Progetto di Data Technology e Machine Learning

January 2019

Gruppo:

Fabbris	Elia	793240
Fumagalli	Matteo	793670

1 Introduzione

1.1 Descrizione del dominio di riferimento e obiettivi dell'elaborato

Abbiamo scelto un contesto automobilistico per il nostro progetto, nel dettaglio abbiamo deciso di trattare gli annunci presenti online di auto usate.
L'obiettivo del nostro elaborato è stato quello di addestrare un algoritmo di Machine Learning, in particolare k-Means, che presi come attributi le caratteristiche di auto fosse in grado di suddividerle in diverse categorie (crossover, hatch,ecc.)

1.2 Scelte di design per la creazione del dataset

Ottenuti i dataset, abbiamo scelto le feature da utilizzare riducendone il numero rispetto a quelle ottenute dall'integrazione. Abbiamo inserito un ulteriore attributo nel dataset "Car sales", reputato necessario ai fini del nostro elaborato.
Sono stati presi in considerazione solamente 4 tipi di auto su i 6 disponibili.
Il tipo di auto "Van", avendo solamente 10 record nel dataset integrato, non è stato preso in considerazione.
L'altro tipo di auto non preso in considerazione è stato "Other": dall'analisi del dataset si evince che questo tipo di auto sia stato usato come valore di default, rendendo così l'insieme dei record categorizzati "Other" poco omogeneo. Avendo all'interno dati che sarebbero dovuti essere categorizzati in altri tipi, il suo utilizzo avrebbe peggiorato notevolmente le performance dell'algoritmo.

2 Data Managment

2.1 Sorgenti dati e scelta del modello di descrizione del dataset

I dati scaricati sono i seguenti:

1 - Car Sale Advertisements

- Descizione: raccolta di dati proveniente da annunci di vendita di auto tra privati in Ucraina;
- Numero di record: 9576;
- Link al dataset.

2 - Car sales

- Descizione: raccolta di macchine contenente dati riguardanti gli aspetti tecnici;
- Numero di record: 156;
- Link al dataset.

Per migliorare le performance è stato necessario aggiungere, tramite dati ottenuti da Wikipedia, un attributo "Height" al dataset "Car sales". In questo modo è stato possibile valutare un'ulteriore caratteristica, fondamentale per il nostro scopo.

Il modello selezionato per il dataset è di tipo tabellare, nello specifico il formato CSV. Qui di seguito le ragioni per cui è stata effettuata questa scelta:

- I dataset di partenza erano forniti in questo formato;
- Il parsing dei dati in questo formato risulta particolarmente agevole;
- Compatibilità con i vari tool utilizzati (Excel, Pentaho Data Integration, Talend Data Preparation);
- Infine abbiamo ritenuto inutile complicare la rappresentazione, considerando che i dati in questo formato risultavano già modellati sufficientemente bene al fine di utilizzarli.

2.2 Dimensioni di qualità relative ai datasets pre-integrazione

Le dimensioni di qualità prese in considerazione per la selezione dei dataset da utilizzare come sorgenti dati sono:

Completezza: ossia la copertura con la quale il fenomeno osservato è rappresentato nel dataset. Nello specifico ci siamo concentrati sulla completezza degli attributi (Attribute Completeness). Abbiamo deciso di reputare accettabile un valore di completezza $\geq 95\%$ per gli attributi di nostro interesse.

Car Sale Advertisements

Attributo	*NON null*	*Completezza (%)*	*Interesse*
car	9576	100	no
price	9576	100	sì
body	9576	100	sì
mileage	9576	100	no
engV	9576	100	no
engType	9576	100	no
registration	9576	100	no
year	9576	100	no
Model	9576	100	sì
drive	9065	95	no

Car sales

Attributo	*NON null*	*Completezza (%)*	*Interesse*
Manufacturer	156	100	sì
Model	156	100	no
Vehicle type	156	100	no
Price	156	100	no
Engine size	156	100	no
Horsepower	156	100	no
Wheelbase	156	100	sì
Width	156	100	sì
Length	156	100	sì
Height	156	100	sì
Fuel capacity	156	100	no
Fuel efficiency	156	100	no

Attualità: dimensione di qualità che misura con quale rapidità i dati sono aggiornati, rispetto al corrispondente fenomeno del mondo reale. Nello specifico abbiamo posto come vincolo la presenza nel dataset di automobili appartenenti al periodo 1978-2018. Il dataset "Car Sale Advertisements" comprende automobili dal 1968 al 2016. Di conseguenza possiamo accettare il risultato ottenuto anche se non viene rispettata completamente la dimensione di qualità: non essendo presenti gli anni 2017 e 2018, abbiamo una copertura del 95% (38 anni su 40).

2.3 Integrazione dei dati

L'integrazione dei dati è stata realizzata utilizzando il tool *Pentaho*. Tale processo è stato eseguito sfruttando l'eterogeneità dell'attributo "Model", il quale è contenuto in entrambi i dataset iniziali.
Il processo di integrazione è stato eseguito eseguendo l'operatore di join "Inner2 tenendo conto appunto dell'attributo "Model" in *Car sales* e *Car Sale Advertisements*.
Il file di output generato è composto dai seguenti attributi:

- "Model", "Price" e "Body" provenienti dal dataset *Car Sale Advertisements*;
- "Manufacturer", "Wheelbase", "Width", "Length" e "Height" provenienti dal dataset *Car sales*.

Va reso noto che gli attributi "Model" e "Manufacturer" sono stati mantenuti nel file di output, nonostante non siano stati necessari per conseguire il nostro obiettivo, in modo da rendere il dataset più chiaro e leggibile per l'uso che ne abbiamo fatto.

2.4 Dimensioni di qualità relative al dataset post-integrazione

Le dimensioni di qualità prese in considerazione per il dataset ottenuto tramite il processo di integrazione sono:

Completezza dataset finale.

Attributo	*NON null*	*Completezza (%)*
Manufacturer	1420	100
Model	1420	100
Price	1394	98.2
Wheelbase	1420	100
Width	1420	100
Length	1420	100
Height	1420	100
body	1420	100

Accuratezza: per la seguente misura di qualità abbiamo deciso di prendere in considerazione l'accuratezza sintattica per l'attributo "body". Attraverso uno script abbiamo definito i valori sintatticamente corretti, dopodichè li abbiamo confrontati con i valori contenuti nel dataset integrato.
Il risultato ottenuto da tale analisi ha fornito un'accuratezza sintattica del 100%.

2.5 Analisi descrittive dei dati integrati

Attributo	**Descrizione**	**Tipo**	**Min/max**	**Valori distinti**
Manufacturer	Marca automobile	Stringa	/	24
Model	Modello automobile	Stringa	/	64
Price	Prezzo di vendita auto usata	Intero ($)	600/222222	474
Wheelbase	Distanza tra i due assi	Decimale (in)	93.4/127.2	53
Width	Larghezza automobile	Decimale (in)	65.7/79.9	43
Length	Lunghezza automobile	Decimale (in)	152/215.3	60
Height	Altezza automobile	Decimale (in)	48/80.5	45
Body	Categoria automobile	Stringa	/	6

3 Machine Learning

3.1 Creazione del training set

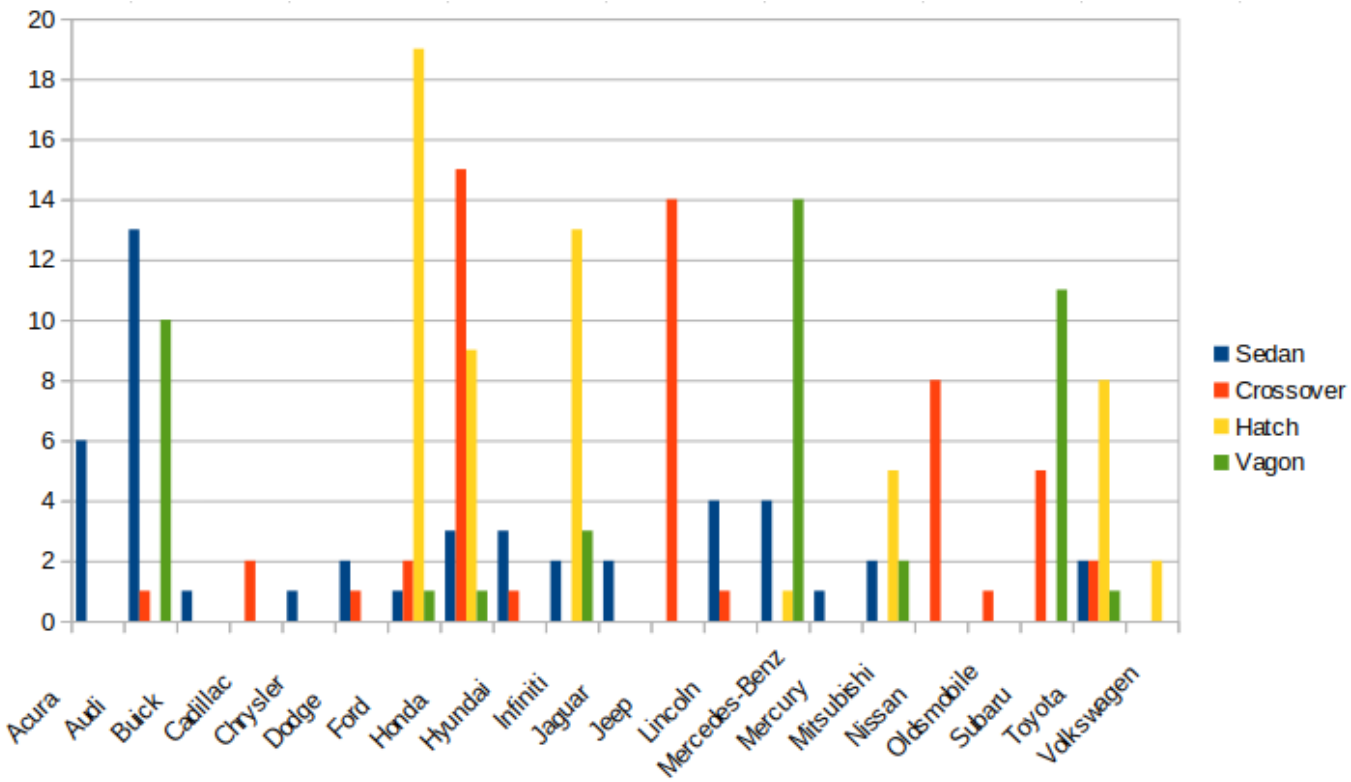
Il dataset ottenuto dal processo di integrazione contiene 1420 valori. Abbiamo così deciso di eliminare tutti i record aventi come attributo body "Other" e "Van" per i motivi elencati nella sezione "Scelte di design per la creazione del dataset".
Attraverso uno script presente su GitHub abbiamo estratto un subsample da 200 record. In particolare, il comando seguente è stato utilizzato per creare il subsample:

```
subsample -n 200 fileOriginale.csv -r > subsample.csv
```

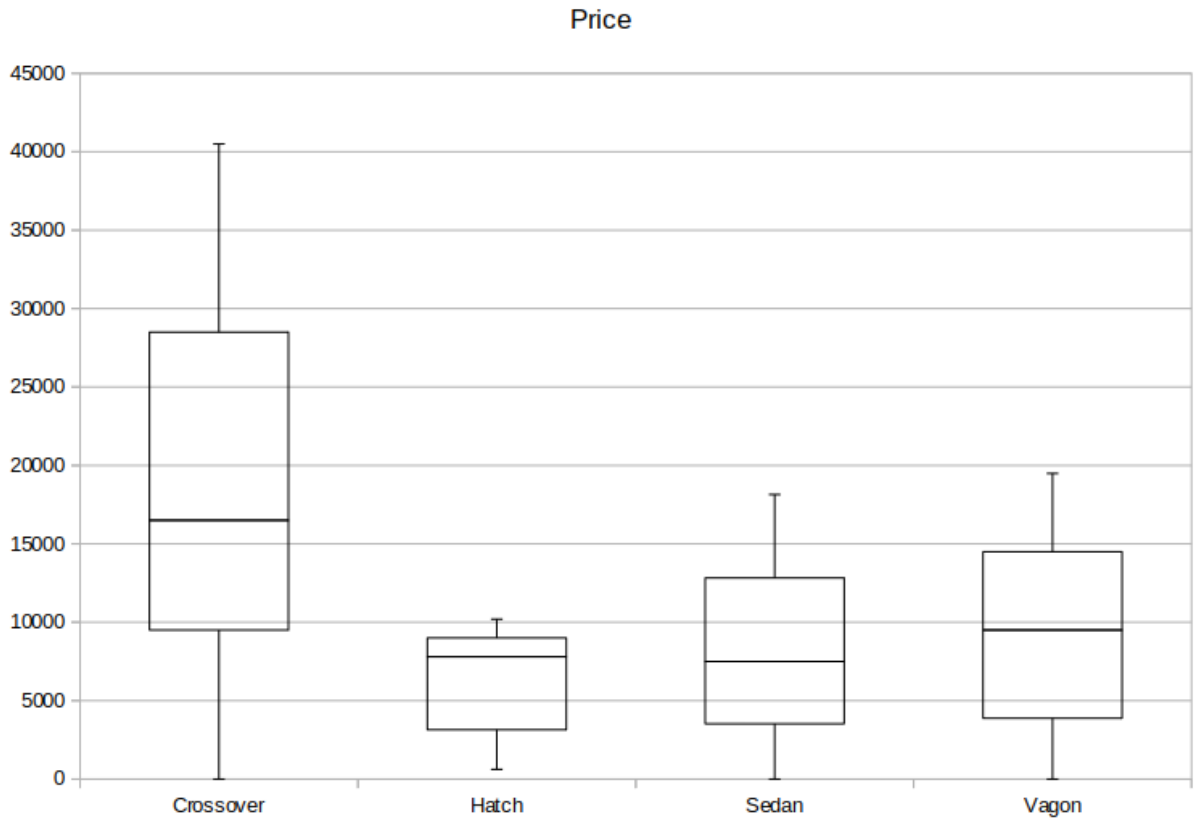
Attraverso il comando *-r* vengono mantenuti gli header.

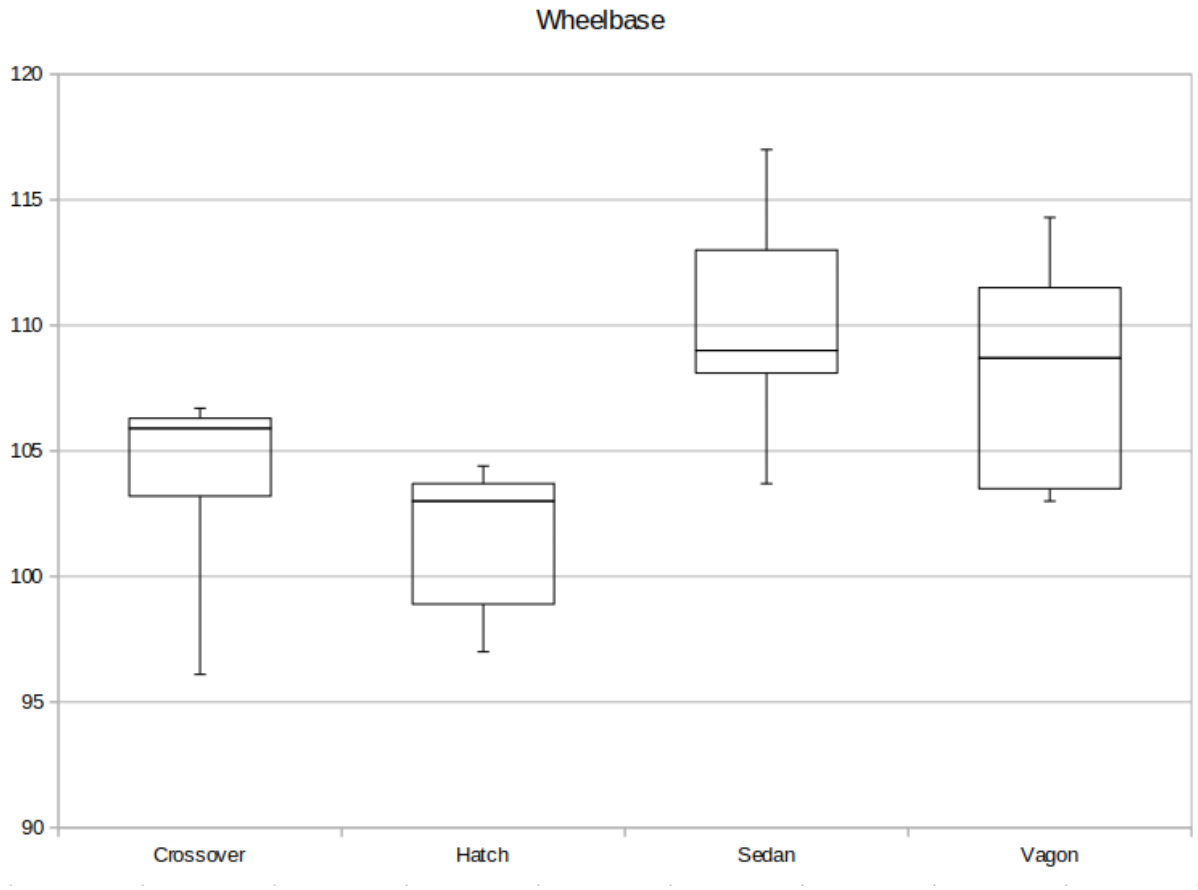
3.2 Analisi esplorativa del training set

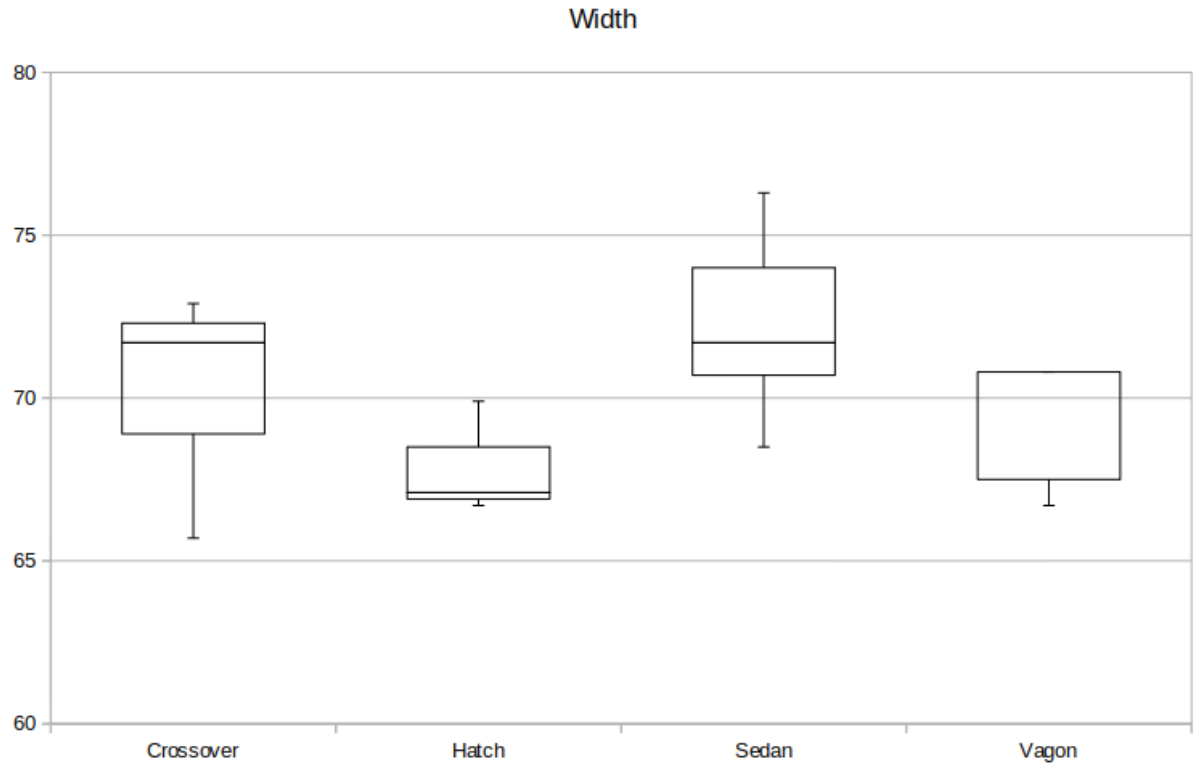
Qui di seguito vengono mostrate le occorrenze, divise per tipo di auto, per ogni marca automobilistica presente nel training set:

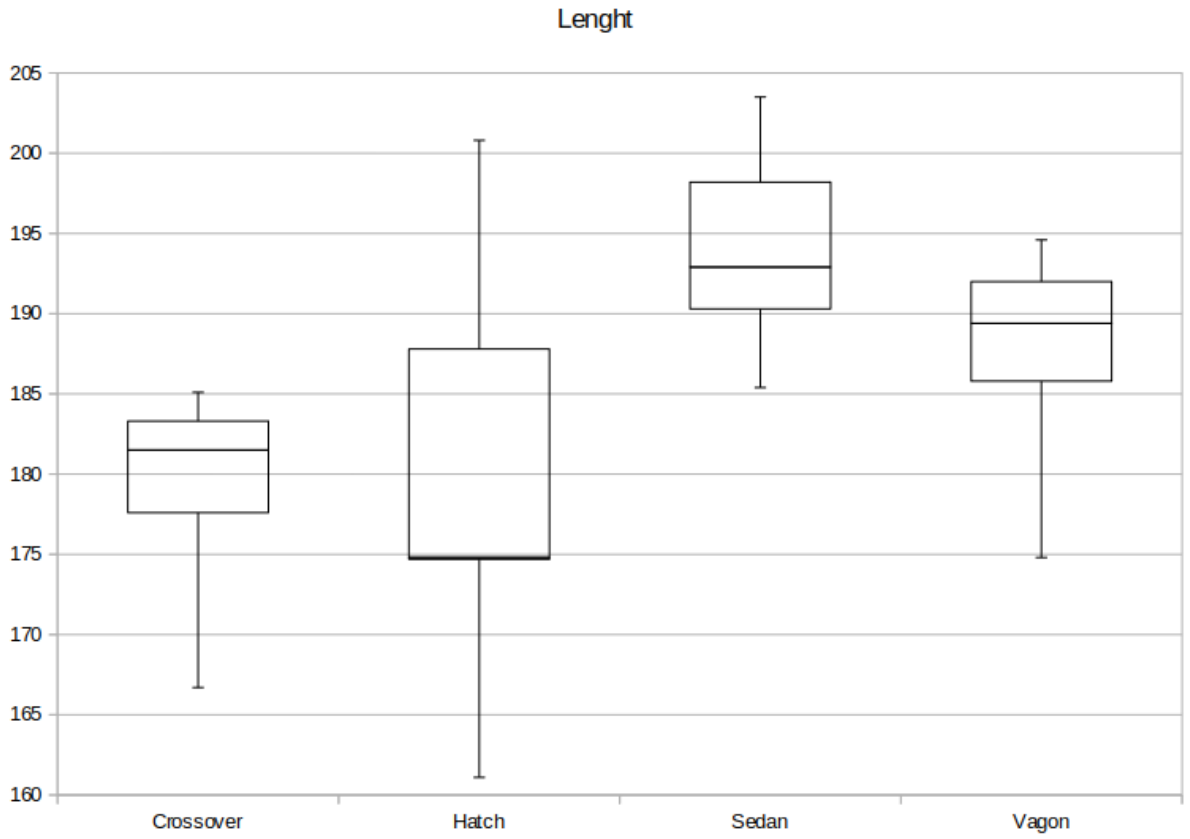


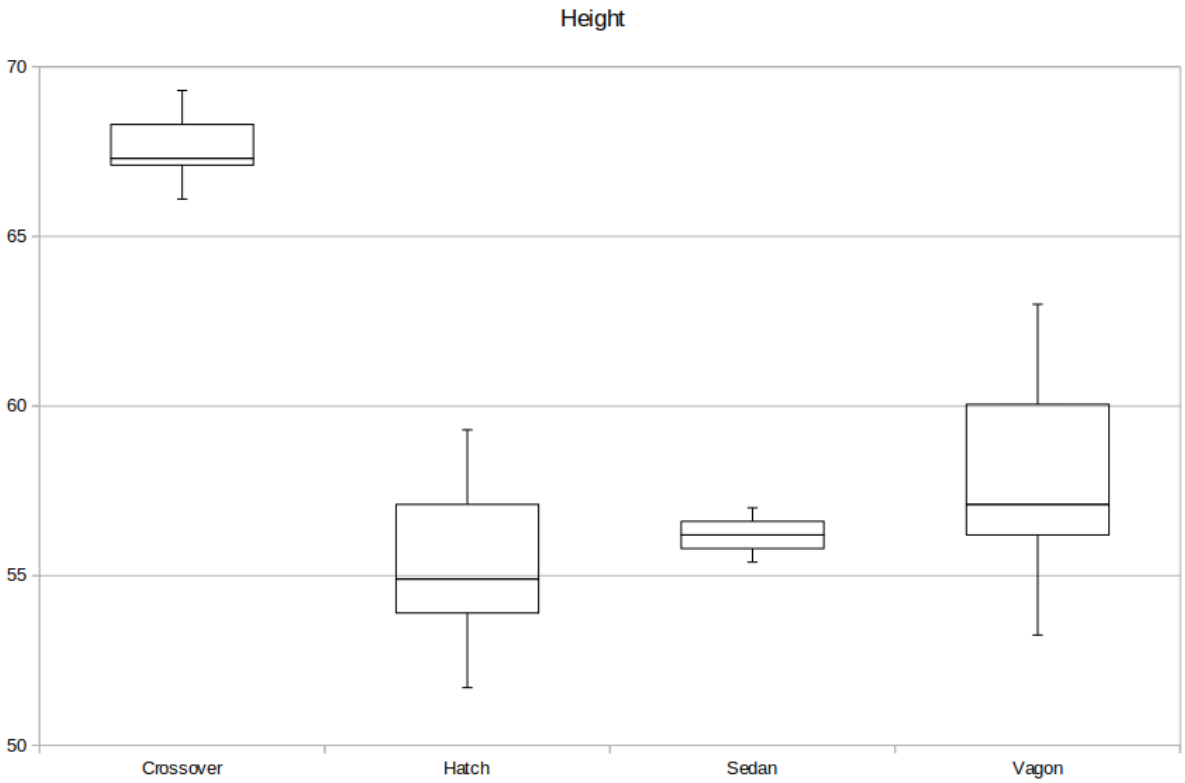
Nei grafici successivi vengono mostrate le 5 caratteristiche usate per la costruzione del modello di clustering k-Means, sottoforma di boxplot, divisi per tipo di auto.


Price
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
Crossover
Hatch
Sedan
Vagon


Wheelbase
120
115
110
105
100
95
90
Crossover
Hatch
Sedan
Vagon


Width
80
75
70
65
60
Crossover
Hatch
Sedan
Vagon


Lenght
205
200
195
190
185
180
175
170
165
160
Crossover
Hatch
Sedan
Vagon



Possiamo dire che:

- Analizzando i grafici "Lenght", "Width", "Wheelbase" e "Height", esiste una netta somiglianza tra le auto contrassegnate come "Sedan" e le auto contrassegnate come "Vagon"; è quindi pensabile che il modello di clustering possa non dividere bene questi tipi di auto.
- Analizzando il grafico "Height", come ci si poteva aspettare, i valori delle auto contrassegnate come "Crossover" sono nettamente più elevati rispetto agli altri tipi.
- Inoltre è possibile che una piccola parte delle auto di tipo "Hatch" venga confusa con le auto di tipo "Sedan" dato che alcuni valori dell'attributo "Lenght" del primo tipo, superiori al 75-esimo percentile, si sovrappongono all'IQR del secondo tipo di auto.

3.3 Utilizzo del modello di clustering K-Means

Come specificato precedentemente il nostro obiettivo è stato quello di voler creare un modello di classificazione non supervisionata, in particolare k-Means, che suddividesse in diversi cluster le auto, dando in input caratteristiche dimensionali.
Per creare il modello di clustering k-Means sono state utilizzate le seguenti librerie:

- *cluster*: contiene funzioni di *Cluster Analysis*;
- *fpc*: contiene funzioni di *Clustering e Cluster validation*;
- *seriation*: necessaria per calcolare e visualizzare matrici di dissimilarità.

Ricavato il dataset di training (sottoinsieme di quello iniziale) abbiamo voluto testare il modello prendendo in considerazione diversi attributi:

1. una prima prova nella quale vengono passati al modello solo caratteristiche dimensionali delle auto, come altezza, lunghezza, larghezza ed interasse;
2. una seconda prova nella quale abbiamo deciso di prendere in considerazione anche il prezzo di vendita di seconda mano di ciascuna auto.

3.3.1 Prova 1

Nel seguente test abbiamo quindi preso solo in considerazione le caratteristiche dimesionali per ogni auto e abbiamo creato un modello di cluster che cercasse di suddividerle in quattro tipologie: "Sedan", "Crossover", "Hatch" e "Vagon".

3.3.2 Prova 2

Nel seguente test abbiamo deciso di prendere in considerazione, oltre alle dimensioni, anche il prezzo di vendita nel mercato dell'usato di ciascuna auto. Quello che ci interessava era sapere se fosse presente una certa correlazione tra la categoria di auto e il corrispondente prezzo da usata.

3.3.3 Creazione del modello

Ricavato il sottoinsieme dei dati e prima di poter applicare l'algoritmo del k-Means abbiamo scalato i dati in modo da normalizzarli ed averli in un intervallo definito. Dopodichè abbiamo applicato l'algoritmo con un numero di cluster differente per le due prove.
Con la funzione *silhouette* abbiamo misurato la similarità tra i cluster ottenuti mentre *clusplot* ci ha permesso di visionare il grafico contente ciascuna istanza e il relativo cluster.
I risultati di ciascuna prova vengono mostrati nel paragrafo 3.5.

3.4 Esperimenti

Successivamente ci è stato chiesto di effettuare i seguenti esperimenti:

- Stima del numero ottimale di cluster usando la misura di Silhouette;
- Stima della Silhouette per ciascun cluster ottenuto rispetto alla soluzione di cluster ottimale;
- Stima della matrice di dissimilarità rispetto alla soluzione di cluster ottimale.

Per il calcolo del numero ottimale di cluster ci siamo affidati alla funzione *cluster.stats* la quale ricava numerose statiche basate sulla distanza tra ciascuno di essi.
Successivamente per il calcolo della matrice di dissimilirità abbiamo utilizzato la funzione *dissplot* contenuta nella libreria *seriation*.

3.4.1 Prova 1

In prima fase, ricavato il dataset e mantenuti attributi esclusivamente numerici escluso *Price*, abbiamo stimato il numero ottimale di cluster attraverso la funzione *cluster.stats*. Il risultato ottenuto è stato pari a otto cluster (Figura 1), considerando un intervallo tra due e otto (abbiamo verificato che per un numero di cluster >8, il valore medio di silhouette decresceva).
Abbiamo poi utilizzato tale risultato per eseguire ulteriori prove e misurare la bontà dell'algoritmo con tale configurazione.

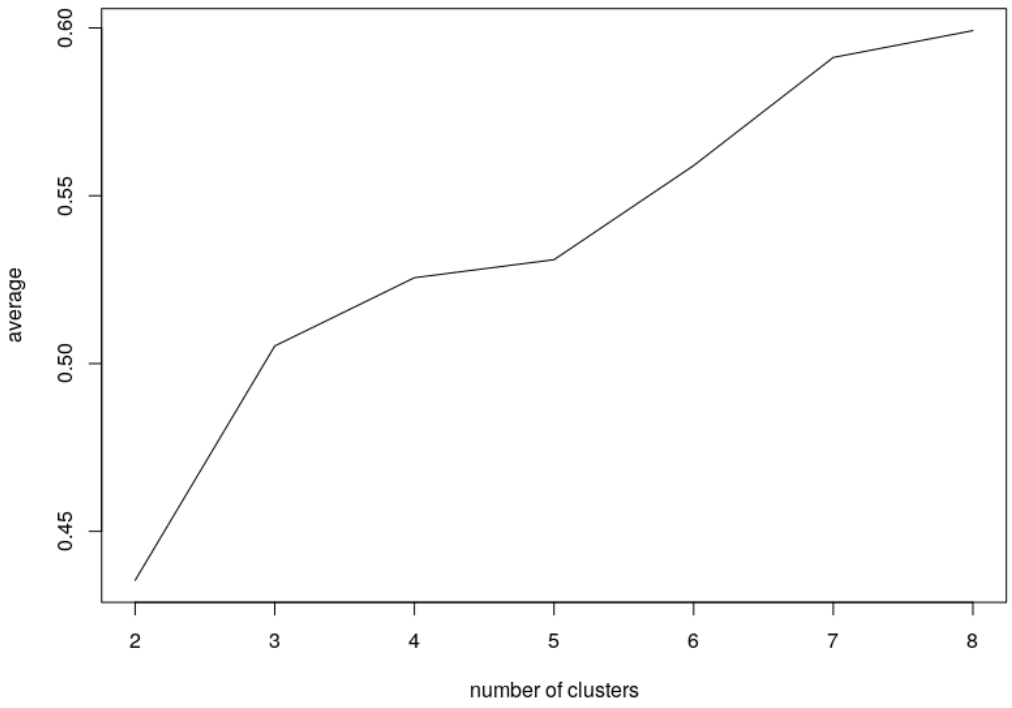


Figure 1: Stima del numero ottimale di cluster. Prova 1.

La funzione *Silhouette* restituisce il numero di elementi contenuti in ciascun cluster e una misura definita nel modo seguente:

$$\frac{b(i) - a(i)}{max(b(i), a(i))} = s(i)$$

dove *b(i)* è la dissimilirità tra l'istanza i e il suo cluster più vicino mentre *a(i)* è la dissimilarità media tra i e tutte le istanze contenute nello stesso cluster. Quindi maggiore sarà il valore di *s(i)* e migliore sarà la qualità del cluster.
Il risultato medio ottenuto è pari a 0.6 come mostrato in Figura 2.

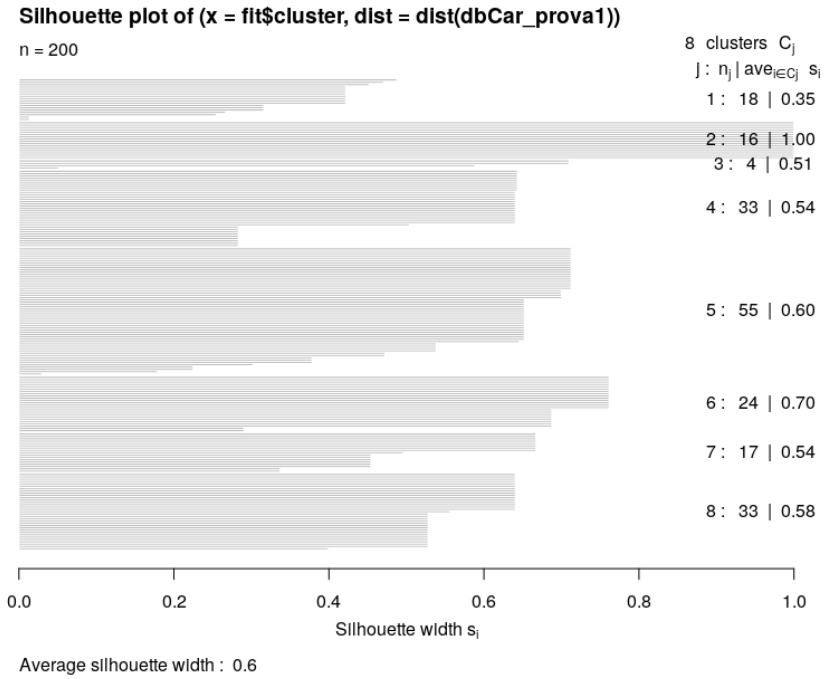


Figure 2: Stima della silhouette con valore ottimale di cluster. Prova 1.

Dopodichè abbiamo calcolato la matrice di dissimilarità attraverso la funzione *dissplot*. Nella figura seguente (Figura 3) viene mostrata la matrice, dalla quale si può notare che i cluster tra loro più simili vengono rappresentati con un colore più scuro e quelli tra di loro più dissimili vengono rappresentati con un colore più chiaro. Inoltre cluster corrispondenti con sè stessi vengono inseriti nella diagonale e di colore scuro.

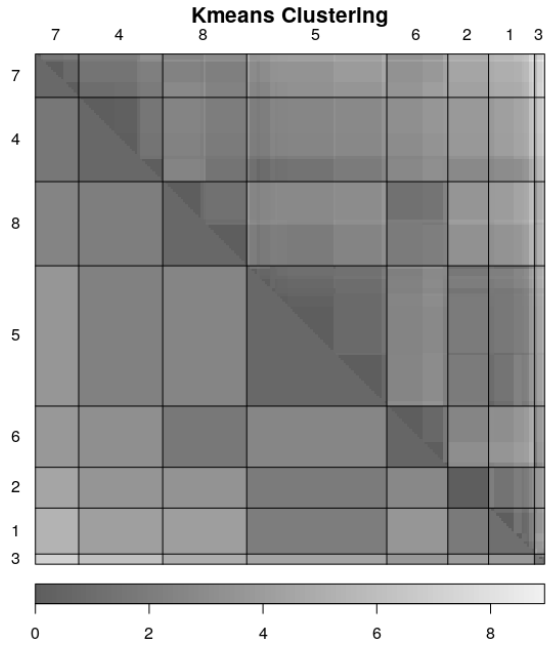


Figure 3: Stima della matrice di dissimilarità con valore ottimale di cluster. Prova 1.

Inoltre abbiamo eseguito l'algoritmo k-Means utilizzando un numero di cluster che concettualmente fosse corretto per noi, ovvero quattro, data la tipologia di dati presenti nel dataset.
Dai risultati che abbiamo ottenuto, in particolare analizzando l'output di *Silhouette* (Figura 4), abbiamo reputato che tale configurazione, anche se non in maniera ottimale, riuscisse comunque a classificare in maniera discreta i dati inseriti.

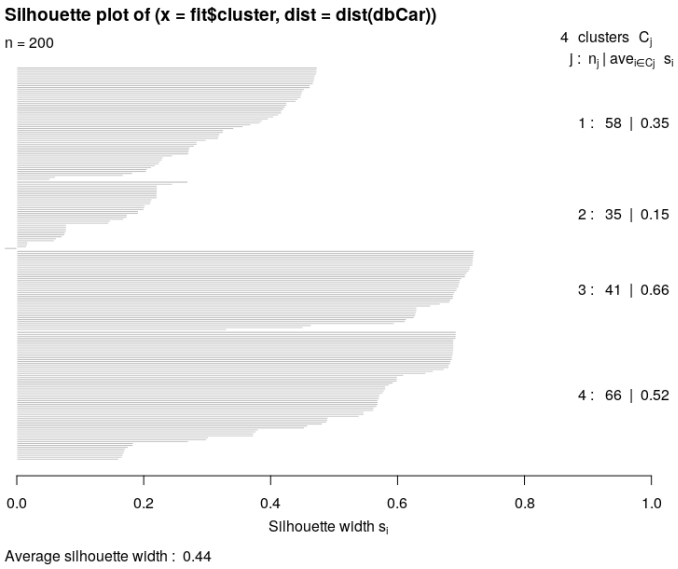


Figure 4: Stima della silhouette con numero di cluster pari a 4. Prova 1.

3.4.2 Prova 2

Nel seguente test, ovvero tenendo conto anche dell'attributo "Price", il numero ottimale di cluster ottenuto è stato invece pari a quattro (Figura 5).

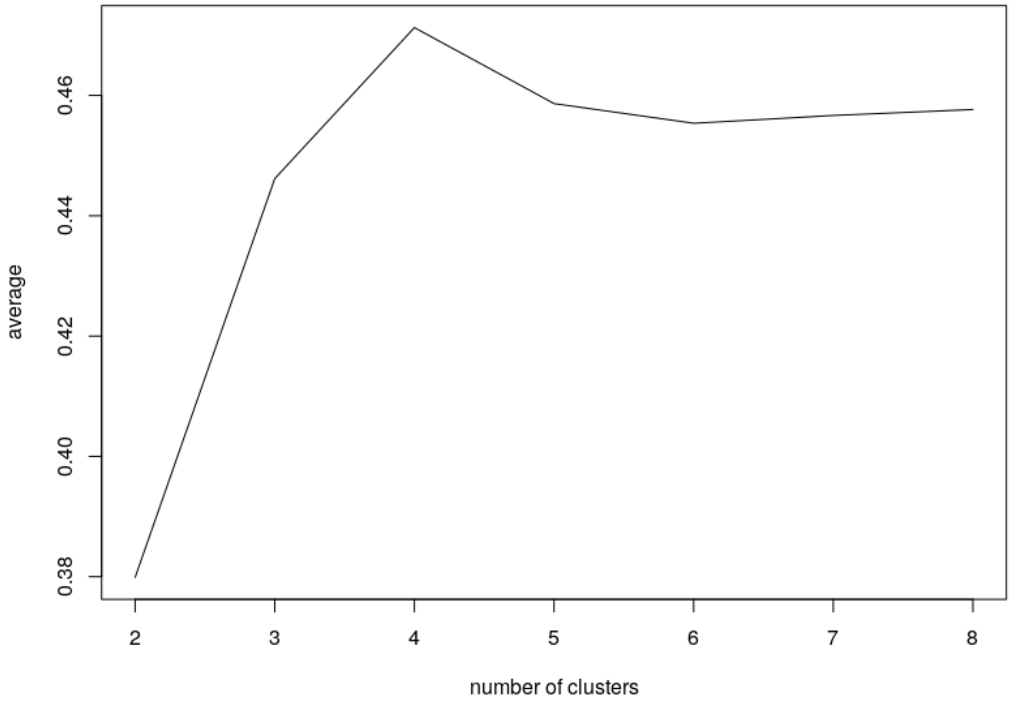


Figure 5: Stima del numero ottimale di cluster. Prova 2.

Attraverso il risultato della funzione *Silhouette* (Figura 6) abbiamo notato un notevole peggioramento, suggerito anche dal valore *s(i)*. Inoltre, analizzando la Figura 6, si può notare che nel cluster 1 sono presenti solamente 7 elementi.

La matrice di dissimilarità ottenuta è mostrata in Figura 7.

Si può notare una tendenza di colore che verte più sullo scuro rispetto alla matrice di dissimilarità della prova precedente (Figura 3).

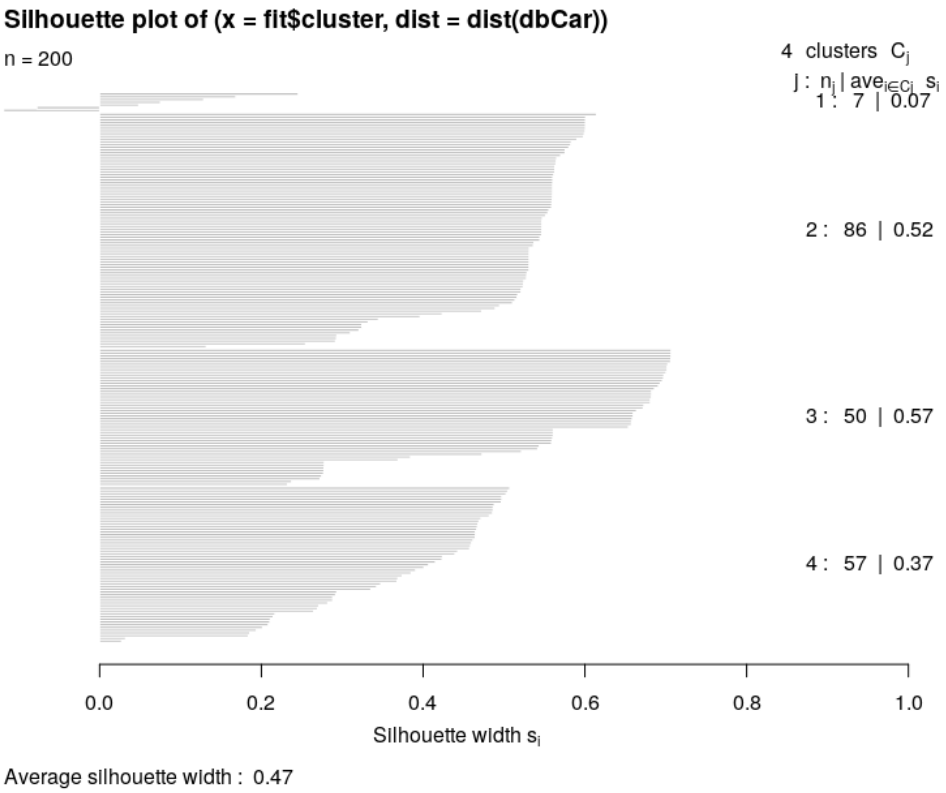


Figure 6: Stima della silhouette con valore ottimale di cluster. Prova 2.

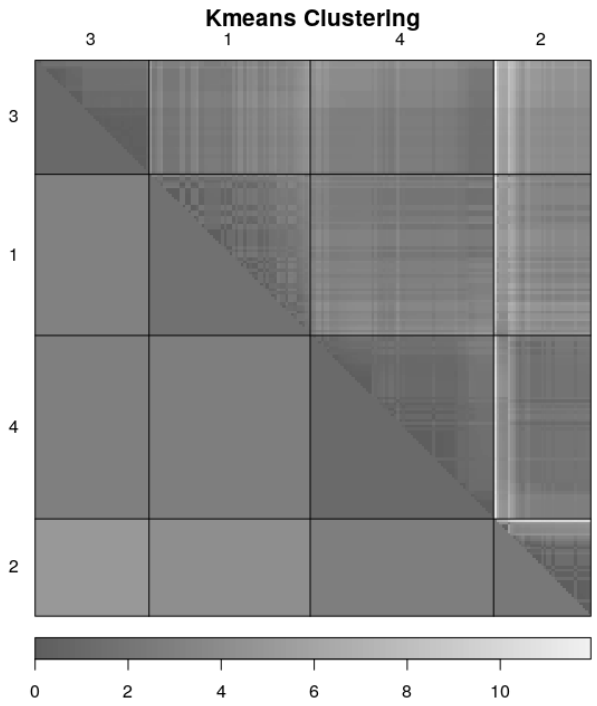


Figure 7: Stima della matrice di dissimilarità con valore ottimale di cluster. Prova 2.

3.5 Analisi dei risultati ottenuti

I grafici contenenti i cluster ottenuti dalle prove mostrate precedentemente sono riporati di seguito (Figura 8, 9 e 10).

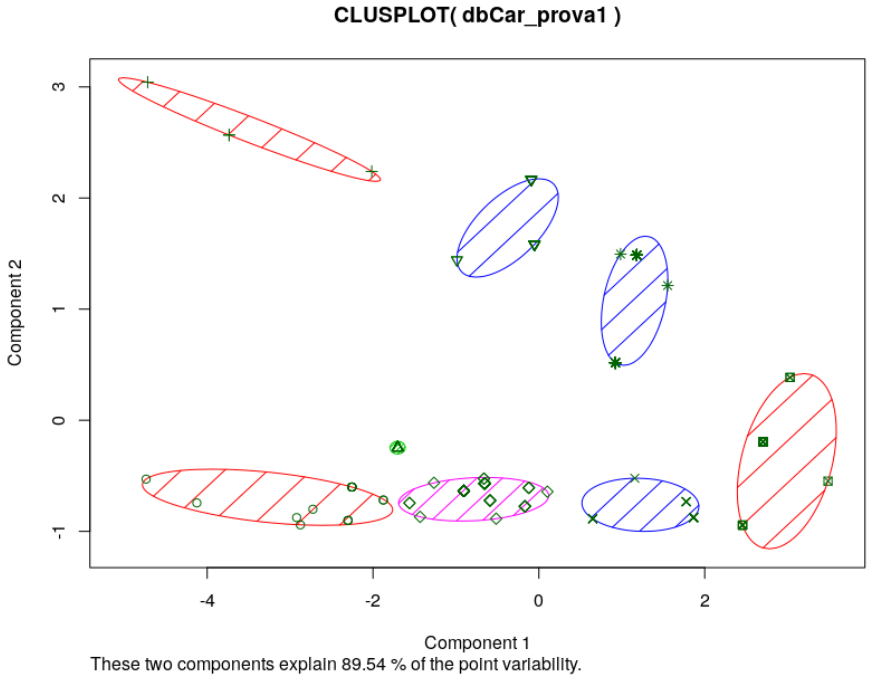


Figure 8: Cluster generati dall'algoritmo k-Means, cluster pari a 8. Prova 1.

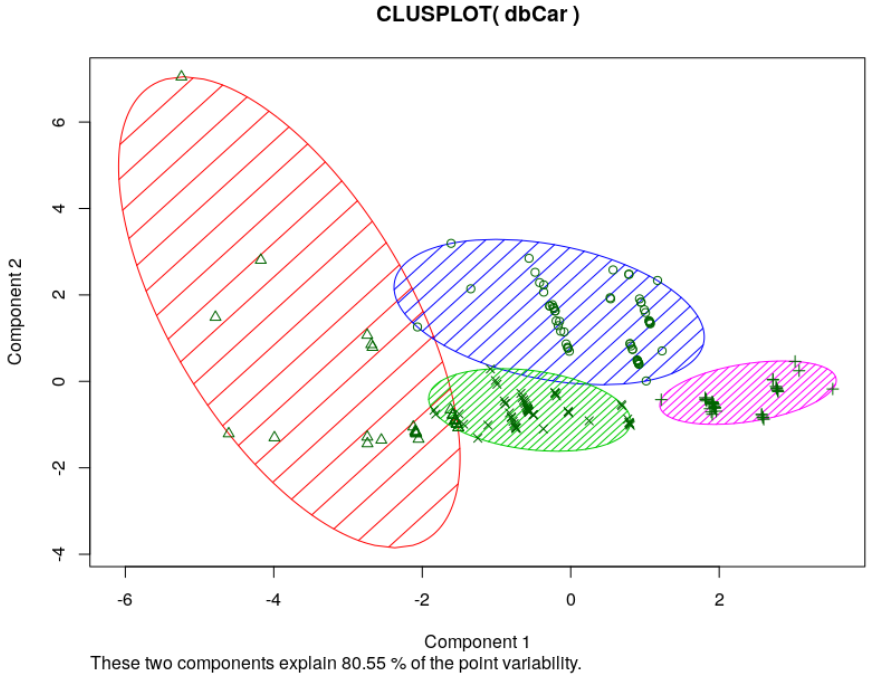


Figure 9: Cluster generati dall'algoritmo k-Means, cluster pari a 4. Prova 1.

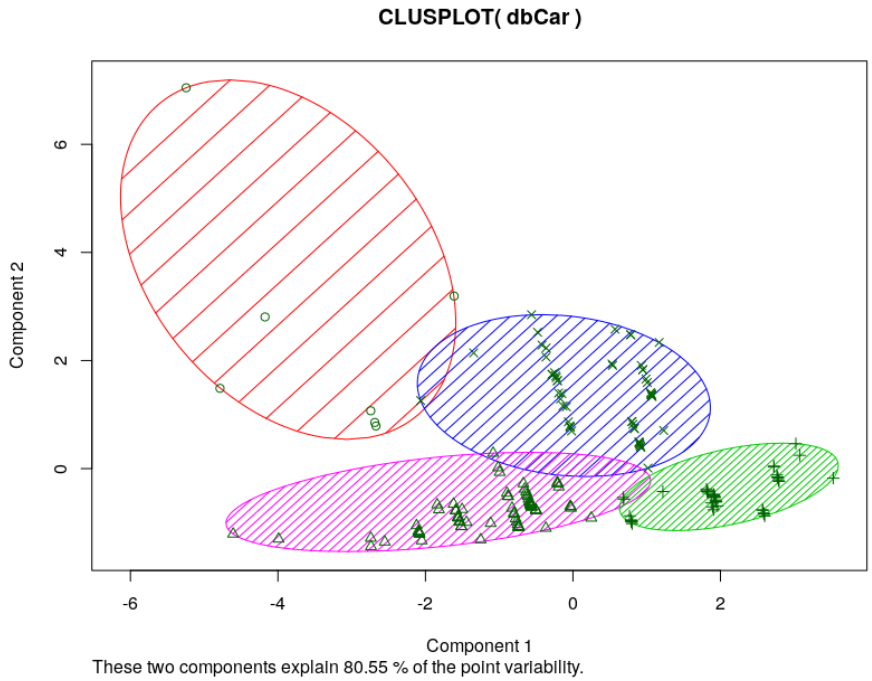


Figure 10: Cluster generati dall'algoritmo k-Means. Prova 2.

Dai risultati ottenuti, mostrati nella tabella segeunte, possiamo dedurre che nella prova 1, nella quale abbiamo usato per la costruzione del modello solo le caratteristiche dimensionali, abbiamo ottenuto sì dei risultati migliori ma non come ci aspettavamo.

Cluster	*Crossover*	*Hatch*	*Sedan*	*Vagon*
1	1	0	17	0
2	1	0	5	10
3	4	0	0	0
4	0	28	2	3
5	0	13	23	19
6	24	0	0	0
7	1	16	0	0
8	22	0	0	11

Mentre utilizzando un numero di cluster pari a quattro il risultato è stato il seguente:

Cluster	*Crossover*	*Hatch*	*Sedan*	*Vagon*
1	47	0	0	11
2	5	0	20	10
3	1	39	0	1
4	0	18	27	21

Dalla tabella precedente si può notare come il modello non riesca del tutto a distinguere due tipologie di auto. Come pronosticato durante l'analisi esplorative del training set, il modello creato non riesce a dividere correttamente i record contrassegnati con "Vagon" con i record contrassegnati con "Sedan", mischiandoli.

La tabella seguente invece mostra la classificazione ottenuta durante la prova 2.
Da come si può notare inserisce praticamente nello stesso cluster i valori contrassegnati con "Vagon", quelli contrassegnati con "Sedan" e una buona parte dei valori contrassegnati con "Hatch". Inoltre crea un cluster contenente solamente 7 valori, anche questi mischiati.

Cluster	*Crossover*	*Hatch*	*Sedan*	*Vagon*
1	4	0	3	0
2	2	13	42	29
3	1	44	2	3
4	46	0	0	11

3.6 Conclusioni

Analizzando quindi i risultati ottenuti e le diverse prove effettuate possiamo dire che il modello di clustering che abbiamo costruito sia riuscito a classificare i dati inseriti in maniera discreta.
Possiamo dire che non abbiamo trovato una correlazione tra "Price" e gli attributi che rappresentano le caratteristiche delle auto.
Come mostrato nel capitolo precedente, il modello lavora meglio nella prova 1, ossia nel caso in cui non viene preso in considerazione l'attributo "Price" e si ha un numero di cluster pari a otto. Valutiamo inoltre come buon risultato anche modello svolto nella prova 1 caratterizzato da quatto cluster.
Va reso noto che non tutte le istanze vengono però classificate correttamente, questo perchè seppure alcune auto appartengano ad una certa categoria hanno caratteristiche molto simili ad altre tipologie.

Per eventuali sviluppi futuri si potrebbe puntare ad un miglioramento della misura di *Silhouette* attraverso l'introduzione di nuovi attributi che caratterizzano una particolare categoria, come per esempio la coppia motrice o l'altezza da terra.