Documentazione Progetto

Corso: Data Technology e Machine Learning

Membri del Gruppo:

- Marco Belotti (793675)
- Francesco Bombarda (794976)
- Antonio Vivace (793509)

Dominio applicativo:

L'ambito entro cui il progetto si colloca riguarda una delle feature fondamentali e trainanti dei videogiochi della serie di successo Pokémon, iniziata con il primo titolo nel 1996. Nello specifico si cercherà di fornire una descrizione delle meccaniche che governano queste Battles tra Pokémon, il tutto ovviamente tenendo in considerazione le peculiarità che li descrivono, alcune delle quali sono riportate di seguito:

- **PS (HP)**. Indica l'energia vitale di un Pokémon. Più alto è il valore di questa statistica, maggiori saranno i danni che il Pokémon potrà ricevere. Se i PS scendono a zero il Pokémon ha perso la battaglia.
- **Attacco**. Statistica da cui dipende l'entità dei danni che il Pokémon può provocare con attacchi fisici.
- **Difesa**. Indica la resistenza di un Pokémon agli attacchi fisici. Maggiore è il valore, minori saranno i danni ricevuti da questo tipo di attacchi.
- **Attacco Speciale**. Statistica da cui dipende l'entità dei danni che il Pokémon può provocare con attacchi speciali.
- **Difesa Speciale**. Resistenza di un Pokémon agli attacchi speciali. Maggiore è il valore, minori saranno i danni ricevuti da questo tipo di attacchi.
- **Velocità**. Indica la rapidità di un Pokémon. Un Pokémon veloce ha maggiori probabilità di avere il primo colpo in un turno di combattimento.

Obbiettivo del progetto:

Il progetto in esame si propone d'individuare un modello di Machine Learning in grado di classificare correttamente il risultato di uno scontro tra Pokemon, il tutto garantendo livelli di accuratezza e performance il più elevati possibili, ricavando le informazioni necessarie per la scelta, da vari dataset disponibili in rete, i quali saranno descritti esaustivamente nei paragrafi che seguiranno.

Raccolta dei dati:

Dopo aver scelto il dominio applicativo di riferimento e gli obbiettivi principali dello stesso, si è reso necessario individuare in rete alcuni dataset contenenti informazioni utili a soddisfare l'analisi richiesta. Queste informazioni sono state ricavate dalla piattaforma Kaggle, la quale ci ha messo a disposizione i seguenti dataset:

- **pokemon.csv**. Contiene l'elenco di tutti i Pokémon e relative statistiche.
- **combats.csv**. Contiene i combattimenti tra vari Pokémon di cui conosciamo l'esito (Utile per il training e il testing del modello).
- **pokemonTypeComp.csv**. Contiene le relazioni di forza e debolezza tra Pokémon sulla base del tipo di appartenenza.
- **test.csv**. Contiene altri combattimenti tra Pokémon di cui non abbiamo informazioni circa l'esito (Utile per mostrare un esempio di esecuzione/funzionamento in contesti reali su dati non ancora classificati).

Descrizione dettagliata dataset:

pokemon.csv		
Attributo	**Tipo di dato**	**Descrizione**
id	int	Identificativo del Pokémon
Name	chr	Nome del Pokémon
Type.1	chr	Primo tipo del Pokémon tra i 18 possibili: Normal, Fire, Water, Electric, Grass, Ice, Fighting, Poison, Ground, Flying, Psychic, Bug, Rock, Ghost, Dragon, Dark, Steel, Fairy
Type.2	chr	Secondo tipo del Pokémon tra i 18 possibili: Normal, Fire, Water, Electric, Grass, Ice, Fighting, Poison, Ground, Flying, Psychic, Bug, Rock, Ghost, Dragon, Dark, Steel, Fairy
HP	int	Punti vita
Attack	int	Valore di attacco
Defense	int	Valore di difesa
Sp.Atk	int	Valore degli attacchi speciali
Sp.Def	int	Valore delle difese speciali
Speed	int	Valore della velocità
Generation	int	Generazione a cui appartiene il Pokémon che può andare dalla prima all'ottava
Legendary	chr	Valore boolean che indica se il Pokémon è leggendario

combats.csv		
Attributo	**Tipo di dato**	**Descrizione**
First_pokemon	int	Id del primo Pokémon
Second_pokemon	Int	Id del secondo Pokémon
Winner	Int	Id del Pokémon vincitore

pokemonTypeComp.csv		
Attributo	**Tipo di dato**	**Descrizione**
Attacking	char	Tipo del pokemon che attacca
Normal	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Normal”. Il dominio dei valori è 2, 1, 0.5 e 0
Fire	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Fire”. Il dominio dei valori è 2, 1, 0.5 e 0
Water	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Water”. Il dominio dei valori è 2, 1, 0.5 e 0
Electric	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Electric”. Il dominio dei valori è 2, 1, 0.5 e 0
Grass	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Grass”. Il dominio dei valori è 2, 1, 0.5 e 0
Ice	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Ice”. Il dominio dei valori è 2, 1, 0.5 e 0
Fighting	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Fighting”. Il dominio dei valori è 2, 1, 0.5 e 0
Poison	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Poison”. Il dominio dei valori è 2, 1, 0.5 e 0
Ground	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Ground”. Il dominio dei valori è 2, 1, 0.5 e 0
Flying	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Flying”. Il dominio dei valori è 2, 1, 0.5 e 0
Psychic	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Psychic”. Il dominio dei valori è 2, 1, 0.5 e 0
Bug	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Bug”. Il dominio dei valori è 2, 1, 0.5 e 0
Rock	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Rock”. Il dominio dei valori è 2, 1, 0.5 e 0
Ghost	Int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Ghost”. Il dominio dei valori è 2, 1, 0.5 e 0
Dragon	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Dragon”. Il dominio dei valori è 2, 1, 0.5 e 0
Dark	Int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Dark”. Il dominio dei valori è 2, 1, 0.5 e 0
Steel	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Steel”. Il dominio dei valori è 2, 1, 0.5 e 0
Fairy	int	Livello di efficienza di un “tipo” di pokemon rispetto ad un altro di tipo “Fairy”. Il dominio dei valori è 2, 1, 0.5 e 0

Test.csv		
Attributo	**Tipo di dato**	**Descrizione**
First_pokemon	int	Id del primo Pokémon
Second_pokemon	Int	Id del secondo Pokémon

Dimensioni di qualità dei dataset:

Al fine di effettuare una prima analisi sui dati raccolti, si è deciso di andare a monitorare quali fossero le misure di qualità relative a completezza ed unicità sui vari dataset da integrare. In particolare attraverso la completezza desideriamo misurare in quale percentuale uno specifico attributo viene valorizzato, fornendo informazioni utili circa la presenza di valori nulli o mancanti. Attraverso la stima dell'unicità invece si vuole andare a verificare la presenza di valori duplicati su specifici attributi contenuti nei dataset, al fine di poter individuare eventuali inconsistenze nei dati.

Di seguito sono riportate le misure di qualità rilevate per ogni dataset utilizzato:

Completezza pokemon.csv

id	Name	Type.1	Type.2	HP	Attack	Defense	Sp.Atk	Sp.Def	Speed	Generation	Legendary
0.00000	0.00125	0.00000	0.48250	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Da questi valori si possono evidenziare due fattori importanti. Il primo è che all'interno del dataset lo 0.125% dei Pokémon non possiede un nome, infatti effettuando ulteriori approfondimenti è stato possibile risalire all'identificativo del Pokémon privo del nome, riportato nella figura sottostante:

```
> na_index <-sapply(pokemon, function(y) which(y==""))
> na_index
$id
integer(0)

$Name
[1] 63
```

Il secondo aspetto da sottolineare riguarda invece la mancanza di oltre il 48% dei valori associati al secondo tipo dei Pokémon, questa importante mancanza quindi ci ha spinti a non tenere in considerazione tale attributo nelle successive fasi di definizione e training del modello di classificazione, un tentativo di correzione sarebbe risultato eccessivamente costoso oltre che infattibile vista la mancanza di un'ulteriore fonte dati.

Unicità pokemon.csv

id	Name	Type.1	Type.2	HP	Attack	Defense	Sp.Atk	Sp.Def	Speed	Generation	Legendary
1.00000	1.00000	0.02250	0.02375	0.11750	0.13875	0.12875	0.13125	0.11500	0.13500	0.00750	0.00250

Attraverso questa analisi abbiamo potuto accertare l'unicità sugli identificativi oltre che sui nomi dei Pokémon (l'unicità sui restanti attributi risulta poco rilevante).

Completezza combats.csv

First_pokemon	Second_pokemon	Winner
0	0	0

In questo caso il dataset combats.csv non presenta alcun valore nullo per nessuno degli attributi in esso contenuti, il che ci consentirà di utilizzare tale dataset per il training del modello senza alcun tipo di problematica.

Unicità combats.csv

First_pokemon	Second_pokemon	Winner
0.01568	0.01568	0.01566

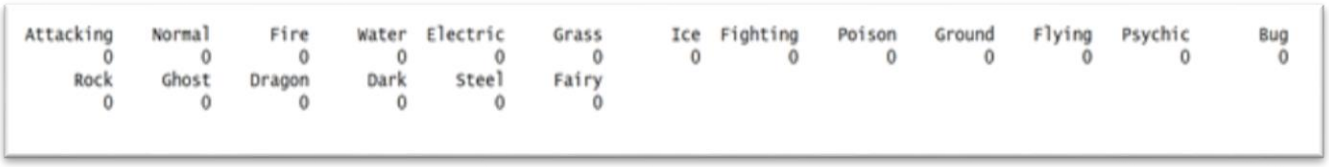
In questo caso il valore di unicità per ogni attributo non è particolarmente utile perché ci informa riguardo al fatto che alcuni Pokémon effettuano più di una battle, ma essendo il numero di record del dataset combats.csv pari a 50.000 mentre quello dei pokemon.csv pari a 800 è naturale che si abbiamo delle ripetizioni sull'utilizzo dei combattenti.

Molto più rivelante è invece considerare l'unicità tra le coppie di combattenti in modo da verificare che il dataset non consideri più volte uno stesso combattimento, magari avente addirittura esito differente.

```
> # Unicità percentuale tra le coppie di combattenti
> unique_versus <- sum(length(unique(combats$First_pokemon, combats$Second_pokemon)))
> Uniqueness_versus <- unique_versus/length(combats$First_pokemon)
> Uniqueness_versus
[1] 1
```

Da quanto riportato dall'immagine si può notare che il valore di unicità per le coppie di combattenti è pari ad 1, quindi all'interno del dataset tutte le coppie vengono considerate una ed una sola volta, scongiurando quindi ogni possibile problematica di inconsistenza sui dati.

Completezza pokemonTypeComp.csv

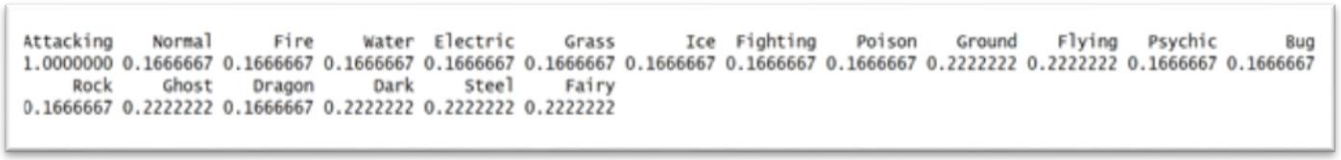


Attacking	Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug
0	0	0	0	0	0	0	0	0	0	0	0	0

Rock	Ghost	Dragon	Dark	Steel	Fairy
0	0	0	0	0	0

Anche in questo dataset non vi è la presenza di valori nulli per nessuno degli attributi in esso contenuti, potranno quindi essere utilizzati in fase di training e testing del modello di Machine Learning.

Unicità pokemonTypeComp.csv



Attacking	Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug
1.0000000	0.1666667	0.1666667	0.1666667	0.1666667	0.1666667	0.1666667	0.1666667	0.1666667	0.2222222	0.2222222	0.1666667	0.1666667

Rock	Ghost	Dragon	Dark	Steel	Fairy
0.1666667	0.2222222	0.1666667	0.2222222	0.2222222	0.2222222

Come naturale l'unico attributo contenente valori univoci è l'attributo Attacking mentre per gli altri attributi l'analisi effettuata ha uno scarso contenuto informativo.

Processo di integrazione:

Dopo aver analizzato nel dettaglio ogni collezione, il lavoro è proseguito con l'integrazione di tutte le feature all'interno di un singolo dataset più facilmente manipolabile per la successive fasi, in cui si procederà con la creazione del modello di Machine Learning. Verranno quindi integrati i dataset **pokemon.csv**, **combats.csv** e **pokemonTypeComp.csv**. In questo modo sarà possibile mettere in relazione all'interno di una stessa collezione tutti i parametri caratteristici dei Pokémon contendenti, oltre alle informazioni riguardo la tipologia e l'esito del combattimento. L'obbiettivo è infatti quello d'individuare la relazione esistente tra le vittorie nei combattimenti, le statistiche dei vari Pokémon ed il tipo del Pokémon stesso.

Integrazione pokemon.csv e combats.csv:

Il procedimento d'integrazione tra i dataset pokemon.csv e combats.csv è risultata particolarmente semplice in quanto entrambe le collezioni sono provviste del medesimo identificativo che nelle precedenti analisi abbiamo scoperto essere effettivamente univoco. L'integrazione ha coinvolto tutti gli attributi disponibili ad eccezione del secondo tipo (Type. 2 contenuto in pokemon.csv) il quale per le misure di qualità analizzate in precedenza sarebbe risultato d'intralcio per le attività di Machine Learning, sono inoltre stati calcolati alcuni parametri aggiuntivi, utili a sottolineare maggiormente gli aspetti di vantaggio/svantaggio tra i contendenti, i

quali sono stati determinati attraverso il calcolo delle differenze tra le caratteristiche peculiari dei Pokémon tra cui attacco, difesa e velocità. È stato inoltre prevista l'aggiunta dell'attributo **winner_first_label** avente dominio yes/no, utile per sapere per ogni combattimento, se il vincitore è colui che ha sferrato per primo l'attacco.

Integrazione con pokemonTypeComb.csv:

L'integrazione con il dataset pokemonTypeComb.csv ha portato ad aggiungere al file integrato un ulteriore parametro, utile per le analisi successive, cioè l'attributo advantage che possiamo ricavare dal dataset "pokemonTypeComp.csv" considerando le tipologie dei due Pokémon contendenti.

Il dataset integrato definitivo è stato poi esportato in formato CSV e nominato "integrated", reperibile all'interno del Workspace del progetto.

Dimensioni di qualità dataset integrato:

Terminato il processo d'integrazione, sono state valutate alcune misure di qualità, anche in questo caso in riferimento a completezza ed unicità sui singoli attributi

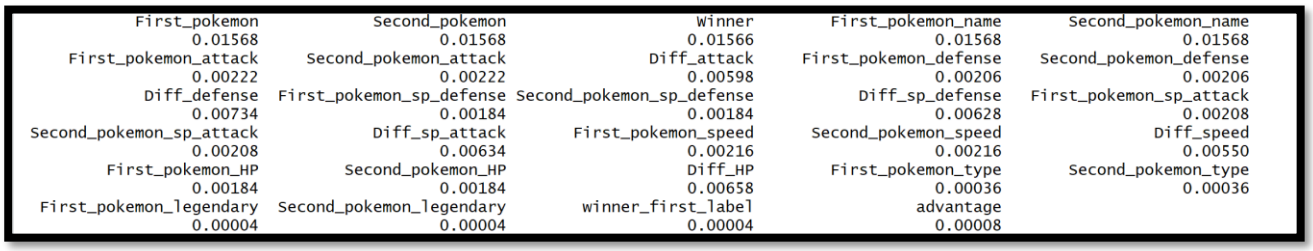
Di seguito sono mostrati i risultati ottenuti:

Completezza per il dataset integrato

First_pokemon	Second_pokemon	Winner	First_pokemon_name	Second_pokemon_name
0.00000	0.00000	0.00000	0.00112	0.00104
First_pokemon_attack	Second_pokemon_attack	Diff_attack	First_pokemon_defense	Second_pokemon_defense
0.00000	0.00000	0.00000	0.00000	0.00000
Diff_defense	First_pokemon_sp_defense	Second_pokemon_sp_defense	Diff_sp_defense	First_pokemon_sp_attack
0.00000	0.00000	0.00000	0.00000	0.00000
Second_pokemon_sp_attack	Diff_sp_attack	First_pokemon_speed	Second_pokemon_speed	Diff_speed
0.00000	0.00000	0.00000	0.00000	0.00000
First_pokemon_HP	Second_pokemon_HP	Diff_HP	First_pokemon_type	Second_pokemon_type
0.00000	0.00000	0.00000	0.00000	0.00000
First_pokemon_legendary	Second_pokemon_legendary	winner_first_label	advantage	
0.00000	0.00000	0.00000	0.00000	

Nessun attributo presenta valori nulli, ad eccezione degli attributi name, sempre per lo stesso motivo presentato in precedenza in relazione alla mancanza del campo nome per il Pokémon con ID=63, tuttavia ai fini del progetto tale problematica risulta ininfluente anche se si sarebbe potuta risolvere eliminando i record influenzati dalla mancanza del dato

Unicità per il dataset integrato



First_pokemon	Second_pokemon	Winner	First_pokemon_name	Second_pokemon_name
0.01568	0.01568	0.01566	0.01568	0.01568
First_pokemon_attack	Second_pokemon_attack	Diff_attack	First_pokemon_defense	Second_pokemon_defense
0.00222	0.00222	0.00598	0.00206	0.00206
Diff_defense	First_pokemon_sp_defense	Second_pokemon_sp_defense	Diff_sp_defense	First_pokemon_sp_attack
0.00734	0.00184	0.00184	0.00628	0.00208
Second_pokemon_sp_attack	Diff_sp_attack	First_pokemon_speed	Second_pokemon_speed	Diff_speed
0.00208	0.00634	0.00216	0.00216	0.00550
First_pokemon_HP	Second_pokemon_HP	Diff_HP	First_pokemon_type	Second_pokemon_type
0.00184	0.00184	0.00658	0.00036	0.00036
First_pokemon_legendary	Second_pokemon_legendary	winner_first_label	advantage	
0.00004	0.00004	0.00004	0.00008	

In questo caso è stata calcolata l'unicità sui singoli attributi non rivelando però particolari peculiarità

Modello di machine Learning Utilizzato:

Una volta terminata la preparazione e l'integrazione dei dati, è possibile procedere con la costruzione del modello di Machine Learning, necessario per risolvere il problema di classificazione presentato. In particolare il modello sul quale ci siamo focalizzati è basato su di un albero di decisione, i motivi che ci hanno spinti ad adottare tale metodologia sono diversi, tra questi sicuramente vi sono:

- **Semplicità.** Indubbiamente gli alberi di decisione sono facili da capire e da eseguire
- **Controllo.** L'uomo può facilmente verificare come la macchina giunge alla decisione ed eventualmente dissentire, rispetto alle reti neurali ad esempio l'albero decisionale è più facilmente comprensibile
- **Problematica in esame.** Gli alberi decisionali sono notoriamente poco adatti a modellare problemi complessi, essenzialmente perché lo spazio delle ipotesi diventa troppo grande, non è però il caso della problematica che abbiamo cercato di risolvere

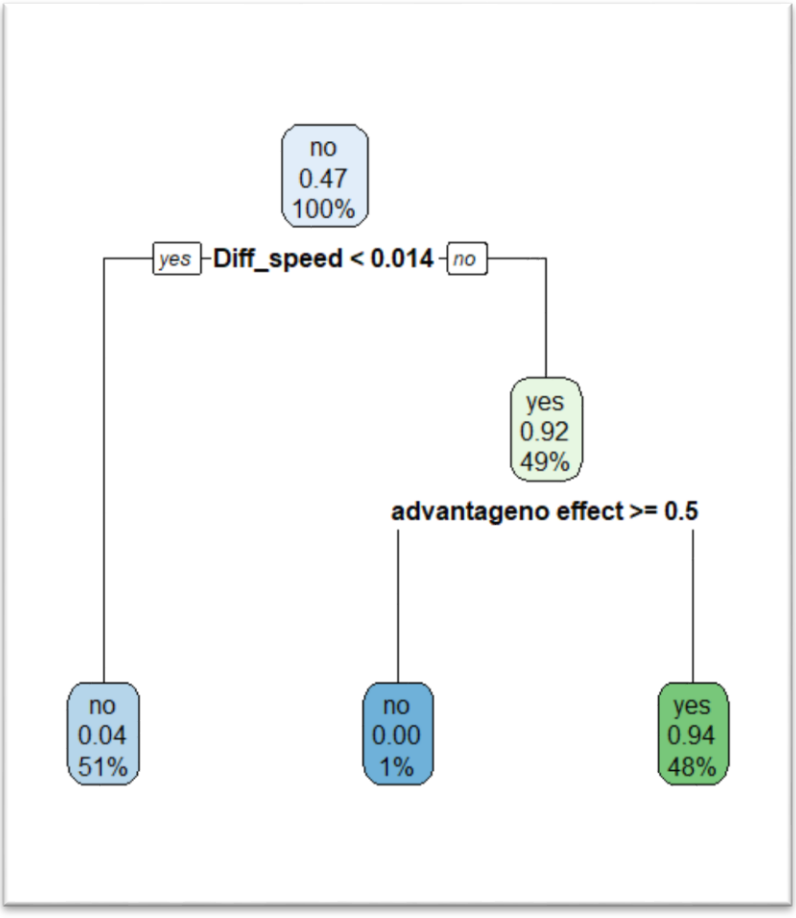
Quindi dopo aver suddiviso il dataset integrato, nelle porzioni di training e testing, come riportato nella figura sottostante

```
# Suddivido tra Training e Testing
split <- createDataPartition(y=temp$winner_first_label, p = 0.75, list = FALSE)
train <- temp[split,]
test <- temp[-split,]
```

È stato creato il modello (a partire dai dati contenuti nella porzione di training), limitatamente ad alcuni attributi, riportati di seguito:

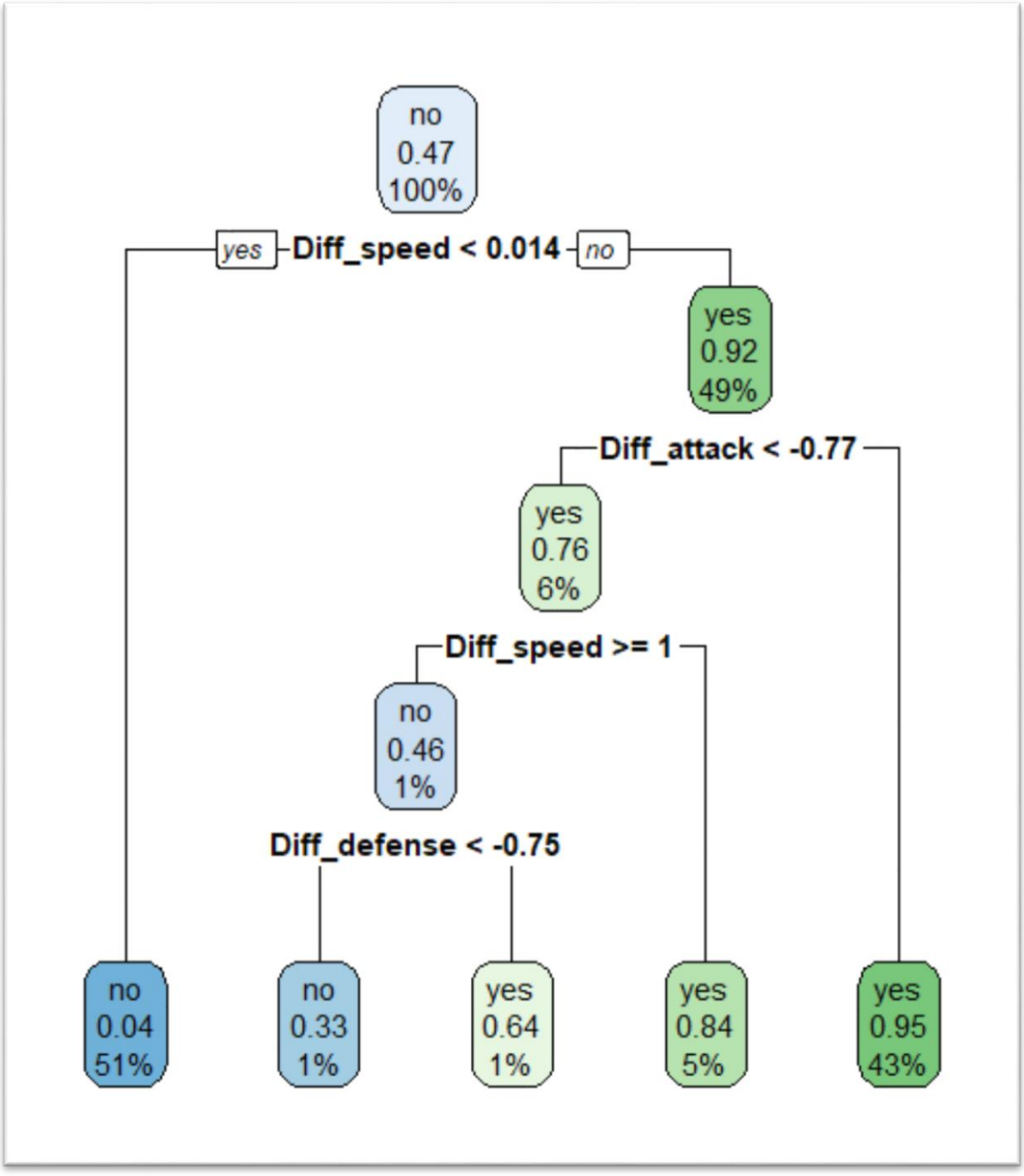
- winner_first_label
- Diff_attack
- Diff_defense
- Diff_sp_defense
- Diff_sp_attack
- Diff_speed
- Diff_HP
- First_pokemon_legendary
- Second_pokemon_legendary
- Advange (In seguito rimosso)

Il risultato emerso dopo il procedimento di training ci ha quindi fornito la seguente rappresentazione del Decision Tree

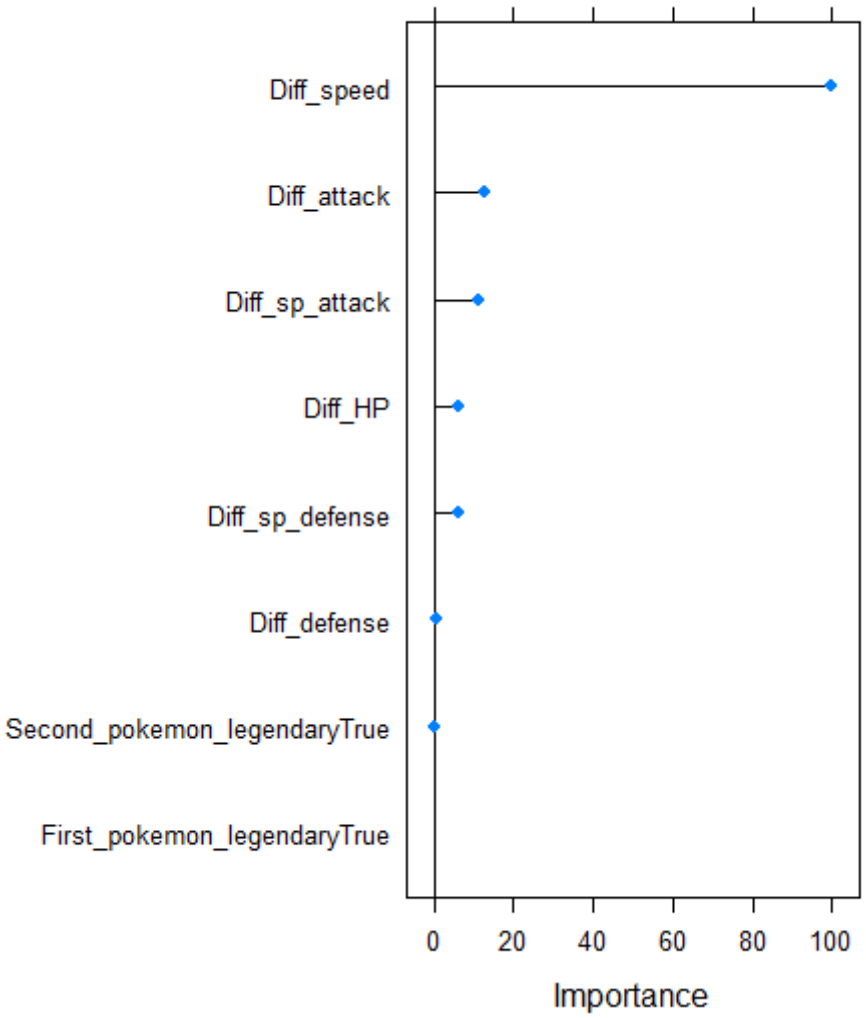


A cui è seguita la fase di testing del modello ed il calcolo delle tradizionali misure di performance, le quali però ci hanno portato a credere ad un eventuale problema di overfitting visto anche il numero estremamente limitato di parametri utilizzati per effettuare la classificazione (come si evince dal grafico riportato poco sopra).

Abbiamo cercato di aggirare questa problematica rimuovendo il parametro advantage riusciendo così ad ottenere la seguente rappresentazione del Decision Tree, il quale tiene in considerazione per effettuare la classificazione del parametro **winner_first_label** un numero più elevato di parametri rispetto al caso presentato precedentemente:



Effettuato il training del modello definitivo (privato del parametro advantage), abbiamo provveduto a determinare un ranking sull'importanza delle feature utilizzate ottenendo la seguente rappresentazione:



Risulta quindi evidente come uno dei parametri più influenti ai fini della classificazione sia la diversità di velocità tra i due Pokémon contendenti, mentre in misura decisamente minima se non addirittura nulla influisca il fatto che uno o entrambi i Pokémon siano leggendari.

Analisi dei risultati:

Dopo aver eseguito il training ed il testing del modello sono state calcolate le seguenti misure di performance:

10 fold Cross-validation Confusion Matrix:

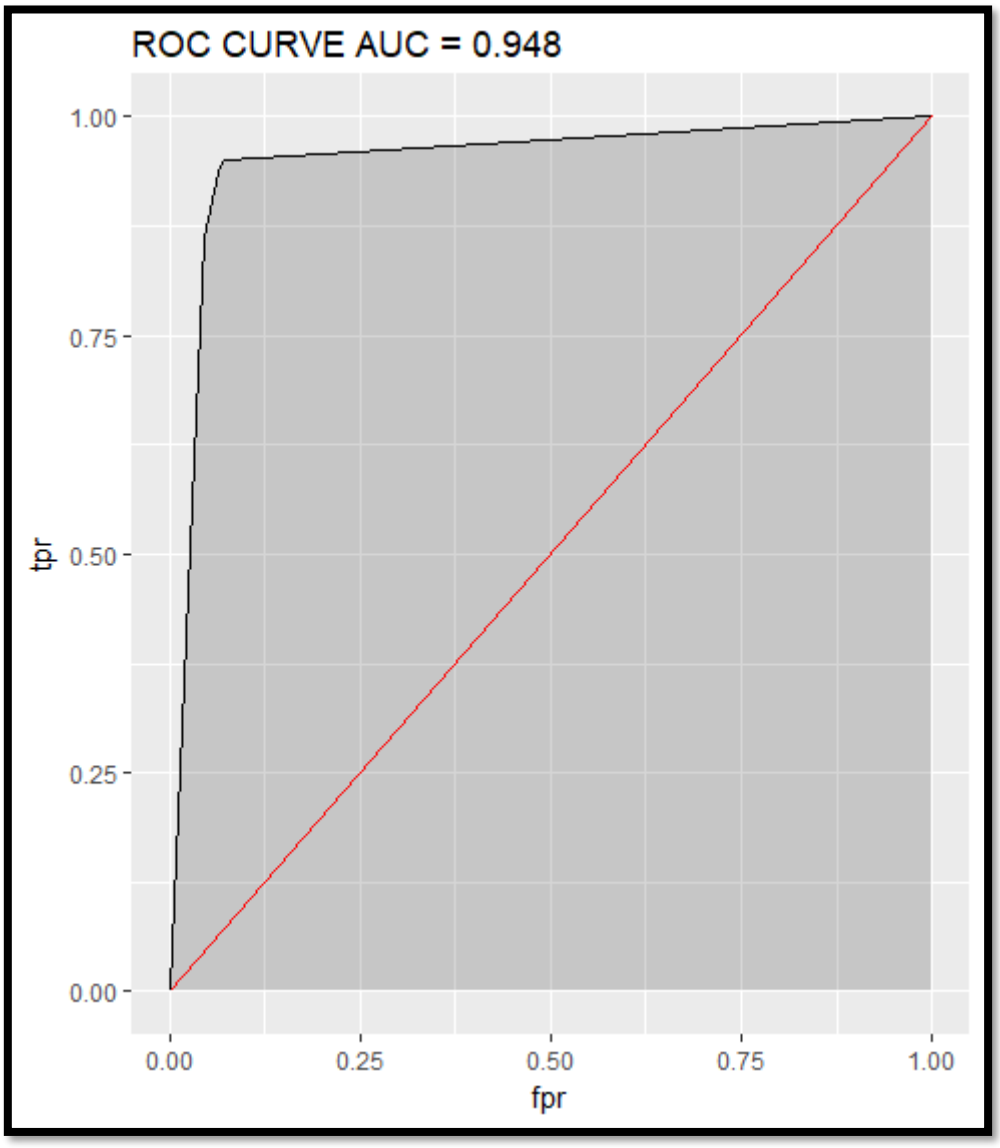
```
Cross-Validated (10 fold) Confusion Matrix

(entries are percentual average cell counts across resamples)

          Reference
Prediction   no  yes
       no  49.4  2.3
       yes  3.4 44.9

 Accuracy (average) : 0.9429

> print(precision)
[1] 0.9361616
> print(recall)
[1] 0.9547749
> print(f)
[1] 0.9453767
```

Prediction Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  6599    0
       yes    0 5900

               Accuracy : 1
                 95% CI : (0.9997, 1)
    No Information Rate : 0.528
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

            Sensitivity : 1.000
            Specificity : 1.000
         Pos Pred Value : 1.000
         Neg Pred Value : 1.000
             Prevalence : 0.528
         Detection Rate : 0.528
   Detection Prevalence : 0.528
      Balanced Accuracy : 1.000

       'Positive' Class : no

> print(precision_pred)
[1] 1
> print(recall_pred)
[1] 1
> print(f_pred)
[1] 1
```

Come si nota dai valori presenti nella Confusion Matrix abbiamo dei valori di accuratezza, precision, recall ed f-measure molto elevati, il che rende la classificazione del parametro **winner_first_label** tipicamente corretta

ROC Curve



Abbiamo inoltre una curva ROC con area under curve molto elevata e prossima ad uno, a testimonianza di quanto affermato precedentemente