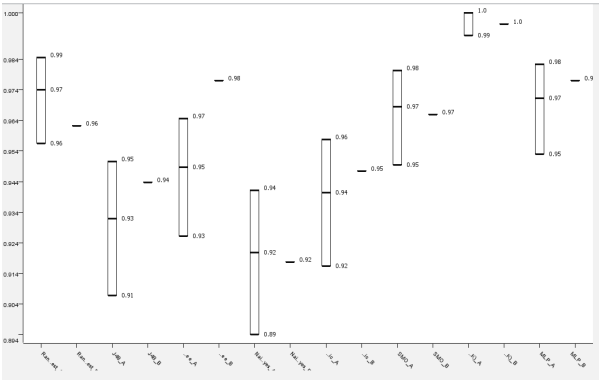

Figura 9. Intervalli di confidenza per F1-measure con dataset bilanciato.

4.4 ROC CURVE

L'ultima valutazione effettuata è fatta tramite la ROC curve, che permette di confrontare i metodi visivamente. Rappresenta, sull'asse verticale, il numero di record positivi inclusi in un sottoinsieme selezionato, espresso come percentuale del numero totale dei record positivi, e sull'asse orizzontale il numero di record negativi inclusi nel sottoinsieme, espresso come percentuale del numero totale di record negativi.

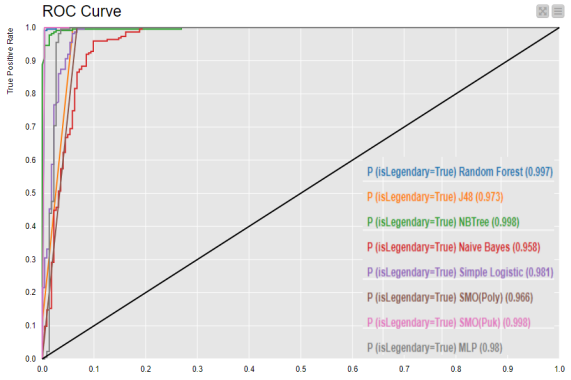

Figura 10. ROC curve

Dai grafici precedenti si osserva che i metodi migliori per risolvere questo problema sono MLP, SMO e RandomForest, ciò è confermato anche dalla ROC curve.

4.5 FEATURE SELECTION

Avendo utilizzato le stesse variabili per il primo e per il secondo problema è interessante capire se quelle più rilevanti per predire la

classe *Total* e la classe *isLegendary* sono le stesse. Applicando il filtro mutlivariato, come visto nel paragrafo 3.6, le variabili più significative risultano invece: *Type_1, hasGender, Height_m*.

5. CONCLUSIONI

In relazione alla prima domanda, attraverso la classificazione con il metodo Holdout e in particolare l'intervallo di confidenza al 90%, risulta che tutti i modelli utilizzati sono accettabili ma i modelli migliori sono i primi: Random Forest, J48, NBTree, NaiveBayes e Simple Logistic.
Per quanto riguarda l'utilizzo della Cross validation sono stati utilizzati dei boxplot per confrontare i risultati ottenuti e anche questa volta si osserva che tre (SMO-poly, SMO-puk e MultiLayer Perceptron) tra gli 8 modelli utilizzati presentano un accuracy più bassa, si è quindi deciso di considerare solo i rimanenti 5. Infine, si è analizzato quali tra le variabili utilizzate siano le più significative per stabilire se un Pokémon è forte, utilizzando un filtro multivariato. Le variabili che risultano più significative sono: *Height_m*, *Weight_kg* e *Pr_Female*. Procedendo con la classificazione dopo aver utilizzato il filtro si osserva un leggero miglioramento del livello di accuracy dei modelli utilizzati.

Rispondendo alla seconda domanda di ricerca posta si è in primis risolto il problema relativo alla classe d'interesse sbilanciata, poiché solo il 6,4% dei Pokémon risultano leggendari, rendendo i valori ottenuti estremamente dipendenti dalla partizione usata.
Utilizzando l'algoritmo SMOTE, sono stati duplicati i valori della classe minoritaria fino a renderli equiparabili a quella in origine maggioritaria, rendendo le classi bilanciate. I migliori modelli risultano: MLP, SMO e RandomForest.
In questo caso le variabili estratte dal filtro sono *Type_1, hasGender, Height_m*, questo implica che, nonostante i Pokémon leggendari siano tra i più forti, le variabili che influenzano il risultato non sono le stesse. Questo potrebbe essere dovuto all'eccessiva semplificazione dell'attributo *Total.*

Per concludere, i risultati della ricerca indicano che i pregiudizi, almeno nel mondo Pokémon, hanno un fondamento.
Il detto l'abito non fa il monaco è ampiamente smentito dall'analisi fatta in questo report, quindi se dovete combattere contro qualche Pokémon 'grande' e 'grosso'... scappate! (a meno che voi non ne abbiate uno più 'grande' e più 'grosso').

6. BIBLIOGRAFIA

1.https://it.wikipedia.org/wiki/Pok%C3%A9mon
2.https://it.wikipedia.org/wiki/Pok%C3%A9mon_Go
3.https://wiki.pokemoncentral.it/Pok%C3%A9mon_leggendari
4.https://www.kaggle.com/alopez247/pokemon
5.Classification Algorithms [Category] — NodePit
6.SMOTE - Node