Relazione progetto Data Mining
Shelter Animal Outcomes

GIACOMO MANZOLI
1130822
Università degli Studi di Padova
9 giugno 2016

Indice

List of Algorithms

1 Introduzione

1.1 Descrizione del dataset

Il dataset contiene 26729 osservazioni relative agli animali che hanno lasciato il rifugio per animali della città di Austin nel periodo che va dall'Ottobre 2013 a Marzo 2016. L'obiettivo è quello di utilizzare i dati del dataset per prevedere quale sarà il destino dei nuovi animali che verranno accolti nel centro.

I dati sono forniti da Kaggle per la competizione "*Shelter Animal Outcomes*"[1] e, trattandosi di una sfida, viene fornito anche un secondo dataset di 11456 osservazioni, per le quali non è nota la variabile risposta, che deve essere utilizzato per fornire al sito le proprie previsioni, al fine di stilare una classifica.

1.2 Descrizione delle variabili

Il dataset è composto da 10 variabili che descrivono lo stato dell'animale quando ha lasciato il rifugio. Più nel dettaglio:

- `AnimalID`: codice univoco che viene affidato all'animale quando è entrato nel rifugio. Nel dataset principale viene fornito sotto forma di stringa, mentre nel dataset secondario viene fornito come numero intero.
- `Name`: nome dell'animale. Nel dataset principale ci sono 7691 animali senza un nome.
- `DateTIme`: data e ora in cui l'animale ha lasciato il rifugio. È espressa nel formato `aaaa-mm-gg hh:mm:ss`.
- `Outcome`: variabile risposta, ha 5 possibili valori:
  - *Adoption*
  - *Died*
  - *Euthanasia*
  - *Return to owner*
  - *Transfer*
- `OutcomeSubtype`: variabile che descrive perché l'animale ha fatto quella particolare fine. Ci sono 17 possibili valori per questa variabile e per 13612 non è disponibile. Questa variabile non è presente sul dataset secondario.
- `AnimalType`: tipo dell'animale, può essere un cane o un gatto.
- `SexuponOutcome`: sesso dell'animale, comprende anche l'informazione se l'animale è stato castrato o meno. In tutto ci sono 5 possibili valori per questa variabile.
- `AgeuponOutcome`: età dell'animale quando ha lasciato il rifugio, viene espressa utilizzando una stringa che descrive l'età, ad esempio: *2 years, 1 week, ecc.*
- `Breed`: razza dell'animale. Comprende anche l'informazione se l'animale è un incrocio di più razze e in qualche caso specifica anche la seconda razza. In tutto ci sono 1380 possibili valori.
- `Color`: colore del pelo dell'animale. Comprende anche le informazioni relative al pelo e ad un eventuale colore secondario. In tutto ci sono 336 possibili valori.

La struttura del dataset una volta caricato in R è la seguente:

[1] `https://www.kaggle.com/c/shelter-animal-outcomes`

```
'data.frame': 26729 obs. of  10 variables:
$ AnimalID      : Factor w/ 26729 levels "A006100","A047759",..
$ Name          : Factor w/ 6375 levels "","," Joanie"," Mario",..
$ DateTime      : Factor w/ 22918 levels "2013-10-01 09:31:00",..
$ OutcomeType   : Factor w/ 5 levels "Adoption","Died",..
$ OutcomeSubtype: Factor w/ 17 levels "","Aggressive",..
$ AnimalType    : Factor w/ 2 levels "Cat","Dog"
$ SexuponOutcome: Factor w/ 6 levels "","Intact Female",..
$ AgeuponOutcome: Factor w/ 45 levels "","0 years","1 day",..
$ Breed         : Factor w/ 1380 levels "Abyssinian Mix",..
$ Color         : Factor w/ 366 levels "Agouti","Agouti/Brown Tabby",..
```

Inoltre, andando a tracciare il grafico con le proporzioni delle varie classi in base al tipo di animale (Figura 1) è possibile osservare che per i cani è più probabile che siano recuperati dai propri padroni rispetto ai gatti, mentre per i gatti è più probabile che vengano trasferiti.

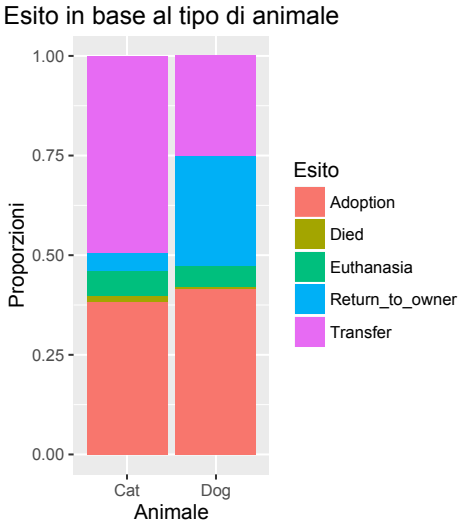

Figura 1: Esito in base al tipo di animale

1.3 Software utilizzato

Nella realizzazione del progetto è stato utilizzato l'ambiente R in versione 3.2.4, esteso con alcune librerie come `ggplot2` e `dplyr` disponibili su CRAN.

aggiornare le libreriie

2 Elaborazione delle variabili

Il dataset si presenta con poche variabili che riassumono molte informazioni o che hanno un numero molto elevato di livelli. È quindi necessario andare ad estrapolare le informazioni da queste variabili, creandone di nuove e di più semplici.

2.1 Il nome dell'animale

La variabile `Name` ha più di 6000 possibili valori distinti e, ragionando a livello di previsione, sembra poco probabile che il nome dell'animale influisca sul suo destino. Tuttavia ci sono 7691 animali che non hanno un nome, questo probabilmente implica che si tratta di animali randagi

che sono stati portati al rifugio[2] e quindi può essere meno probabile che il padrone li venga a recuperare. Analogamente se è stato trovato un'animale smarrito e con una targhetta con il nome al collo, è più probabile che il suo padrone vada a recuperarlo.

Il nome può quindi essere riassunto da una nuova variabile booleana `HasName` che specifica se l'animale ha un nome o meno.

Tracciando il grafico (Figura 2) per visualizzare la ripartizione delle varie classi per la variabile risposta in base al valore di `HasName` si ha che le osservazioni fatte sembrano essere confermate dai dati: indipendentemente dal tipo di animale, se questo ha un nome è più probabile che venga recuperato dal suo padrone.

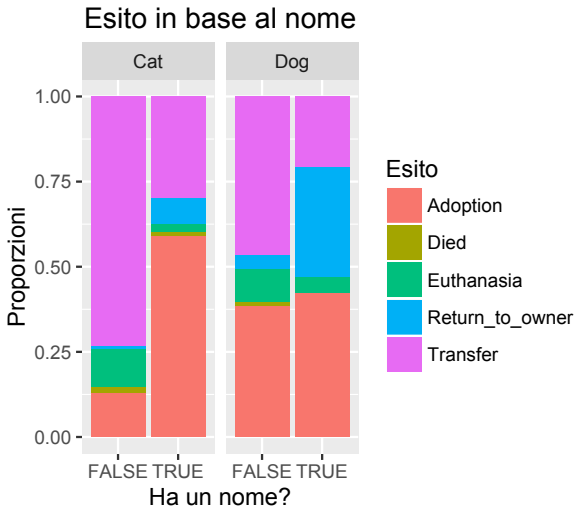

Figura 2: Esito in base al nome

2.2 La data di uscita

Difficilmente la data in cui l'animale ha lasciato la struttura può tornare utile per effettuare previsioni future. Si possono però estrarre altre informazioni come la fascia oraria e il giorno della settimana in cui l'animale ha lasciato la struttura.

Queste due informazioni possono tornare utili perché, ad esempio la gente nei weekend ha più tempo libero e quindi è più probabile che abbia tempo per andare ad adottare un cane, oppure che per motivi logistici i trasferimenti possono essere fatti solamente la mattina.

Per estrapolare ciò vengono definite due nuove variabili `DayOfWeek` che assume come valore il giorno della settimana e `TimeOfDay` che può assumere come valori:

- *Mattina*: se l'ora è compresa tra le 6 e le 12.
- *Pomeriggio*: se l'ora è compresa tra le 12 e le 17.
- *Sera*: se l'ora è compresa tra le 17 e le 20.
- *Notte*: per le restanti ore.

Come si può notare dal grafico (Figura 3) c'è un picco sulle adozioni nella fascia serale. Sempre dal grafico si può notare che la notte sono più frequenti i trasferimenti, anche se la maggior parte di questi si viene svolta durante le ore diurne.

Per quanto riguarda il giorno della settimana, dal grafico Figura 4, si può notare come le adozioni siano più probabili nei week-end.

[2]Non vengono fornite informazioni a riguardo

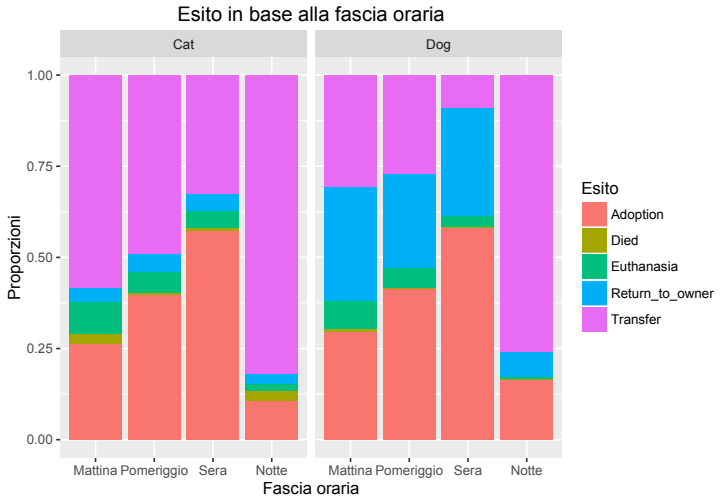

Figura 3: Esito in base alla fascia oraria

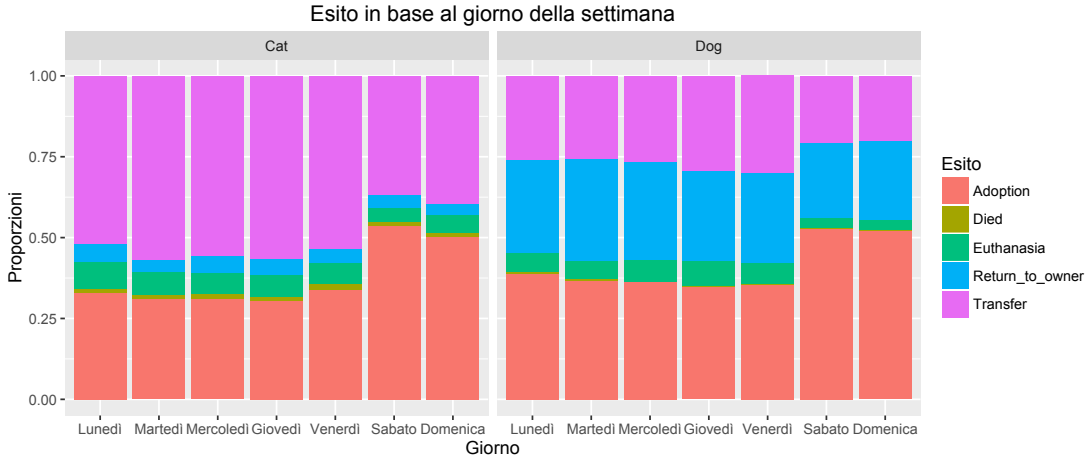

Figura 4: Esito in base al giorno della settimana

2.3 Sesso e stato dell'animale

La variabile `SexuponOutcome` prevede 6 possibili livelli che racchiudo l'informazione relativa al sesso e al fatto se l'animale è stato castrato o meno. C'è poi un livello *Unknown* per gli animali per i quali non si hanno informazioni e in più ci sono dei valori non disponibili.

Questa variabile è stata quindi scomposta in `Gender` che specifica il sesso dell'animale e `Status`, che specifica se l'animale è stato castrato o meno. Entrambe le variabili hanno un terzo possibile valore *Unknown* che rappresenta i dati non disponibili.

Dal grafico riportato in Fig 5 si può notare che indipendentemente dal fatto che indipendentemente dal sesso e dal tipo di animale, se l'animale è stato castrato è più probabile che venga adottato, se invece non è stato castrato oppure non ci sono informazioni a riguardo, è più probabile che venga trasferito.

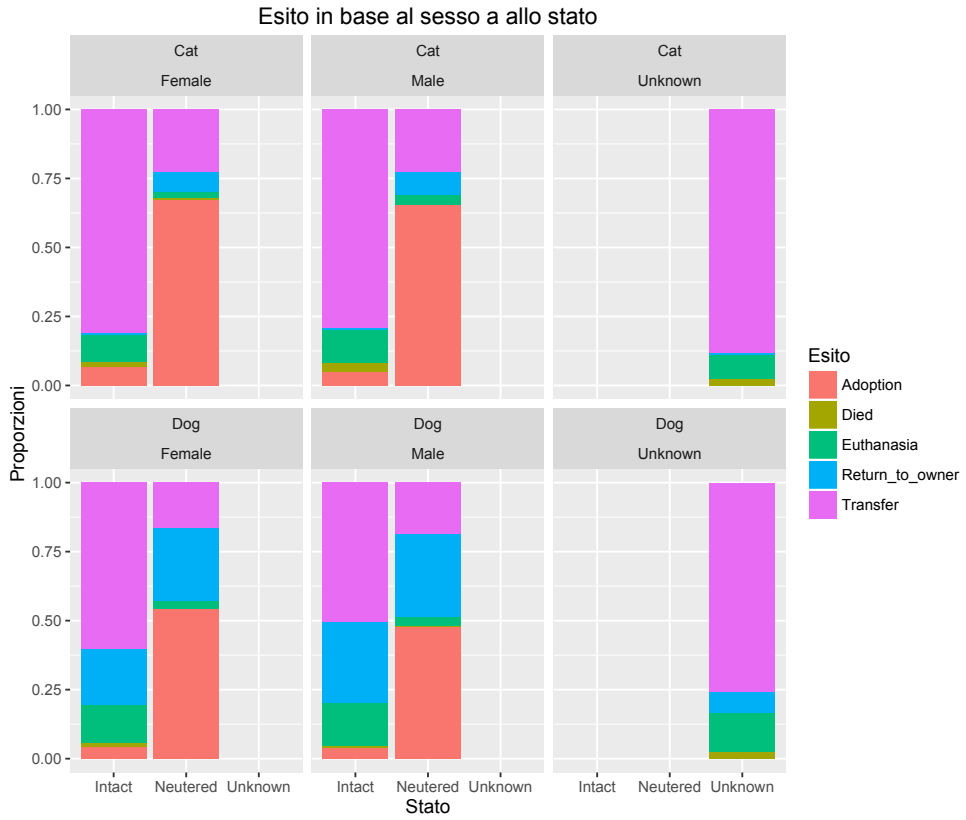

Figura 5: Esito in base al sesso e allo stato dell'animale

2.4 L'età dell'animale

L'età dell'animale è memorizzata nella variabile `AgeuponOutcome` in un formato molto confusionale in quanto viene espressa come *2 anni*, *3 mesi*, ecc. Inoltre ci sono dei casi in cui alcuni valori non hanno la *s* del plurale, ovvero tra i possibili valori della variabile ci sono ad esempio *2 week* e *2 weeks*, che quindi vengono considerati come valori distinti quando in realtà non lo sono.

La prima modifica è quindi quella di normalizzare i valori, esprimendoli con un numero intero che approssima l'età dell'animale espressa in giorni, così facendo risulta più semplice classificare gli animali per fascia d'età. Infatti, si può assumere che un cucciolo è più probabile che venga adottato rispetto ad un animale più anziano, mentre gli animali troppo piccoli non possono essere adottati per legge.

Conviene quindi creare una nuova variabile `AgeCategory` con 5 possibili livelli:

- *Neonato*: da 0 a 29 giorni.
- *Cucciolo*: da 30 a 365 giorni.
- *Adulto*: da 366 a 3650 giorni (10 anni).
- *Anziano*: più di dieci anni.
- Sconosciuta.

Nella normalizzazione dei valori si è scelto di mantenere le 18 osservazioni con i valori mancati per l'età, marcandoli come sconosciuti, questo perché nel secondo dataset sono presenti delle osservazioni per le quali l'eta non è nota. Inoltre si può ipotizzare che questi animali siano randagi e quindi che per questo motivo la loro eta non è nota. Questa ipotesi deriva dal fatto che tra i possibili valori della variabile `OutcomeSubtype` c'è il valore *SCRP* che indica un trasferimento relativo al programma di recupero dei gatti randagi[3] e quasi tutte le osservazioni con l'età mancante hanno proprio quel valore come `OutcomeSubtype`.

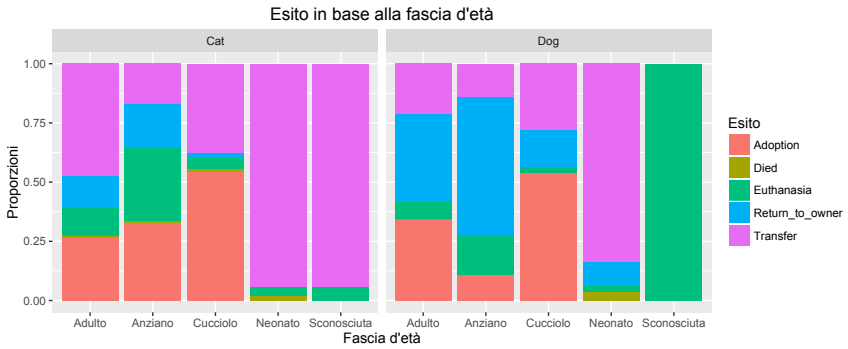

Figura 6: Esito in base alla fascia d'età dell'animale

Come si può notare dal grafico Figura 6, nessuno degli animali neonati viene adottato ed è più probabile che vengano trasferiti. Per quanto riguarda la probabilità di adozione, questa è maggiore per i cuccioli e più bassa per i cani anziani. I gatti anziani hanno una maggiore probabilità di adozione rispetto ai cani anziani e questo può essere dovuto al fatto che la speranza di vita di un gatto è maggiore rispetto a quella di un cane.

2.5 Razza

Le informazioni relative alla razza dell'animale sono racchiuse nella variabile `Breed`, la quale ha 1340 possibili valori e specifica anche se l'animale è un incrocio o meno.

Osservando alcuni dei possibili valori, si può notare che se l'animale non è di razza,, il valore della variabile comprende o due razze oppure la razza principale, seguita da *Mix*. Si è scelto quindi di scomporre la variabile `Breed` nelle variabili `PrimaryBreed` (220 livelli), `SecondaryBreed` (144 livelli) e `IsMix` (booleana).

Ci sarebbero ulteriori informazioni che possono essere estratte da questa variabile, come la stazza dell'animale, la quale a sua volta va ad influire sull'aspettativa di vita e quindi sulla corretta classificazione della fascia d'età e sul carattere dell'animale. Tuttavia per estrarre queste informazioni in modo corretto è necessaria un'elevata conoscenza del dominio, ci si è quindi limitati alla scomposizione della variabile `Breed`.

Come si può notare dal grafico Fig 7, un cane di razza ha più probabilità di essere adottato rispetto ad un cane non di razza, mentre per i gatti sembra che avvenga il contrario. Non sono disponibili i grafici per le variabili `PrimaryBreed` e `SecondaryBreed` perché il numero elevato di possibili valori renderebbe i grafici incomprensibili.

[3] `http://www.maddiesfund.org/austin-animal-services-stray-cat-return-program.htm`

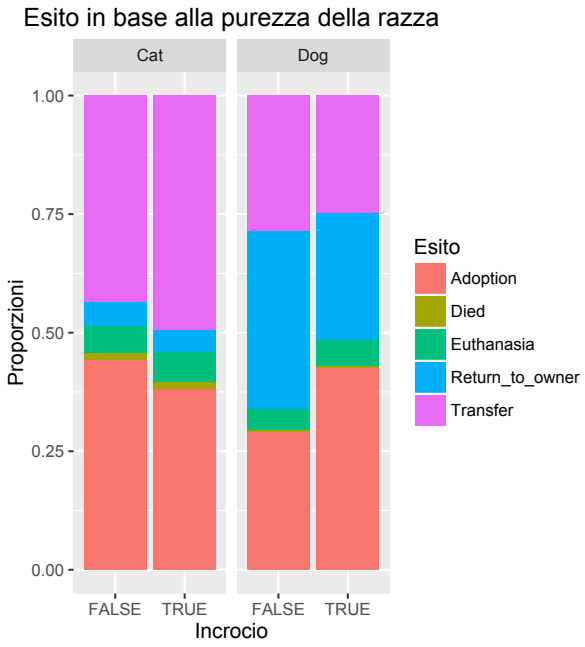

Figura 7: Esito in base alla fascia d'età dell'animale

2.6 Colore

Come per la razza, anche per il colore le informazioni sono racchiuse nell'unica variabile `Color` e, sempre come per la razza, queste informazioni sono state suddivise nelle variabili:

- `PrimaryColor`: colore principale, 29 livelli.
- `SecondaryColor`: colore secondario, 24 livelli.
- `Pattern`: pattern del pelo, 10 livelli.
- `HasComplexColor`: valore booleano che specifica se il pelo dell'animale ha più colori o un certo pattern particolare.

Sarebbe poi necessario andare a normalizzare i valori dei colori, dato che ci sono più livello che indicano lo stesso colore, come *Orange* e *Apricot* (albicocca), ma da una prima analisi grafica (Fig **??**) sembra che fissato il tipo di animale, le informazioni sul pelo non influiscano sull'esito e in quei pochi casi che questo succede può essere dovuto al fatto che si hanno troppe poche osservazioni con quel determinato colore.

2.7 Riassunto delle trasformazioni

Dopo aver applicato le trasformazioni precedentemente descritte ed aver eliminato le variabili `AnimalID` e `OutcomeSubtype` il dataset ha assunto la seguente struttura:

```
'data.frame': 26711 obs. of  15 variables:
$ OutcomeType    : Factor w/ 5 levels "Adoption","Died",..
$ AnimalType     : Factor w/ 2 levels "Cat","Dog"
$ AgeCategory    : Factor w/ 4 levels "Adulto","Anziano",..
$ DayOfWeek      : Factor w/ 7 levels "Lunedì","Martedì",..
$ TimeOfDay      : Factor w/ 4 levels "Mattina","Pomeriggio",..
$ Gender         : Factor w/ 3 levels "Female","Male",..
$ Status         : Factor w/ 3 levels "Intact","Neutered",..
```

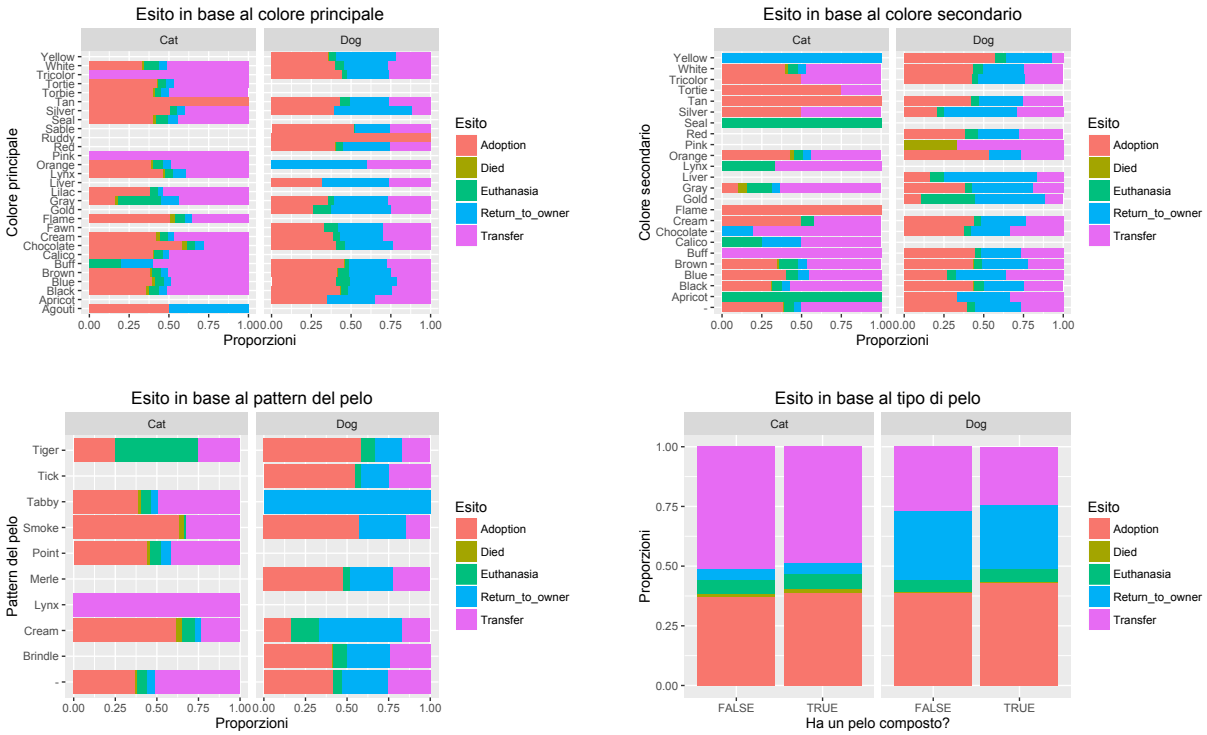

Figura 8: I grafici della prima riga rappresentano rappresentano l'esito in base al colore principale e secondario. I grafici della seconda riga rappresentano l'esito in base al pattern e al fatto se il pelo dell'animale è composto o meno. Salvo alcuni casi dovuti al fatto che sono presenti poche osservazioni con quel determinato valore per una delle variabili, non si nota una correlazione tra il pelo e l'esito dell'animale una volta stabilito se si tratta di un cane o un gatto.

```
$ PrimaryColor   : Factor w/ 29 levels "Agouti","Apricot",..
$ SecondaryColor : Factor w/ 24 levels "-","Apricot",..
$ Pattern        : Factor w/ 10 levels "-","Brindle",..
$ HasComplexColor: Factor w/ 2 levels "FALSE","TRUE"
$ PrimaryBreed   : Factor w/ 220 levels "Abyssinian","Affenpinscher",..
$ SecondaryBreed : Factor w/ 144 levels "-","Affenpinscher",..
$ IsMix          : Factor w/ 2 levels "FALSE","TRUE"
$ HasName        : Factor w/ 2 levels "FALSE","TRUE"
```

3 Modelli

Una volta sistemati i dati, sono stati provati vari modelli per vedere quale riesce ad effettuare le previsioni migliori. In particolare sono stati utilizzati:

- Regressione logistica multinomiale
- Alberi di classificazione
- MARS
- Reti Neurali
- GAM
- Random Forest
- Bagging

- Boosting

Per calcolare e confrontare i modelli, le 26000 osservazioni sono state suddivise in due insiemi, un insieme di validazione contenente il 20% delle osservazioni e che viene utilizzato per confrontare gli errori commessi dai vari modelli e un altro set di dati composto dal restante 80% delle osservazioni. Le osservazioni di quest'ultimo set sono poi state suddivise in altri due sottoinsiemi, il *test set* contenete il 20% e il *train set* contenente le restanti osservazioni.

Con questa suddivisione è possibile utilizzare il *train set* per calcolare il modello, provando più valori per gli eventuali iper-parametri, il *test set* per confrontare quale configurazione di iper-parametri funziona meglio ed infine combinare i due set per calcolare la versione del modello da confrontare con gli altri modelli sui dati del *validation set*.

Infine, una volta trovata la configurazione ottimale di ogni modello, questo viene ricalcolato anche sulla totalità delle osservazioni per poi utilizzarlo per effettuare le previsioni sul dataset secondario in modo da poterle caricare su Kaggle, ottenendo così un'ulteriore metrica per la valutazione della bontà del modello. Non è precisato su che cosa si basa il punteggio attribuito da Kaggle, tuttavia minore è il valore attribuito, migliore è il modello e per entrare nella top 100 dei punteggi migliori è necessario scendere sotto lo 0.7236.

3.1 Regressione logistica multiclasse

Il primo modello utilizzato è stato quello che effettua la regressione logistica multiclasse. Tra le varie implementazioni della regressione logistica disponibili per R si è scelto di utilizzare `multinom` presente nel pacchetto `nnet` e che simula la regressione logistica utilizzando una rete neurale. È stata scelta questa particolare implementazione perché la funzione `glm` utilizzata in laboratorio funziona solo per la regressione binomiale e la funzione `mlogit` dell'omonimo pacchetto si è rilevata eccessivamente complessa.

Il modello calcolato da `multinom` prevede, tra i vari iper-parametri, il valore di `decay` e confrontando l'effetto dei vari valori sul test set, si è trovato come valore migliore 0.001, il quale è stato utilizzato per calcolare il modello di regressione sulla versione combinata del test e train set, producendo un errore sul test di classificazione pari a 0.3365762, ovvero circa del 33.66%.

Gli errori di classificazione sono riportati nella Tabella 2, dalla quale si può notare che la maggior parte degli errori commessi riguardano i trasferimenti che vengono classificati erroneamente come adozione. Un'altra cosa che si può notare è che nessuna degli animali viene classificato come deceduto per morte naturale e questo può essere causato dal fatto che nel dataset ci sono poche osservazioni che descrivono animali morti per cause naturali.

		Valori osservati				
		Adoption	Died	Euthanasia	Return to owner	Transfer
Valori predetti	Adoption	1827	1	38	364	462
	Died	0	0	0	0	0
	Euthanasia	3	1	40	17	21
	Return to owner	242	4	71	446	142
	Transfer	120	36	173	104	1233

Tabella 1: Errori di classificazione con il modello di regressione logistica.

Per quanto riguardo il punteggio ottenuto su Kaggle, il modello calcolato su tutto il dataset principale ha ottenuto un punteggio di 0.89358.

3.2 Alberi di classifcazione

Per calcolare l'albero di classificazione è stato utilizzato il modello disponibile nel pacchetto `tree`.

L'albero è stato costruito utilizzando le osservazioni presenti nel train set, espandendolo fino ad ottenere una devianza interna alle foglie minore di 0.002 (`mindev=0.002`). Dopodiché sono stati usati i dati del test set per potare l'albero in modo da limitare l'overfitting, ottenendo come albero migliore quello con 21 foglie.

L'albero così ottenuto ha ottenuto un errore di classificazione pari a 0.3543499, ovvero leggermente peggiore rispetto alla regressione logistica,

La Tabella **??** riporta gli errori di classificazione commessi dal modello, per i quali valgono le stesse considerazioni fatte per la regressione logistica, alle quali si aggiunge il fatto che anche la classe *Euthanasia* non viene mai assegnata.

C'è però da tenere in considerazione che questo modello è stato calcolato senza utilizzare le variabili `PrimaryBreed` e `SecondaryBreed` perché con la funzione `tree` non è possibile usare variabili di tipo `Factor` con più di 32 livelli ed entrambe le variabili sforavano questo limite. Nonostante ciò il punteggio ottenuto su Kaggle è di 0.87657, ovvero migliore rispetto a quello ottenuto dalla regressione logistica.

		Valori osservati				
		Adoption	Died	Euthanasia	Return to owner	Transfer
Valori predetti	Adoption	1765	2	33	352	431
	Died	0	0	0	0	0
	Euthanasia	0	0	0	0	0
	Return to owner	289	3	86	465	206
	Transfer	138	37	203	114	1221

Tabella 2: Errori di classificazione con l'albero di classificazione.