Report Data Mining

Ferri Lorenzo (607828)
email lorenzoferri1995@gmail.com

Pappolla Roberta (534109)
email r.pappolla@studenti.unipi.it

Varesi Marco (591464)
email marcovaresi1996@gmail.com

Data Mining (654AA), Anno accademico 2019/2020

Indice

1 Data Understanding **1**
1.1 Data Semantics . . . 1
1.2 Distributions and Statistics . . . 2
1.3 Assessing data quality (missing values and outliers) . . . 4
1.4 Pairwise Correlations and Elimination of Redundant Variables . . . 6
1.5 Variables transformations . . . 8

2 Clustering **8**
2.1 Density-Based Clustering . . . 8
2.2 K-Means Clustering . . . 9
2.3 Hierarchical Clustering . . . 10
2.4 Results Analysis . . . 11

3 Association Rules Mining **14**
3.1 Frequent Patterns . . . 14
3.2 Association Rules . . . 16
3.3 Results Analysis . . . 17

4 Classification **18**
4.1 Learning of decision tree . . . 18
4.2 Decision tree interpretation . . . 18
4.3 Decision tree validation . . . 19
4.4 The best prediction model . . . 19

Elenco delle figure

1 Distribuzioni Attributi Numerici . . . 3
2 CrossTab VehicleAge . . . 3
3 CrossTab Make pre e post eliminazione MMR non consistenti . . . 5
4 BoxPlot Attributi Numerici . . . 5
5 Scatter Plot dei prezzi dei veicoli . . . 6
6 Scatter Plot di Densità dei prezzi dei veicoli . . . 7
7 Scatter Plot VehBCost vs VehOdo . . . 7
8 Distanze dal Quinto punto più vicino . . . 9
9 Dendogramma con il metodo complete sulla sinistra e col metodo ward sulla destra . . . 11
10 Scatter Plot Risultati DBSCAN . . . 12
11 Centroidi K-means Clustering. . . . 12
12 Distribuzione della variabile VehicleAge in base ai cluster . . . 14
13 Numero di Frequent Patterns per livello di *Min Supp* . . . 15
14 Numero di Association Rules per livello di *Min Conf* . . . 16
15 Albero decisionale con profondità 3 . . . 19

Elenco delle tabelle

1 Tabella dei missing value e degli outliers . . . 4
2 Top 5 clusterizzazioni . . . 11
3 Composizione valori VehicleAge all'interno della K-Means Clustering. . . . 13
4 Tabella dei possibili valori assegnati agli attributi . . . 18
5 Top 10 delle combinazioni degli Hyperparameter . . . 20
6 Valore di precision accuracy recall e f1 score per i top 4 model . . . 20
7 Risultati modello finale . . . 20

1 Data Understanding

1.1 Data Semantics

Il dataset fornito presentava alcuni errori sintattici ed incongruenze semantiche. Per diversi attributi inoltre era possibile ricavare nuovi elementi. Di seguito si riporta un elenco degli attributi più importanti (specialmente per quanto riguarda le analisi successive) e quelli che hanno subito delle modifiche:

- **Make**, variabile categorica che rappresenta il nome della casa produttrice del veicolo. Il valore "Toyota Scion" è stato sostituito con il valore "Scion", essendo valori riferiti al medesimo Make.
- **Nationality**, variabile categorica che indica la nazionalità della casa produttrice del veicolo. L'attributo presenta molti valori per "AMERICAN" e pochi per gli altri ('Other', 'Other asian', 'Top ten asian'), che quindi sono stati sostituiti tutti con un unico valore: 'Not American'.
- **Model**, variabile categorica che rappresenta il modello del veicolo. Questo attributo è stato scorporato per ricavarne un ulteriore attributo rilevante, l' "Engine". Erano inoltre presenti diversi valori del Trim che sono stati inseriti nell'apposito attributo.
- **SubModel**, variabile categorica che rappresenta la tipologia di veicolo oggetto della vendita. Anche in questo caso l'attributo è stato rielaborato scorporandolo negli attributi "SubModelSpecifics", "Doors" ed "Engine" e ricavandone alcuni valori del Trim.
- **SubModelSpecifics**, variabile categorica che rappresenta le specifiche del sotto-modello del veicolo, ricavata dallo scorporamento dell'attributo "SubModel".
- **Engine**, variabile categorica che rappresenta la tecnologia del motore montato sul veicolo, ricavata dallo scorporamento degli attributi "Model" e "SubModel". I valori di cui è composto sono principalmente la cilindrata (es. 3.5L) ed il numero di cilindri (es. V8).
- **Trim**, variabile categorica che rappresenta gli optional inclusi nel veicolo venduto.
- **Doors**, variabile categorica che rappresenta il numero di porte del veicolo venduto all'asta. E' stata ricavata dallo scorporamento dell'attributo "SubModel". Il dataset è costituito quasi esclusivamente da veicoli con 4 porte.
- **WheelType**, variabile categorica che rappresenta il tipo di cerchioni montati sul veicolo. E' costituita da tre valori: Alloy, Covers e Special.
- **WheelTypeId**, variabile categorica numerica che identifica con un ID il tipo di cerchioni del veicolo indicato nell'attributo "WheelType": 1.0 = Alloy; 2.0 = Covers; 3.0 = Special. Era presente anche un valore pari a 0.0, a cui erano associati 4 oggetti, che è stato sostituito con un missing value.
- **WheelDrive**, variabile categorica che indica il numero di ruote motrici, ricavata dallo scorporamento dell'attributo "SubModel". E' costituita da due valori: 2WD e 4WD.
- **AcquisitionType**, variabile categorica assente dal dataset fornito.
- **VehYear**, variabile ordinale che indica l'anno di costruzione del veicolo venduto all'asta. Il suo dominio è [2001,2010].

- **PurchaseDate**, variabile ordinale che indica la data dell'asta in cui il veicolo è stato venduto. Tutte le vendite sono state rilevate nel biennio 2009-2010. Sono presenti 7 date inconsistenti con gli attributi "VehYear" e "VehAge", che sono state corrette.

- **VehAge**, variabile numerica che indica gli anni posseduti dalla macchina al momento della vendita all'asta. I suoi valori possono essere ricavati dalla formula: VehAge = PurchaseDate - VehYear. Il suo dominio è [0,9].

- **VehBCost**, variabile numerica che indica il prezzo a cui il veicolo è stato venduto all'asta. Il suo dominio è [1, 36485];

- **VehOdo**, variabile numerica intera che indica il numero di miglia percorsi dal veicolo al momento della vendita. Il suo dominio è [9878, 109217].

- **MMRAcquisitionAuctionAveragePrice**, variabile numerica che rappresenta una stima del prezzo all'asta del veicolo, identificato da modello, sottomodello e anno di produzione, in condizioni medie al momento della vendita all'asta.

- **MMRAcquisitionAuctionCleanPrice**, variabile numerica che rappresenta una stima del prezzo all'asta del veicolo, in condizioni sopra la media al momento della vendita all'asta.

- **BYRNO**, variabile numerica intera che rappresenta l'identificativo dell'acquirente. Un'analisi più dettagliata del Dataset ha evidenziato che la maggioranza dei compratori ha acquistato un elevato numero di veicoli. Se ne deduce dunque che i partecipanti alle aste siano rivenditori di usato o proprietari di noleggio auto.

- **WarrantyCost**, variabile numerica relativa al costo addizionale dovuto alla garanzia sulla macchina. La garanzia avrà una durata di 36 mesi o di 36000 miglia.

- **isBadBuy**, variabile categorica binaria che definisce la qualità dell'acquisto dal punto di vista dell'acquirente. Costituisce la variabile target.

Sono stati eliminati sia dal Training che dal Test Set fin da subito i seguenti attributi, in quanto evidentemente non significativi per le analisi successive: Color; WheelType (perché già presente sotto forma di Id: 'WheelTypeId').

1.2 Distributions and Statistics

Sono state generate le distribuzioni dei valori di tutti gli attributi del Dataset, usando i grafici: barcharts, per gli attributi categorici; istogrammi e funzioni di densità di probabilità, per gli attributi numerici. Gli istogrammi sono stati rappresentati con un numero di Bins dato dalla formula di Sturges ($\log_2 n + 1$) ed escludendo gli outliers. Sono stati inoltre generati i CrossTab di alcuni attributi rilevanti condizionati alla variabile "IsBadBuy", per verificare come si distribuiscono i valori di tale variabile sui valori di ciascuno degli altri attributi.
Per quanto riguarda le distribuzioni dei valori si riportano gli istogrammi degli attributi "VehBCost", "VehOdo" e "VehicleAge" e la funzione di densità di probabilità dell'attributo "MMRAcquisitionAuctionAveragePrice":

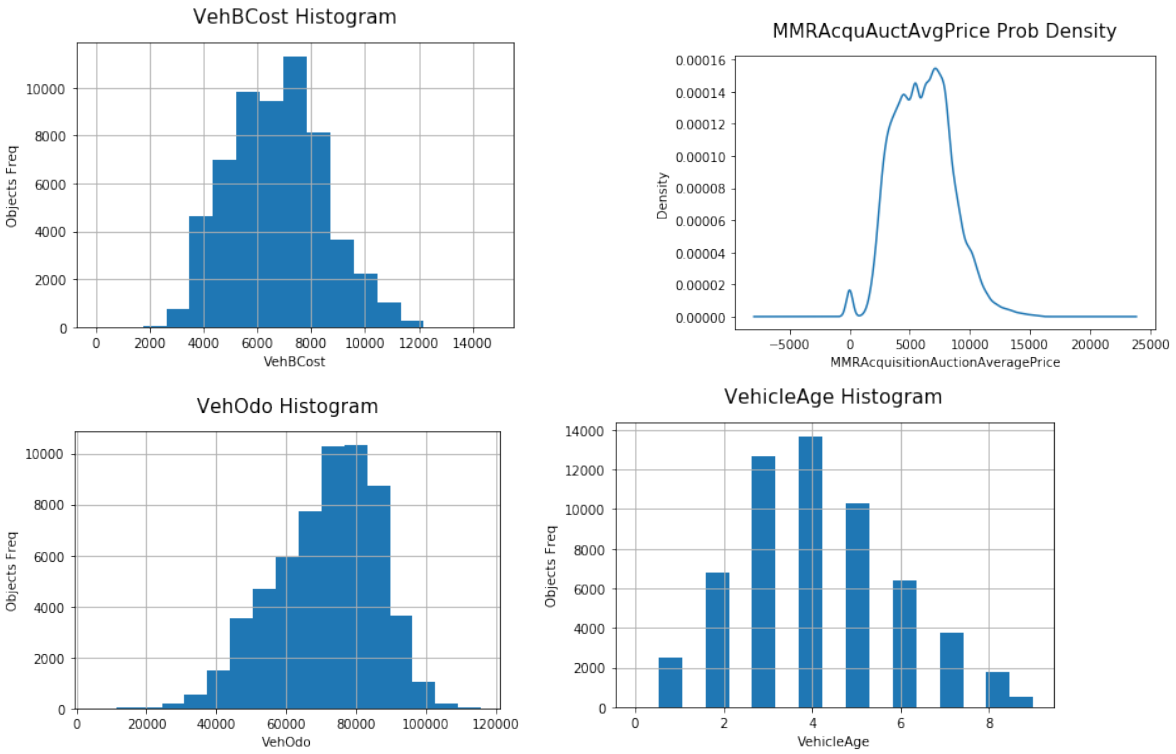


Figura 1: Distribuzioni Attributi Numerici

per quanto riguarda invece i CrossTab si riporta l'unico risultato realmente interessante, riguardante l'attributo "VehicleAge":

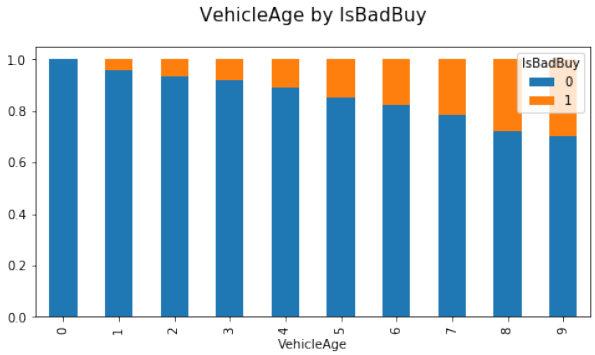


Figura 2: CrossTab VehicleAge

Da questo grafico è evidente come il valore 1 dell'attributo "IsBadBuy" sia gradualmente sempre più presente in percentuale nei veicoli con un'età via via maggiore. Ne deriviamo un primo risultato importante: la soddisfazione degli acquirenti è più bassa se i veicoli hanno un'età elevata, indipendentemente dal fatto che essi possano essere stati pagati meno.

Sono state prodotte le statistiche di tutte le variabili numeriche ed in relazione anche a quanto si evince dai grafici delle distribuzioni possiamo asserire che gli attributi "VehBCost", "VehOdo", "VehicleAge" e "MMRAcquisitionAuctionAveragePrice" si distribuiscono Normalmente con parametri Media e Deviazione Standard ottenuti dalle statistiche:

Media(VehBCost) = 6.730,01;
Media(VehOdo) = 71.478,09;
Media(VehicleAge) = 4,17;
Media(MMRAcquAuctAveragePrice)= 6.128,13;

StdDev(VehBCost) = 1.762,07;
StdDev(VehOdo) = 14.591,22;
StdDev(VehicleAge) = 1,71;
StdDev(MMRAcquAuctAveragePrice)= 2.456,63;

1.3 Assessing data quality (missing values and outliers)

Nella tabella 1 sono riportate tutte le variabili con il relativo numero di missing value.Occorre sottolineare che le variabili relative all'indice MMR presentano numerosi valori a 0 (oltre 600 per la stima relativa al tempo di vendita del veicolo e oltre 300 per la stima al tempo corrente). Questi valori sono stati considerati come errori e sostituiti con dei valori NULL.

	Miss. Val.		Miss. Val.
IsBadBuy	0	PurchDate	0
Auction	0	VehYear	0
VehicleAge	0	Make	0
Model	0	Trim	1911
Submodel	7	Color	7
Transmission	8	WheelTypeID	2573
WheelType	2577	VehOdo	0
Nationality	4	Size	4
TopThreeAmericanName	4	AUCGUART	55703
MMRAcquisitionAuctionCleanPrice	13	VNZIP1	0
MMRAcquisitonRetailCleanPrice	13	VehBCost	0
MMRCurrentAuctionCleanPrice	245	WarrantyCost	0
MMRCurrentRetailCleanPrice	245	PRIMEUNIT	55703
MMRAcquisitionAuctionAveragePrice	13	BYRNO	0
MMRAcquisitionRetailAveragePrice	13	VNST	0
MMRCurrentAuctionAveragePrice	245	IsOnlineSale	0
MMRCurrentRetailAveragePrice	245		

Tabella 1: Tabella dei missing value e degli outliers

Come si può notare dalla tabella, sono stati eliminati sia dal Training che dal Test Set i seguenti attributi perché contenenti un numero troppo elevato di Missing Values: PRIMEUNIT; AUCGUART; SubModelSpecifics; Wheeldrive. La sostituzione di quest'ultimi infatti altererebbe sostanzialmente la distribuzione dell'attributo.

Per quanto riguarda la sostituzione dei missing value degli attributi SubModel, Size, Trim e Transmission è stata utilizzata la moda del dataset raggruppato per modello sottomodello e anno del veicolo. Per gli attributi Nationality e TopThreeAmericanName invece è stata utilizzata la moda ma sul raggruppamento generato dal solo attrobuto Make. Un'analisi differente è stata applicata all'attributo WheelTypeID. La caratteristica dei cerchi è infatti stata associata al tipo di trim della vettura, utilizzando quindi un raggruppamento per modello, sottomodello e trim.

Per quanto riguarda l'attributo MMRAcquisitionAuctionAveragePrice è necessario fare un discorso più ampio prendendo in considerazione la definizione dell'indice MMR (definita nel paragrafo 1.1). Raggruppando i veicoli in base a modello, sottomodello, anno e giorno dell'asta

si dovrebbe ottenere una varianza dell'indece MMR (determinato il giorno dell'asta) pari a 0, ma in 5589 macchine questa regola non viene rispettata e otteniamo una varianza superiore a 250000. Questi casi sono stati considerati errori e l'eliminazione di queste automobili dal dataset non modifica la distribuzione di nessun attributo (nell'immagine 3 viene mostrata la distribuzione dei BadBuy, prima e dopo la rimozione delle vetture, in base all'attributo Make). Una volta eliminate le vetture associate a un indice MMR inconsistente, la sostituzione dei missing value è stata eseguita utilizzando la media del valore del MMR calcolata sul raggruppamento generato dal modello, sottomodello, anno e giorno dell'asta.

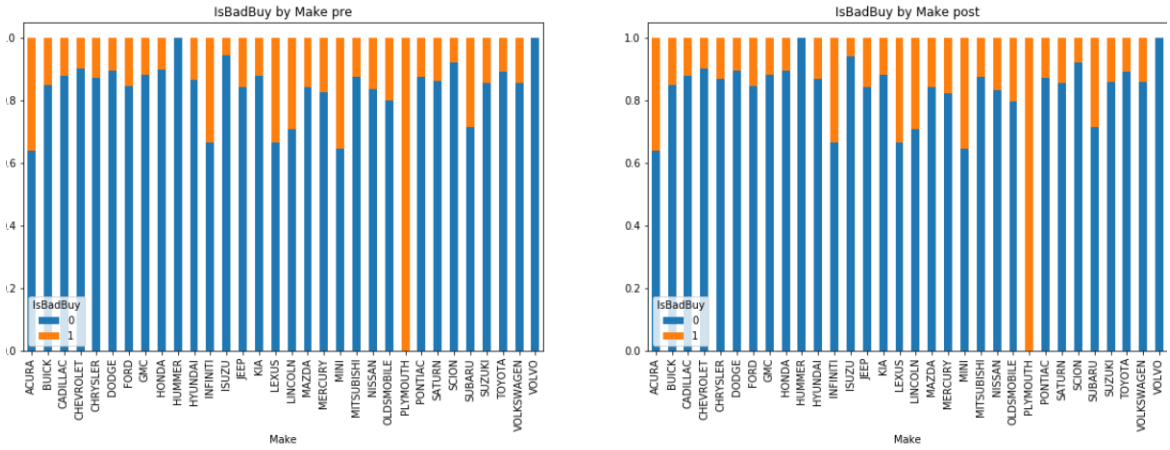


Figura 3: CrossTab Make pre e post eliminazione MMR non consistenti

Sono stati rilevati inoltre gli Outliers degli attributi numerici mediante i loro Box Plot. Di seguito si riportano quelli degli attributi che si riferiscono al prezzo del veicolo: 'VehBCost' e 'MMRAcquisitionAuctionAveragePrice', raggruppati in base al valore dell'attributo 'IsBadBuy'. Essi evidenziano come gli Outliers relativi ad un prezzo elevato del veicolo siano maggiormente associati al valore 1 dell'attributo 'IsBadBuy'. Questo indica che è possibile che esista una certa relazione negativa tra il costo elevato del veicolo e la soddisfazione dell'acquirente.

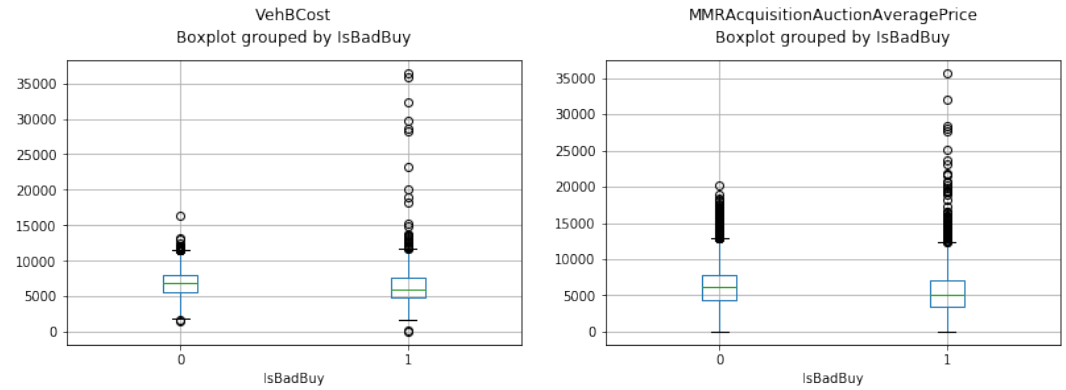
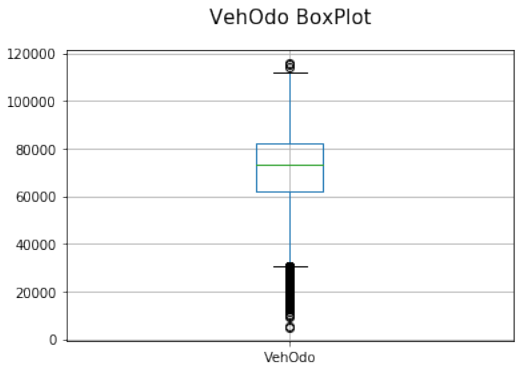


Figura 4: BoxPlot Attributi Numerici

Il BoxPlot dell'attributo "VehOdo" rileva come outliers i veicoli con miglia inferiori a 30.000.



Gli outliers identificati sono outliers del singolo attributo perciò non è obbligatorio eliminarli dal dataset. L'unica modifica effettiva, oltre all'analisi della frequenza degli attributi categorici (sono stati individuati 21 modelli associati a una singola macchina), è la sostituzione del make TOYOTA SCION con SCION per accorpare il singolo veicolo al gruppo di macchine SCION.

1.4 Pairwise Correlations and Elimination of Redundant Variables

E' stata calcolata la correlazione (di Pearson, di Spearman e di Kendall) e sono stati rappresentati gli Scatter Plot di ogni coppia di variabili numeriche dopo aver riempito i missing values. Il risultato più interessante riguarda i prezzi MMR, che sono altamente correlati tra loro. In particolare si riportano i seguenti coefficienti di correlazione di Pearson:

PCorr(MMRAcquisitionAuctionAveragePrice; MMRAcquisitionAuctionCleanPrice) = 0,9898
PCorr(MMRAcquisitionRetailAveragePrice; MMRAcquisitonRetailCleanPrice) = 0,9896
PCorr(MMRCurrentAuctionAveragePrice; MMRCurrentAuctionCleanPrice) = 0,99
PCorr(MMRCurrentRetailAveragePrice; MMRCurrentRetailCleanPrice) = 0,9894

Anche il costo del veicolo all'asta "VehBCost" è risultato correlato ai prezzi MMR. L'indice di correlazione di Spearman per la coppia di variabili (MMRAcquisitionAuctionAveragePrice; VehBCost) è pari a 0,846.
Possiamo fornire una rappresentazione grafica della correlazione tra questi attributi raffigurando a titolo esemplificativo gli Scatter Plot per le coppie di variabili (MMRAcquisitionAuctionAveragePrice; MMRAcquisitionAuctionCleanPrice) e (MMRAcquisitionAuctionAveragePrice; VehBCost):

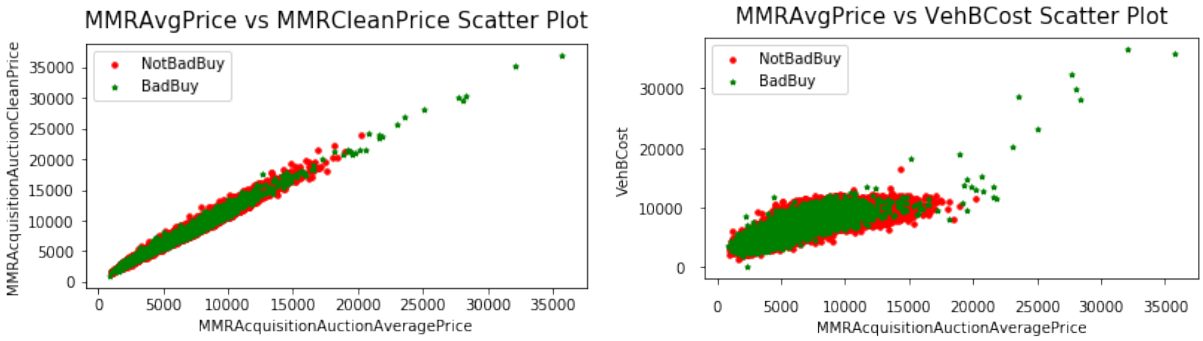


Figura 5: Scatter Plot dei prezzi dei veicoli

Gli Scatter Plot per le altre coppie di prezzi MMR sono simili a quello riportato sopra. Oltre all'informazione sulla dipendenza lineare della coppia di variabili questi Scatter Plot forniscono una distinzione tra i veicoli che hanno costituito un pessimo acquisto e gli altri. Viene messo in evidenza un risultato che ritroveremo anche in altre analisi successive: i veicoli molto costosi, in termini di prezzo MMR e costo di acquisto, sono associati, per la maggior parte, ad acquisti pessimi.

La correlazione lineare tra i prezzi MMR e il prezzo di acquisto del veicolo è più evidente se si considera la parte più densa dello Scatter Plot. Questo può essere fatto fissando una certa trasparenza dei punti, in modo che solo le zone più dense rimangano visibili, o mediante una funzione specifica che colori in modo diverso le zone con densità gaussiana maggiore.

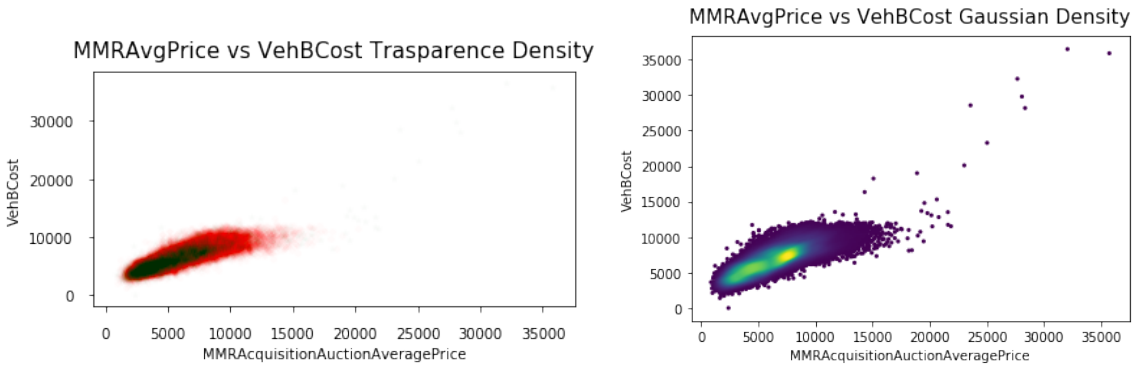


Figura 6: Scatter Plot di Densità dei prezzi dei veicoli

Oltre a quelli già visti un altro Scatter Plot interessante è quello della coppia di variabili "VehOdo" e "VehBCost", da cui si evince che veicoli con miglia percorse superiori a 100.000 sono stati acquistati all'asta a prezzi più bassi della media.

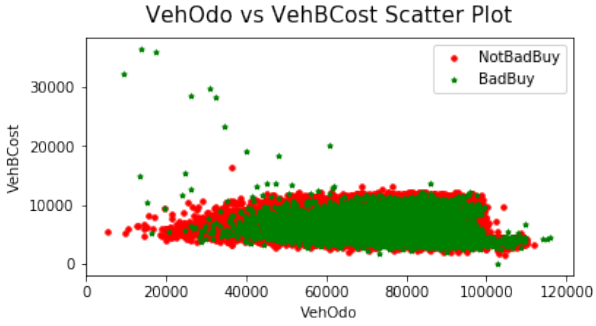


Figura 7: Scatter Plot VehBCost vs VehOdo

Alla luce di questi risultati sono stati eliminati tutti gli attributi dei prezzi MMR tranne uno, l'MMRAcquisitionAuctionAveragePrice, perché dopo aver calcolato le loro correlazioni e aver proiettato i loro valori sugli Scatter Plot è risultata una dipendenza lineare talmente significativa da rendere ridondante l'uso di tutti questi per le successive analisi. Inoltre per tutti gli MMR calcolati al tempo corrente non è stato possibile eseguire un'analisi approfondita poichè non è specificato il giorno in cui l'indice è stato calcolato (il dataset non è stato riempito

in un solo giorno, determinando quindi l'inconsistenza del MMR al tempo corrente, infatti non è specificato il giorno dell'inserimento del record).

1.5 Variables transformations

Sono stati generati Datasets con dati appositamente trasformati per ciascuna delle analisi successive e ne sono stati esportati i rispettivi files.

- **Trasformazioni per il Clustering:**
  E' stata creata un'ulteriore variabile, "CostOverOdo", usata per l'analisi, i cui valori sono dati dalla divisione VehBCost/VehOdo. Sono stati eliminati tutti gli attributi categorici. Gli attributi numerici definitivi su cui l'analisi è stata eseguita sono: "VehBCost", "MMRAcquisitionAuctionAveragePrice", "VehOdo", "WarrantyCost", "VehicleAge", "CostOverOdo". I valori di tutti questi attributi sono stati standardizzati al fine di renderne confrontabili tra loro le distanze.

- **Trasformazioni per la Association Rules Mining:**
  I valori dell'attributo "PurchDate" sono stati accorpati per mese anziché lasciarli divisi in giorni. L'attributo "RefId" è stato eliminato. I valori degli attributi numerici sono stati sostituiti dal rispettivo Bin di appartenenza. I Bins sono stati definiti dividendo il range di valori in un numero di intervalli uguali dato dalla formula di Sturges (17 Bins). A ciascun valore di tutti gli attributi è stata aggiunta una stringa costituita dal carattere Underscore più l'intestazione della colonna stessa, al fine di rendere riconducibile qualsiasi valore dei pattern al proprio attributo di appartenenza. Le stesse trasformazioni sono state eseguite anche sul Test Set per rendere possibili i prediction tasks mediante Rules Mining.

- **Trasformazioni per la Classification:**
  Sono stati eliminati, oltre a RefID, gli attributi Engine e Doors (ricavati dalle caratteristiche del modello) a causa della presenza di un numero troppo elevato di missing value. I valori dell'attributo "PurchDate" sono stati accorpati per mese anziché lasciati divisi in giorni. Anche in questo caso è stata creata la variabile "CostOverOdo" e inoltre è stata applicata la codifica onehot per tutti gli attributi categorici.

2 Clustering

2.1 Density-Based Clustering

L'analisi DBSCAN è stata eseguita su tutto il dataset fissando pari a 5 il valore del parametro "min samples" (o anche "Min Points"). Cioè per ogni punto del dataset preso in esame devono essere presenti almeno 5 punti (compreso il punto in esame) nell'intorno definito dall'EPS affinché il punto in esame stesso possa essere considerato un "Core Point". Il parametro "EPS", cioè la distanza massima entro la quale devono essere presenti il numero minimo di punti, è stato usato come threshold, facendolo variare per ottenere la miglior divisione di clusters possible. E' stato disegnato preventivamente il grafico delle distanze di ciascun punto del dataset dal suo quinto punto più vicino, ordinando i punti per valore crescente di questa

distanza. Il risultato, che è riportato qui sotto, è stato utile per individuare possibili livelli interessanti dell'EPS e immaginare il possibile esito dell'analisi.

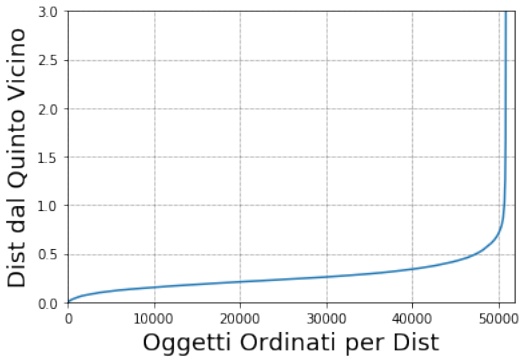


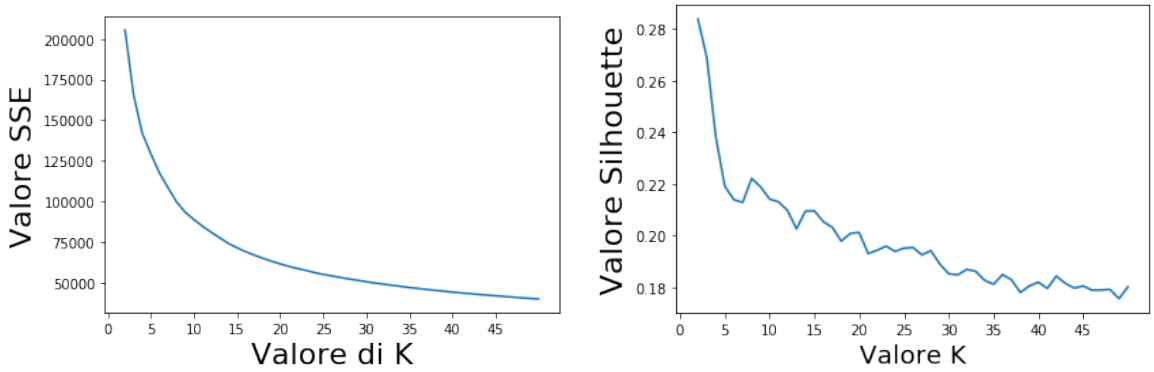
Figura 8: Distanze dal Quinto punto più vicino

Da questa raffigurazione infatti si nota come la maggior parte degli oggetti del dataset siano molto vicini tra loro, con distanze dal rispettivo quinto punto più vicino inferiori a 0,5. Pochi oggetti possono essere considerati considerevolmente distanti da tutti gli altri e non addensati tra loro. La clusterizzazione è stata eseguita per 20 iterazioni facendo variare in ciascuna di esse il parametro EPS da 0 a 2 con step di 0,1. Sono stati esclusi i risultati che portavano ad un numero di clusters superiore a 10, perché tali clusterizzazioni erano accomunate da un numero elevato di clusters contenenti pochissimi oggetti, che quindi non rappresentavano un risultato considerevole. Le rimanenti sono 13 clusterizzazioni con EPS da 0,8 a 2, delle quali è stato calcolato il coefficiente di "Silhouette".
Per scegliere il migliore tra questi risultati è stata adottata la seguente procedura di validazione: l'idea è che la miglior clusterizzazione possibile è quella il cui coefficiente di validazione si discosta di più dal coefficiente di validazione di un'analisi con identici parametri ma performata su un dataset con valori casuali. E' stato dunque generato un nuovo dataset di pari dimensioni di quello usato per l'analisi, i cui valori di ciascun attributo sono stati riempiti con valori numerici continui casuali, distribuiti secondo una Normale Standard (come per i valori del dataset analizzato). Su tale dataset sono state performate analisi DBSCAN con gli stessi insiemi di parametri usati per l'analisi principale e per ciscun risultato è stato calcolato il coefficiente di Silhouette, che è stato sottratto dal coefficiente di Silhouette dell'analisi principale con uguali parametri. Infine la miglior clusterizzazione è stata ritenuta quella per cui questa differenza nei coefficienti di validazione è risultata maggiore. In particolare il Clustering con più guadagno sul coefficiente di validazione del dataset random è quello con EPS = 1,3. Questo parametro aveva portato all'individuazione di due Clusters, che verranno analizzati nell'ultima sezione di questo capitolo, con un coefficiente di Silhouette dello 0.78 rispetto ad un coefficiente dello 0.34 del corrispondente Clustering sul dataset casuale.

2.2 K-Means Clustering

Per cominciare si è deciso di compiere diverse analisi K-means sull'intero database utilizzando differenti combinazioni delle variabili numeriche che si ritenevano più significative. Per stabilire il miglior valore di "K" si è utilizzato un piano cartesiano, nel quale sull'ascissa vengono rappresentati i valori che "K" può assumere, scegliendo come valore massimo 50, e sull'ordinata l'errore, Sum Square Error. Ogni punto rappresenta quindi una clusterizzazione con un valore di k ed il suo relativo errore, così facendo si forma una curva, che chiameremo SSECurv.

Una volta ottenuto il grafico si può notare che l'andamento è decrescente, in questa parte del grafico con un piccolo aumento del valore di "K" si ottiene un netto miglioramento dell'errore, questo finché la curva non cambia andamento. da quel punto in poi è meno conveniente incrementare "K", poiché l'errore ottiene un miglioramento marginale.



Si è quindi deciso di utilizzare sei variabili: VehicleAge, VehOdo, MMRAcquisitionAuctionAveragePrice, VehBCost, WarrantyCost e CostOverOdo, ottenendo così cluster in sei dimensioni. Vediamo ora i diversi passaggi di questa analisi. Per prima cosa si è cercato di estimare un intervallo di valori di "K" per effettuare le analisi, attraverso la SSE Curv, si è quindi scelto l'intervallo [2,15]. Ma come è possibile vedere dal grafico il cambio di andamento nella curva non è molto pronunciato, per questo si è deciso di verificare e restringere ulteriormente l'intervallo generando la Silhouette Curv, qui si può notare un picco positivo tra i valori 6 e 10 di "K", presumibilmente intorno al valore 8-9. Si decide così di restringere l'intervallo a [6,10].

Per ogni valore di "k" si effettua un clustering con numero massimo di iterazioni pari a 150, per poi scegliere il clustering con la miglior combinazione di minor errore e miglior Silhoutte, che corrisponde al picco visibile nella Silhouette Curv. Si sono così ottenuti 5 differenti clustering con SSE e Silhoutte nei seguenti domini:

Dominio SSE: [88670.23648138119 , 117353.92303693786];

Dominio Silhouette: [0.212505269706655 , 0.22194872680433517].

A questo punto possiamo affermare che quel picco corrisponde al valore di "K" pari a 8 con valori: SSE 99462.59055013148 che corrisponde alla mediana dei valori di SSE dei differenti clustering ottenuti e Silhouette 0.22194872680433517. Analizzeremo i Cluster ottenuti nell'aposita sezione.

2.3 Hierarchical Clustering

Una prima esecuzione dell'algoritmo gerarchico su tutto il dataset ha ottenuto risultati poco rilevanti. Anche impostando un numero elevato di cluster, l'algoritmo suddivideva il dataset in un cluster contenente la quasi totalità dei samples e gli altri cluster con poche decine di records. Per questo si è deciso di utilizzare come base per l'algoritmo il risultato del DBSCAN, ovvero il cluster privo degli outliers. In questo secondo approccio i risultati sono stati molto più incoraggianti. Nell'immagine 9 si possono notare i due dendogrammi ottenuti sia utilizzando

come metodo il completo che il ward e usando come metrica la distanza euclidea. Occorre sottolineare come il numero di variabili analizzate sia pari a 6, le stesse utilizzate anche dal K-means.

Come già utilizzato per l'algoritmo del DBSCAN, la valutazione della qualità dei cluster ottenuti è stata determinata dal valore della silhouette a cui è stato sottratto il valore della silhouette calcolato su un dataset con gli stessi parametri ma valori casuali, distribuiti secondo una normale standard. Come si può facilmente vedere dai dendogrammi, il numero di cluster ottimale è compreso tra 3 e 6, per questo le successive analisi si concentreranno su questi valori. Occorre precisare che nel codice sono anche stati analizzati i metodi average e single, ma che sono stati scartati perchè: il primo otteneva valori di silhouette molto più bassi rispetto agli altri approcci (la differenza con il dataset casuale si aggira intorno ai 0.09, rispetto ai 0.35 degli altri metodi); il secondo otteneva delle suddivisioni in cluster non rilevanti (tutte le prove hanno generato un singolo cluster contenente la quasi totalità dei samles e gli altri composti al massimo da 3 samples).

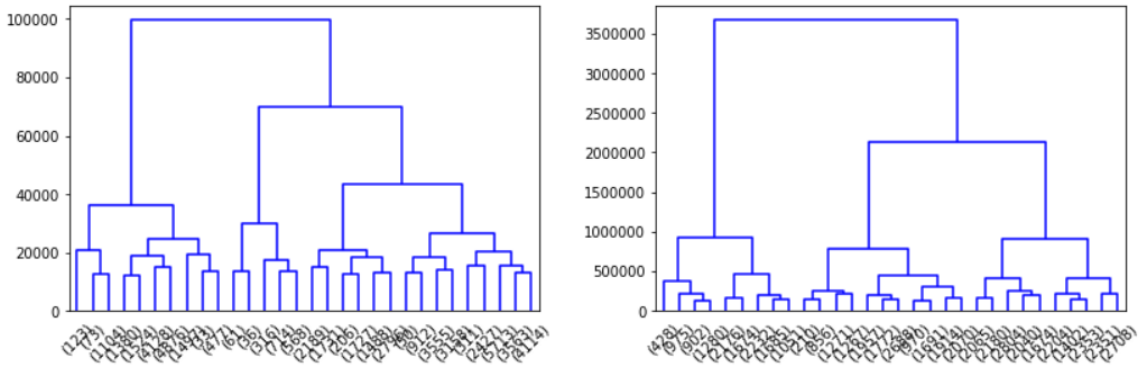


Figura 9: Dendogramma con il metodo complete sulla sinistra e col metodo ward sulla destra

Nella tabella 2 sono indicate le migliori 5 suddivisioni in cluster e i rispettivi valori di silhouette. Con Validation si intende il valore ottenuto dalla differenza tra le silhouette. La tabella mostra come il migliore approccio sia la suddivisione in 4 cluster dell dataset, utilizzando il metodo complete.

Top	N cluster	Metodo	Silhouette	Validation
1	4	complete	0.4167	0.4357
2	5	complete	0.3663	0.3801
3	5	ward	0.3385	0.3523
4	6	ward	0.3318	0.3379
5	3	complete	0.3273	0.3307

Tabella 2: Top 5 clusterizzazioni

2.4 Results Analysis

Il risultato dell'analisi DBSCAN divide il dataset in due clusters: un cluster molto denso che comprende 50.850 oggetti (quasi tutti gli oggetti del dataset) ed un cluster poco denso di soli 56 oggetti. Si riportano qui i grafici degli Scatter Plot dei clusters risultanti per gli attributi principali.

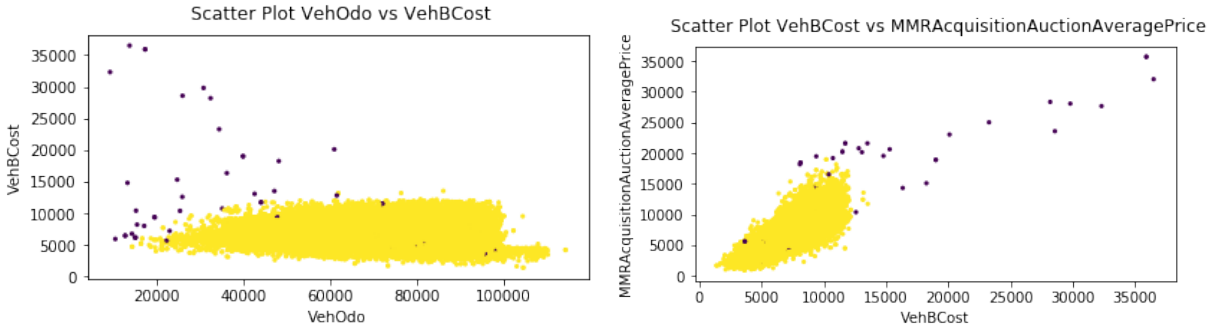


Figura 10: Scatter Plot Risultati DBSCAN

Gli oggetti del cluster denso hanno valori degli attributi numerici coerenti con i valori generalmente presenti nel dataset. Gli oggetti del cluster non denso invece hanno valori non usuali, che corrispondono a veicoli con prezzo elevato e miglia basse: VehBCost superiore a 12.500, MMR Price superiore a 17.500 e VehOdo inferiore a 20.000. A questi oggetti l'analisi ha assegnato label -1, stante a rappresentare oggetti che potrebbero disturbare le altre analisi. Per questo motivo questi oggetti sono stati eliminati nel performare le clusterizzazioni successive. Il risultato più interessante di questa clusterizzazione riguarda la classe dei Bad Buy: dei 56 oggetti nel cluster poco denso 29 hanno valore dell'attributo "IsBadBuy" uguale a 1. Questo significa che circa il 51% dei veicoli con le caratteristiche sopra citate hanno rappresentato acquisti non positivi (in particolare i veicoli con prezzo elevato), rispetto ad un 12,6% di Bad Buy sull'intero dataset.

Pe quanto riguarda K-Means i cluster ottenuti sono 8, composti come segue:

0	1	2	3	4	5	6	7
8333	5350	479	6427	6393	8722	6154	8992

Osserviamo ora invece una rappresentazione grafica dei valori dei centroidi dei differenti cluster, che si è deciso di rappresentare ancora normalizzati vista la grande differenza dei valori assunti dalle diverse variabili:

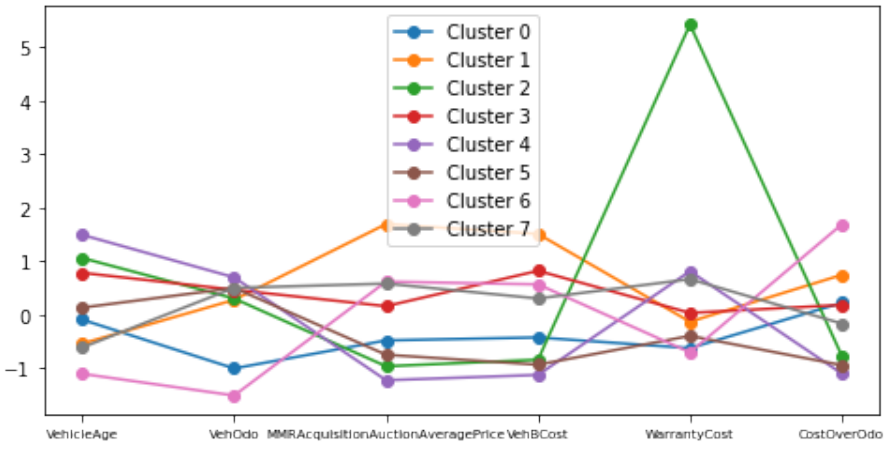
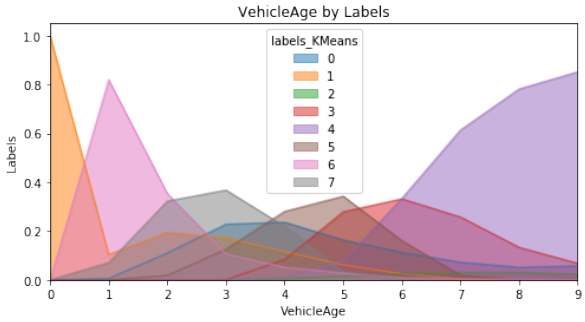


Figura 11: Centroidi K-means Clustering.

Dalla composizione dei cluster si evince che il Cluster 2 è composto da soltanto 479 oggetti, numero discretamente inferiore rispetto alla composizione degli altri cluster. Andando a controllare i centroidi del suddetto cluster si può notare che presenta dei valori decisamenti elevanti per quanto concerne la variabile WarrantyCost, si intuisce quindi che questi oggetti possono rappresentare degli outliers per la variabile stessa.
I differenti cluster sono stati analizzati con scatterplot a due dimensioni con combinazioni di tutte le variabili, crosstab e la loro rappresentazione grafica attraverso dei Bar chart. Da queste analisi non risalta nulla di particolare poiché all'interno dei cluster sono presenti tutti i valori, o quasi, delle variabili del database.



Ma osservando i grafici concernenti la variabile VehicleAge possiamo notare che vi è una certa distribuzione all'interno dei cluster. Come è possibile eccepire dalla tabella 3 e dal grafico accanto riportato, ogni cluster è così composto: il numero di oggetti parte da un valore x di VehicleAge con una piccola quantità di oggetti, ma al valore x+1 la quantità di oggetti comincia ad incrementare, questo accade per il valore x+2 e così via fino ad x+n, per poi raggiungere il valore $y > x + n$ nel quale si ha il più alto numero di oggetti(che sono rappresentati dai picchi positivi visibili nel grafico), per poi iniziare a decrescere all'aumentare di y. Prendiamo ad esempio il cluster 3 che ha 3 oggetti con valore 3 di VehicleAge, questi oggetti incrementano fino a 2473 al valore 5 per poi decrescere fino ad avere 32 oggetti per il valore 9. Possiamo quindi affermare che la formazione dei cluster sia influenzata maggiormente dai valore che VehicleAge assume anche se non si ha una netta separazione dei valori all'interno dei cluster, questo probabilmente accade perché il database è molto omogeneo, in più è possibile che siano ancora presenti outliers che disturbano l'algoritmo e quindi la formazione dei cluster.

K-Means Cluster VehicleAge	0	1	2	3	4	5	6	7
0	0	1	0	0	0	0	0	0
1	14	244	0	0	0	0	1917	166
2	674	1194	0	0	0	113	2166	1977
3	2488	1878	0	3	0	1369	1159	4010
4	2716	1334	51	962	101	3226	594	2520
5	1450	545	127	2473	680	3039	251	303
6	639	142	137	1879	1887	909	52	16
7	241	12	103	861	2051	66	10	0
8	84	0	50	217	1272	0	5	0
9	27	0	11	32	402	0	0	0

Tabella 3: Composizione valori VehicleAge all'interno della K-Means Clustering.

Per quanto riguarda l'algoritmo gerarchico, i cluster ottenuti sono 4 e sono caratterizzati dal seguente numero di samples: 15215, 23803, 1695, 10137. Anche in questo caso, come per il K-means, la variabile più interessante risulta essere VehicleAge. Come si può notare dall'immagine 12, il cluster identificato con la label 3 è composto per la maggior parte da

automobili recenti (principalmente con uno, due o tre anni di utilizzo). Come già anticipato, la mancanza di separazioni nette per altre variabili categoriche è dovuta alla presenza nel dataset di samples molto omogenei. Per concludere, a livello pratico, l'approccio per il clustering che ha ottenuto i migliori risultati, e quindi la migliore separazione in cluster, è stato il K-means. E' infatti quest'ultimo l'approccio in cui, analizzando le vetture presenti nei diversi clusters, sono state trovate maggiori informazioni sulla distribuzione delle macchine.

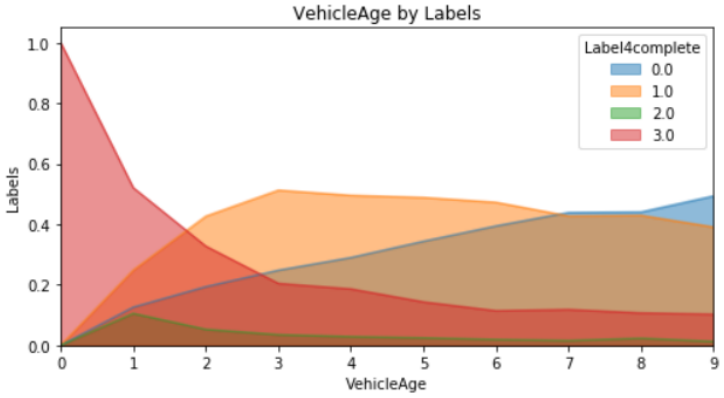


Figura 12: Distribuzione della variabile VehicleAge in base ai cluster

3 Association Rules Mining

L'estrazione di Pattern è stata eseguita sia sull'intero dataset trasformato che su datasets composti soltanto da attributi categorici e numerici. Sono stati estratti in due fasi distinte i Pattern Frequenti, cioè quegli Itemsets che si ripresentano nel dataset con una probabilità assoluta superiore ad un livello minimo (il "Minimum Support". es. Pattern xy: $\Pr(xy) >= Min\ Supp$) e le Regole Associative, cioè le associazioni di Itemsets di ciascun Pattern Frequente con una probabilità condizionata superiore ad un ulteriore livello minimo (il "Minimum Confidence". es. rule x -> y: $\Pr(y \mid x) >= Min\ Conf$). Infine sono stati analizzati i risultati più interessanti.

3.1 Frequent Patterns

Prima dell'estrazione vera e propria dei Frequent Patterns è stato verificato come varia il numero di Itemsets Frequenti al variare del livello di *Min Supp* dall'1% al 100%, con step dell'1%, per Itamsets composti da 2 fino a 4 Items e appertenenti alle tre tipologie: "All Itemsets", sono tutti i possibili pattern frequenti che rispettano il limite di Support definito; "Closed Itemsets", sono tutti i pattern frequenti i quali sovrainsiemi sono meno frequenti di essi (vengono cioè esclusi solo i sottoinsiemi frequenti tanto quanto i Closed Itemsets); "Maximal Itemsets", sono tutti i pattern frequenti i quali sovrainsiemi sono pattern che non rispettano i limiti di Support (cioè sono i pattern più ampi possibili immediatamente precedenti a quelli non frequenti). Rispetto al caso di tutti i possibili pattern frequenti gli altri due ci permettono di individuare gli Itemsets con più Items possibili che rispettano i limiti di Support. I Closed Itemsets in particolare sono interessanti perchè rappresentano i pattern più ampi possibili per ogni livello di Support e per questo sono stati usati per l'estrazione dei pattern.
Di seguito si riportano i grafici del numero di Patterns nei tre casi per livello di Support dall'1% al 30%.

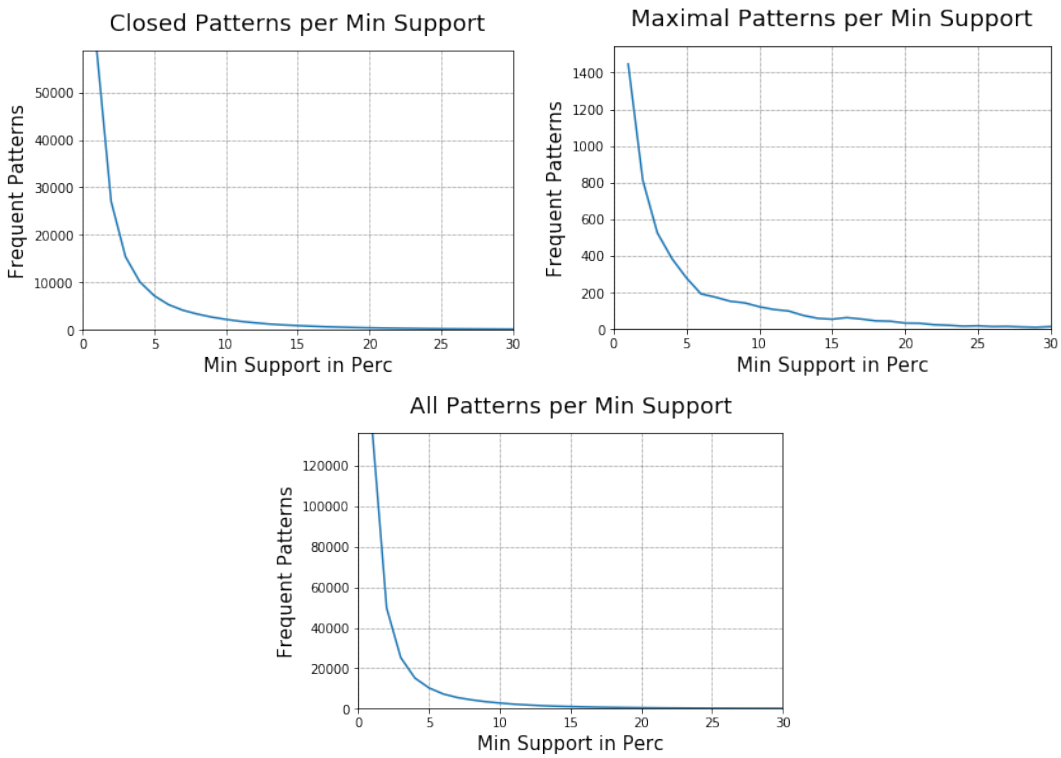


Figura 13: Numero di Frequent Patterns per livello di *Min Supp*

In particolare con *Min Supp* = 1% si hanno 136.340 Frequent Patterns totali, 58.758 Closed Patterns e 1.446 Maximal Patterns. L'azzeramento del numero di pattern in tutti e tre i casi si ha con *Min Supp* = 94%. La riduzione più rapida del numero di patterns si ha quando il *Min Supp* passa dall'1% al 5%, successivamente la curva si appiattisce. Questo significa che fissando il livello di *Min Supp* pari al 5% troviamo pressochè gli stessi pattern che troviamo con threshold maggiori. Questi rappresentano i pattern più frequenti in assoluto, che risultano però spesso banali essendo riferiti a combinazioni di Items già di per se molto frequenti nel dataset. Scegliendo una soglia inferiore al 5% invece emergono sempre più pattern diversi che, nonostante la minor probabilità assoluta, possono risultare più interessanti in termini di probabilità realtiva (per estrarne delle Rules significative).

L'estrazione effettiva dei patterns quindi è stata eseguita prendendo in considerazione i Closed Itemsets e usando un livello di *Min Supp* dell'1% (corrispondente ad una frequenza assoluta minima di 510 oggetti). Sul Dataset totale questi parametri portano ad ottenere 404.153 Frequent Patterns. I pattern così trovati sono stati ordinati per Support decrescente per mettere in evidenza i più frequenti. I 42 patterns più frequenti in assoluto hanno un Support superiore al 50% e sono quelli con valore "AUTO" per l'attributo "Transmission", "0" per l'attributo "IsOnlineSale", "0" per l'attributo "IsBadBuy", "AMERICAN" per l'attributo "Nationality", "4D" per l'attributo "Doors", "SEDAN" per l'attributo "SubModel", "1.0" per l'attributo "WheelTypeID" e "MANHEIM" per l'attributo "Auction". Alcuni esempi sono: ('Transmission = AUTO', 'IsOnlineSale = 0') con Support del 93,83%, pari a 47.766 oggetti del Dataset, è il pattern più frequente in assoluto; ('IsBadBuy = 0', 'Transmission = AUTO', 'IsOnlineSale = 0') con Support dell'82,05%, è il pattern di tre Items più frequente in assoluto; ('Nationality' = AMERICAN, 'IsBadBuy' = 0, 'Transmission'= AUTO, 'IsOnlineSale' = 0) con Support del

68.24%, è il pattern di quattro Items più frequente in assoluto. Come si nota però i pattern descritti definiscono il modello di auto più comune in assoluto nel dataset: una berlina con 4 porte e cerchi in lega prodotta da una casa americana che è stata aquistata direttamente all'asta della casa Manheim e che non ha costituito un pessimo acquisto.
Patterns decisamente più interessanti cominciano a presentarsi in corrispondenza di Support inferiori al 5%, che sono 387.392. La differenza, di soli 16.761 Patterns rispetto al totale, è rappresentata per la quasi totalità dai pattern contenenti uno dei valori banali degli attributi riportati sopra. Analizzando in particolare la situazione del valore dell'attributo "IsBadBuy" uguale ad 1 si contano 2970 Patterns totali e 46 composti da 2 Items, dei quali 27 con in coppia un attributo categorico e 19 con in coppia un attributo numerico.
Tra quelli con in coppia un attributo categorico si riportano: ('IsBadBuy = 1', 'Trim = BAS') con Support del 3.06%; ('IsBadBuy = 1', 'VehicleAge = 5') con Support del 2.58%; ('IsBadBuy = 1', 'VNST = TX') con Support del 2.55%; ('IsBadBuy = 1', 'VehicleAge = 4') con Support del 2.46%; ('IsBadBuy = 1', 'VehicleAge = 6') con Support del 2.00%; ('IsBadBuy = 1', 'VehicleAge = 3') con Support del 1.76%; ('IsBadBuy = 1', 'Size = COMPACT') con Support dell'1.75%; ('IsBadBuy = 1', 'Trim = SE') con Support dell'1.64%; ('IsBadBuy= 1', 'VNST = FL') con Support dell'1.58; ('Size = MEDIUM SUV', 'IsBadBuy = 1') con Support dell'1.55%;
Tra quelli con in coppia un attributo numerico si riportano: ('IsBadBuy = 1', 'VehBCostBin = [3649.4, 5473.6)') con Support del 4.43%; ('IsBadBuy = 1', 'MMRAcquisitionAuctionAveragePriceBin = [2625.9, 4367.8)') con Support del 3.91%; ('IsBadBuy = 1', 'VehBCostBin = [5473.6, 7297.8)') con Support del 3.86%; ('IsBadBuy = 1', 'WarrantyCostBin = [813.8, 1165.6)') con Support del 3.32%; 'MMRAcquisitionAuctionAveragePriceBin = [4367.8, 6109.7)') con Support del 3.05%; ('IsBadBuy = 1', 'WarrantyCostBin = [1165.6, 1517.4)') con Support del 2.87%.

3.2 Association Rules

Anche in questo caso è stato verificato come varia il numero di Association Rules al variare del livello di *Min Conf* dal 10% al 100%, con step del 2%, per Itamsets composti da 2 a 4 Items e *Min Supp* dell'1%. Se ne riporta la rappresentazione grafica.

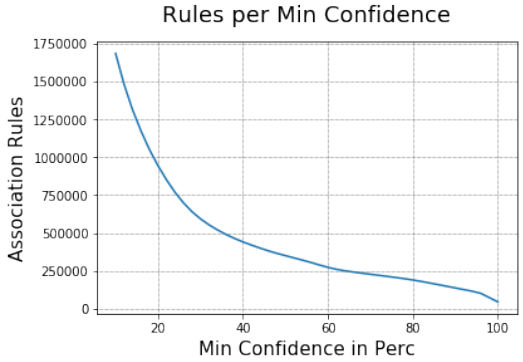


Figura 14: Numero di Association Rules per livello di *Min Conf*

In questo caso la curva passa da 1.684.485 Rules con un *Min Conf* del 10% ad un minimo di 45.408 Rules qando il *Min Conf* è pari al 100%. L'inclinazione elevata fino alla soglia del 60% suggerisce la selezione di un threshold maggiore che isoli regole più stringenti. Successivamente la curva risulta sostanzialmente piatta per soglie dal 60% all'80%, da dove si ha

un'ulteriore accelerazione nella decrescita del numero di Rules. Ciò significa che l'estrazione delle regole con una soglia inferiore all'80% non porterebbe ad un significativo incremento del numero di Rules e soglie superiori eliminerebbero rules interessanti.

L'estrazione effettiva delle Rules dunque è stata eseguita usando un livello di *Min Conf* dell'80% e di *Min Supp* dello 0,01% (corrispondente ad una frequenza assoluta minima di 5 oggetti), per Itemsets composti da 2 a 4 Items. Il motivo di un livello di Support Minimo così basso è che solo su questi livelli si trovano Rules con la variabile target tra le "consequence". Sul Dataset totale questi parametri portano ad ottenere 10.945.398 Association Rules. Le migliori Rules sono quelle con un Support percentuale inferiore al 5% (per scartare le Rules banali) e Lift Superiore a 2, che sono 4.597.326. Tra queste la maggior parte riguardano modelli specifici di veicoli che si presentano in associazione a caratteristiche peculiari di essi. Le più interessanti per gli scopi dell'analisi sono le 339 Rules contenenti il valore 1 dell'attributo "IsBadBuy" come consequence della Rule, che sono state ordinate per Support decrescente.
Si riportano le seguenti contenenti esclusivamente attributi categorici: ('VNZIP1 = 74135', 'PurchYearMonth = 2010-06', 'WheelTypeID = 1.0') –> 'IsBadBuy = 1', con Support dello 0.033%, Confidence del 100% e Lift pari a 7.98; ('PurchYearMonth = 2010-06', 'BYRNO = 99761', 'WheelTypeID = 1.0') –> 'IsBadBuy = 1', con Support dello 0.033%, Confidence del 100% e Lift pari a 7.98; ('VNST = NV', 'Engine = 4.0L', 'WheelTypeID = 1.0') –> 'IsBadBuy = 1', con Support dello 0.018%, Confidence del 90% e Lift pari a 7.18; ('VNST = NV', 'Size = MEDIUM SUV', 'TopThreeAmericanName = FORD') –> 'IsBadBuy = 1', con Support dello 0.018%, Confidence dell'81.81% e Lift pari a 6.53; ('Trim = XLS', 'PurchYearMonth = 2010-04', 'Size = MEDIUM SUV') –> 'IsBadBuy = 1', con Support dello 0.018%, Confidence dell'81.81% e Lift pari a 6.53; ('Model = EXPLORER', 'Trim = XLS', 'PurchYearMonth = 2010-04') –> 'IsBadBuy = 1', con Support dello 0.016%, Confidence dell'88.89% e Lift pari a 7.09;
Le seguenti contenenti esclusivamente attributi numerici: ('MMRAcquisitionAuctionAveragePriceBin = [20044.9, 21786.8)') –> 'IsBadBuy = 1', con Support dello 0.009%, Confidence dell'83.33% e Lift pari a 6.65; ('MMRAcquisitionAuctionAveragePriceBin = [18303.0, 20044.9)', 'WarrantyCostBin = [462.0, 813.8)') –> 'IsBadBuy = 1', con Support dello 0.009%, Confidence dell'83.33% e Lift pari a 6.65.
E le seguenti relative a combinazioni di attributi sia categorici che numerici: ('VNZIP1 = 78754', 'VehBCostBin = [1825.2, 3649.4)', 'Size = MEDIUM') –> 'IsBadBuy = 1', con Support dello 0.023%, Confidence dell'80% e Lift pari a 6.39; ('VehicleAge = 9', 'VehBCostBin = [1825.2, 3649.4)', 'Auction = ADESA') –> 'IsBadBuy = 1', con Support dello 0.022%, Confidence dell'84.61% e Lift pari a 6.75; ('BYRNO = 25100', 'VehBCostBin = [1825.2, 3649.4)', 'Size = MEDIUM') –> 'IsBadBuy = 1', con Support dello 0.02%, Confidence dell'83.33% e Lift pari a 6.65.

3.3 Results Analysis

Le 339 Rules così estratte sono state usate tutte per eseguire la Classification della variabile target "IsBadBuy". E' stata aggiunta al dataset una colonna che esprime la predizione del valore 1 della variabile target, associandolo ad ogni record del dataset che contiene gli stessi valori delgi attributi presenti in almeno una delle 339 Rules con valore "IsBadBuy = 1" tra le Consequence. Il resto dei valori sono stati riempiti con 0, sia per il Training Set che per il Test Set. La predizione è poi stata valutata in termini di Positive Accuracy, Positive Precision e Positive Recall (la predizione del valore 0 della variabile target non è interessante). Si riportano

dunque i seguenti coefficienti per il Training Set: Accuracy = 88%, Precision = 81%, Recall = 11%; e i seguenti coefficienti per il Test Set: Accuracy = 87%, Precision = 25%, Recall = 2%.

4 Classification

4.1 Learning of decision tree

Inizialmente ci si è concentrati sulla massimizzazione delle performance utilizzando come metro di valutazione l'accuratezza del modello. Con modello intendiamo un albero decisionale binario allenato su un training set e validato utilizzando apposite tecniche di validation. Nonostante il risultato ottenuto potesse sembrare ottimo, il valore di recall per l'attributo IsBadBuy, con valore 1, era pari al 3%. Alla luce di questi dati si è preferito perdere un po' di accuratezza del modello, per riuscire ad aumentare il valore di recall. Si è quindi deciso di minimizzare il più possibile il numero di falsi positivi rispetto ai falsi negativi. E' infatti intuibile come sia da considerare un errore più grave comprare un veicolo che risulterà essere un cattivo acquisto, rispetto al non acquisto di un veicolo poichè il modello erroneamente lo determina come un pessimo acquisto. Si è quindi optato per utilizzare il recall come metro di giudizio delle performance del modello.

Per poter effettuare più prove possibili si è deciso, oltre all'utilizzo di diversi valori per i parametri (tutti i valori testati sono mostrati in tabella 4) di eseguire tutte i test sia con la funzione di guadagno GINI che con l'ENTROPIA. A livello pratico gli alberi decisionali ottenuti utilizzando come gain function l'entropia si sono rilevati più performanti.

Gli hyperparameter utilizzati sono i seguenti:

- min_samples_split, il numero di samples minimo richiesto per eseguire uno split in un nodo interno;
- min_samples_leaf, il numero minimo di samples richiesto a un nodo per essere un nodo foglia;
- max_depth, che indica la profondità massima dell'albero, in questo caso settata a None, ovvero non è stato posto nessun vincolo alla profondità;
- Per quanto riguarda gli altri parametri della funzione $DecisionTreeClassifier$ sono stati utilizzati i loro valori di default.

max_depth	None
min_samples_split	2, 5, 10, 15, 20, 50, 100, 150, 200, 500
min_samples_leaf	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 50, 100

Tabella 4: Tabella dei possibili valori assegnati agli attributi

4.2 Decision tree interpretation

Questa sottosezione si concentrerà esclusivamente sull'interpretazione dell'albero decisionale finale e quindi sull'importanza che ogni singolo attributo ha nel determinare la previsione del valore di IsBadBuy. Una prima analisi, senza controllare nello specifico l'albero, ha evidenziato come 'CostOverOdo' sia l'attributo con maggiore peso per la previsione finale (13.08%),

seguito, in ordine, da 'MMRAcquisitionAuctionAveragePrice' (12.25%), 'VehOdo' (12.25%) e 'VehBCost' (12.13%). Questo risultato non deve stupire poichè le miglie percorse, il prezzo pagato e l'indice MMR sono tutte variabili oggettivamente molto rilevanti per poter determinare un buon acquisto.
Passiamo adesso ad osservare l'albero generato e in maniera particolari ai nodi che ci portano alla prima foglia disponibile, che è situata a profondità 5 dell'albero. Prima di tutto, come si era precedentemente detto, uno degli attributo più importante per quanto concerne la distribuzione della variabile IsBadBuy è VehiclAge, infatti rappresenta il primo nodo dell'albero decisionale con un'entropia pari a 0.544, che splitta i dati a seconda che essi abbiano un valore minore o uguale a 4.5. Nel caso in cui la condizione è verificata (True) ci troveremo in presenza di 30622 oggetti, a questo punto si verificherà la variabile WheelTypeID con un valore minore uguale di 1.5, si prosegue per il valore True con 13195 oggetti con variabile Auction1 ≤ 0.5 (ovvero se la casa d'asta non è MANHEIM, poichè stiamo utilizzando la codifica onehot) con entropia pari a 0.545. Avanzando sul ramo False ci imbattiamo in 7277 osservazioni che assumono un valore ≥ 0.5 (la casa d'asta è MANHEIM), la variabile che viene presa in considerazione in questo nodo è MMRAcquisitionAuctionAveragePrice ≤ 17212 con entropia pari a 0.43. Procedendo per il ramo False troviamo solo 18 oggetti che vengono divisi per il valore di CostOverOdo≤ 0.184 ottenendo un'entropia di 0.503: True $\rightarrow$ entropia 0.918, 2 osservazioni con valore 0 di IsBadBuy ed uno con valore 1, quindi questa foglia assume la classe "0"; False $\rightarrow$ entropia 0, nella quale tutti e 15 gli oggetti assumono valore 1 che è la classe della foglia stessa.
Qui di seguito sono mostrati nodi e rami dell'albero ottenuto con profondità pari a 3.

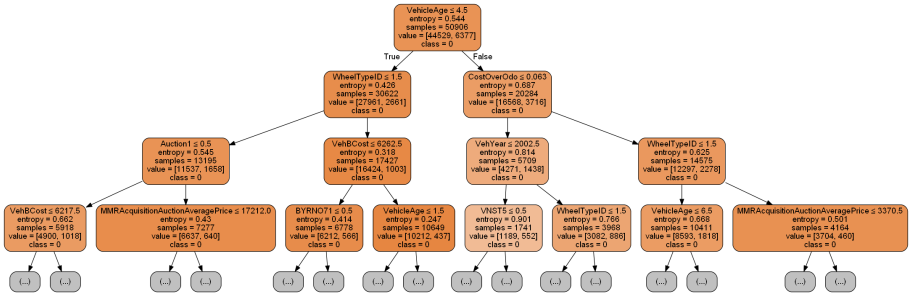


Figura 15: Albero decisionale con profondità 3

4.3 Decision tree validation

Per la ricerca dei migliori hyperparameter è stata utilizzata la funzione *RandomizedSearchCV*, su tutto il training set, che implementa al suo interno come metodo di validazione la CrossValidation. Una volta ottenuti gli hyperparameter ottimali (nel nostro caso abbiamo preso i migliori 4) sono stati costruiti i corrispettivi modelli utilizzando come validazione la tecnica dell'HoldOut, ovvero utilizzando solo il 70% del training set per allenare il modello e il restante 30% è stato utilizzato come validation set.

4.4 The best prediction model

Per la selezione del miglior modello è stata eseguita la funzione *RandomizedSearchCV*. Nella tabella 5 sono indicate le 10 migliori combinazioni di hyperparameter in funzione al valore di recall ottenuto. Le prime 4 combinazioni sono state analizzate più nello specifico per determinare la migliore. Per ognuna di esse è stato costruito un apposito modello (utilizzando

il 30% del training set come validation set) per poter analizzare nello specifico i valori di accuracy, recall, precision e f1 score. I risultati sono riportati nella tabella 6.

Il modello generato dalla prima combinazione tra le migliori 4 ha ottenuto i valori maggiori sia per il recall che per la precision (0.22, 0.21), nonostante i valori siano molto simili a quelli ottenuti dagli altri alberi decisionali, per questo la scelta degli hyperparameter migliori è ricaduta su quelli del primo modello.

Nella tabella 7 vengono infine mostrati i valori finali ottenuti sul test set (i missing value sono stati riempiti allo stesso modo del training set), allenando il modello su tutto il training set, utilizzando la migliore combinazinoe di hyperparameter.

Come si può vedere il modello finale è caratterizzato da un'accuracy dell'80% e da un recall pari al 21%, questo indica che più o meno il 20% delle macchine considerate come pessimi acquisti verranno individuate correttamente dall'albero decisionale.

top	gain function	min_samples_split	min_samples_leaf	Validation_score
1	entropia	2	3	0.193
2	entropia	2	1	0.192
3	entropia	5	3	0.190
4	entropia	10	1	0.187
5	entropia	10	3	0.185
6	entropia	15	5	0.179
7	entropia	5	4	0.179
8	entropia	5	5	0.179
9	gini	2	1	0.178
10	entropia	2	5	0.178

Tabella 5: Top 10 delle combinazioni degli Hyperparameter

	valore di IsBadBuy	accuracy test	precision	recall	f1 score	accuracy training
1	0	0.8046	0.89	0.89	0.89	0.9550
	1		0.22	0.21	0.22	
2	0	0.8032	0.89	0.89	0.89	1
	1		0.21	0.21	0.21	
3	0	0.8024	0.89	0.89	0.89	0.9546
	1		0.21	0.21	0.21	
4	0	0.8067	0.89	0.89	0.89	0.9578
	1		0.21	0.20	0.21	

Tabella 6: Valore di precision accuracy recall e f1 score per i top 4 model

valore di IsBadBuy	accuracy test	precision	recall	f1 score	accuracy training
0	0.8056	0.89	0.89	0.89	0.9544
1		0.20	0.21	0.21	

Tabella 7: Risultati modello finale