PROJECT WORK
DATA MINING 2019/20

Stefano Berti (582945), Andrea Bongiorno (517437),
Marco Ciompi (537856), Gianmarco Di Mauro (58795)

1. INTRODUZIONE

Il presente elaborato descrive l'analisi condotta sul dataset "*Do not Get Kicked!",* realizzato dalla start-up Carvana e contenente oltre 70 mila records, relativi alla compravendita di auto all'asta. L' obiettivo dell'analisi è quello di prevenire il rischio di un cattivo acquisto, a partire da oltre 30 attributi relativi a veicoli usati.

L'analisi si articola in cinque fasi.
La prima fase è stata dedicata all'esplorazione e comprensione del dataset mediante l'utilizzo di differenti indicatori statistici e ricerche, ovvero *Data Understanding*. Nella successiva fase di *Data Preparation* il dataset è stato manipolato, corretto e arricchito in modo tale da renderlo utilizzabile per le successive fasi di analisi, che riguardano: *Clustering, Pattern Mining e Classificazione.*

2. DATA UNDERSTANDING

2.1 Descrizione del dataset

Il dataset Carvana pubblicato viene diviso in due parti: *training set* e *test*, con un rapporto sul totale dei dati di 60:40. Il training set è composto da 58386 records, ognuno descritto da 34 attributi di diversa natura. Gli attributi sono stati catalogati come descritto dalla seguente tabella:

TIPOLOGIA		ATTRIBUTO
CATEGORICO	NOMINALE	Auction, Make, Model, Trim, SubModel, Color, Nationality, TopThreeAmericanName, AUCGUART, VNST, WheelTypeID, BYRNO, VNZIP1
	ORDINALE	WheelType, Size, VehYear, VehicleAge
	BINARIO	Transmission, PRIMEUNIT, IsBadBuy, IsOnlineSale
NUMERICO	CONTINUO	VehOdo, MMRAcquisitionAuctionAveragePrice, MMRAcquisitionAuctionCleanPrice, MMRAcquisitionRetailAveragePrice, MMRAcquisitonRetailCleanPrice, MMRCurrentAuctionAveragePrice, MMRCurrentAuctionCleanPrice, MMRCurrentRetailAveragePrice, MMRCurrentRetailCleanPrice, VehBCost, WarrantyCost
	DATE	PurchDate

Di seguito forniamo una breve descrizione per ogni attributo:

- *Auction*: casa d'aste per la vendita del veicolo;
- *Make, Model, SubModel, Transmission, Trim, Size, Color:* rispettivamente produttore, modello, sotto-modello, cambio (manuale o automatico), livello di equipaggiamento, dimensione e colore del veicolo;
- *TopThreeAmericanName*: indica seil produttore del veicolo appartiene ad una delle 3 principali case madri americane e specifica a quale, "altro" altrimenti;
- *VNST, VNZIP:* sigla e codice postale dello stato americano in cui il veicolo è stato battuto all'asta;
- *WheelType, WheelTypeID*: informazioni sul cerchione del veicolo (in lega, speciale o copri-cerchio) e relativo identificativo numerico (1, 2, 3);
- *VehOdo, VehBCost, WarrantyCost*: rispettivamente il chilometraggio del motore, il prezzo pagato alla casa d'asta e il costo della garanzia di un veicolo entro i 36 mesi o le 36 mila miglia;
- *AUCGUART, PRIMEUNIT*: rispettivamente il rischio di acquisto del veicolo (il livello di garanzie assicurate dal venditore) e un booleano che indica se il veicolo ha una elevata domanda di mercato;
- *IsBadBuy*: la variabile dipendente che indica se l'acquisto è stato cattivo o meno;
- *PurchDate*: data di acquisto all'asta del veicolo;

- *IsOnlineSale:* indica se il veicolo è stato originariamente acquistato all'asta o meno.
- *MMR*s:* gli attributi che iniziano con MMR sono delle stime effettuate da un ente chiamato Manheim ed indicano il prezzo ideale di un veicolo in base alle sue condizioni (Average o Clean), al momento del primo acquisto o al prezzo di mercato attuale (Acquistion o Current) e se comprato al dettaglio o all' asta (Retail o Auction). Per esempio MMRCurrentAuctionAveragePrice indica il prezzo (ideale) attuale di un veicolo comprato all'asta in condizioni "medie" di usura.

Notiamo che gli attributi *KickDate* e *AcquisitionType* menzionati nella documentazione che accompagna il dataset non sono effettivamente presenti nel dataset stesso.

2.2 Accuratezza sintattica

Con *"accuratezza aintattica"* ci riferiamo alla presenza di errori sintattici nel dataset, intesi principalmente come refusi.

Dalle nostre verifiche sono emersi errori negli attributi *Trim, Model* e *SubModel.* In ognuno di questi casi abbiamo trovato:

- Valori identici ma scritti in modo diverso (es: Pacifica vs PACIFICA);
- Caratteri mancanti (es: SE- vs SE-R);
- Caratteri non validi al termine di una stringa (es: /).

Molti di questi errori sono stati corretti mediante l'utilizzo di espressioni regolari in Python, ma non escludiamo l'esistenza di errori simili per questi 3 attributi che non sono facilmente ispezionabili a causa dell'elevato range di valori possibili. L'individuazione di tutti gli errori avrebbe richiesto tempistiche eccessive.

Oltre alla presenza di refusi, abbiamo notato anche la presenza di informazioni aggiuntive oltre a quelle che l'attributo avrebbe dovuto descrivere.
In particolare abbiamo notato che l'attributo *Model* spesso conteneva informazioni non strettamente collegate con il modello del veicolo come:

- Tipo di iniezione del carburante: EFI, MPI, SFI;
- Numero di ruote motrici: AWD, 2WD, 4WD, FWD.

Per gestire questo "rumore" abbiamo utilizzato nuovamente delle espressioni regolari, questa volta al fine di derivare una nuova colonna chiamata ModelSimple, contenente solo il nome del modello del veicolo senza informazioni aggiuntive. Analogamente siamo riusciti ad estrarre dall'attributo *SubModel* il numero di porte di quasi ogni veicolo, arricchendo così il dataset.
Abbiamo provato ad aggiungere al dataset informazioni sull'iniezione e sulla trazione ma queste erano spesso mancanti o incomplete e abbiamo quindi deciso di scartarle.

Abbiamo ritenuto non validi i valori OTHER ASIAN e TOP LINE ASIAN per l'attributo *Nationality*, in quanto contengono un'informazione non strettamente legata alla nazionalità del veicolo e per questo abbiamo deciso di correggerli tutti semplicemente in ASIAN.

Infine, per l'attributo *Make* abbiamo trovato il valore TOYOTA SCION che però non è un produttore esistente. Il valore corretto per il record in questione è semplicemente SCION, marchio di lusso della Toyota.

Le distribuzioni precise dei valori originali e di quelli derivati verranno presentate nella sezione successiva.

2.3 Accuratezza semantica

Con *"accuratezza semantica"* ci riferiamo alla verifica di tutti quei vincoli impliciti nella semantica dei vari attributi. Abbiamo verificato che:

- Nessun attributo numerico indicante un prezzo avesse valore negativo;
- Tutte le macchine di uno stesso produttore avessero la stessa nazionalità;
- Tutte le macchine di uno stesso modello fossero dello stesso produttore;

2.4 Missing Value

Il dataset Carvana presenta diversi missing values, per ogni attributo il numero di valori mancanti è mostrato nella tabella sottostante. I numeri tra parentesi indicano i cosiddetti *hidden missing values*, ovvero quei valori diversi da NULL o NaN ma che comunque non hanno un significato per l'attributo a cui si riferiscono o non rappresentano alcun tipo di informazione. Abbiamo considerato come mancanti i seguenti valori:

- 0 e 1 per gli *MMRs;*
- NOT AVAIL per *Color;*
- 0 per *WheelTypeID.*

Come si evince dalla tabella riportata sopra, gli attributi PRIMEUNIT e AUCGUART sono mancanti in oltre il 90% dei record e pertanto è stato deciso di eliminarli completamente.

La correzione di tutti i missing values verrà descritta nella sezione 3 di questo documento.

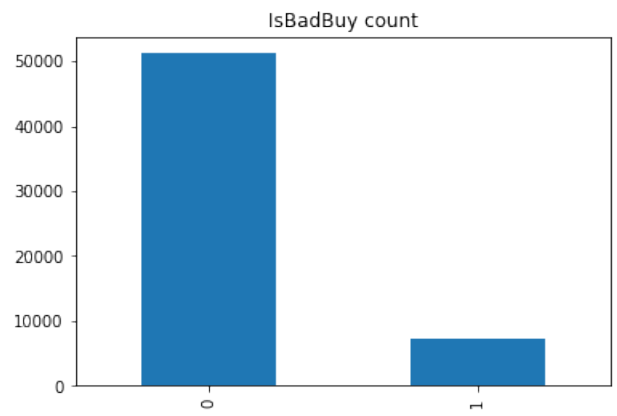
Trim	1911
SubModel	7
Color	7 (77)
Transmission	8
WheelTypeID	2573(4)
WheelType	2577
Nationality	4
Size	4
TopThreeAmericanName	4
MMRAcquisitionAuctionAveragePrice	13 (648)
MMRAcquisitionAuctionCleanPrice	13 (648)
MMRAcquisitionRetailAveragePrice	13 (648)
MMRAcquisitonRetailCleanPrice ‘	13 (648)
MMRCurrentAuctionAveragePrice	245 (393)
MMRCurrentAuctionCleanPrice	245 (393)
MMRCurrentRetailAveragePrice	245 (393)
MMRCurrentRetailCleanPrice	245(393)
PRIMEUNIT 55703	55703
AUCGUART 55703	55703

2.5 Distribuzione delle variabili

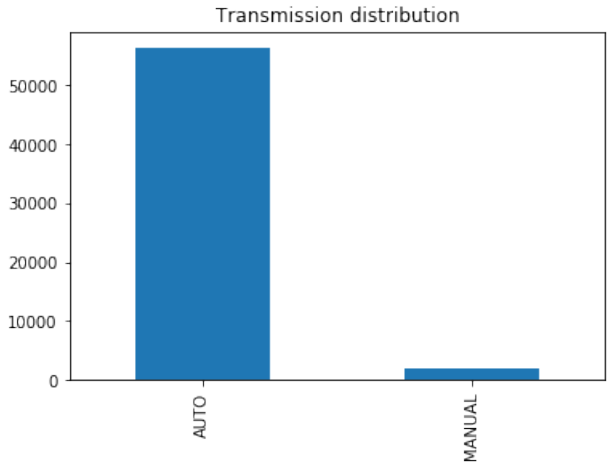
L' osservazione della distribuzione di ogni variabile è cruciale per giudicare l'importanza di ogni attributo e avere un'idea del loro impatto nella fase di analisi. Riportiamo di seguito i risultati che riteniamo più significativi.

IsBadBuy. Il dataset è fortemente sbilanciato: l'87,65% dei veicoli è stato classificato come *buon acquisto*. Questa caratteristica avrà un peso notevole soprattutto nella fase di classificazione, in cui si renderà necessario l'*undersampling* del dataset.

Transmission. Anche per questo attributo si nota un forte sbilanciamento verso i veicoli con il cambio automatico, che comprendono quasi il 97% del dataset.

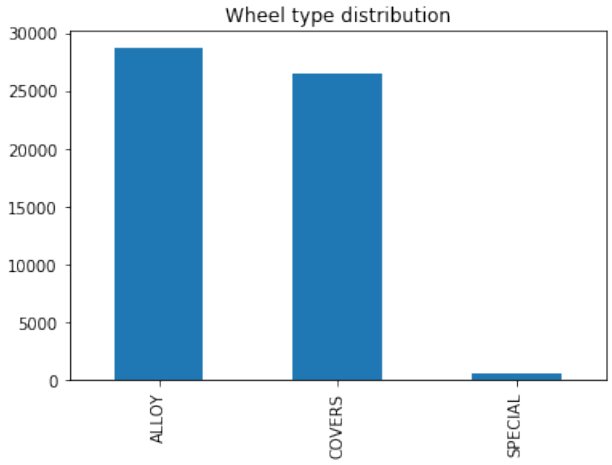


WheelType, WheelTypeID. É possibile individuare la seguente corrispondenza tra i valori di questi due attributi: *Alloy -> 1, Covers ->2, Special -> 3.* Come è possibile notare dal grafico in figura, *Alloy* (cerchi in lega) e *Covers* (copri-cerchi) sono equamente distribuiti tra i record del dataset.

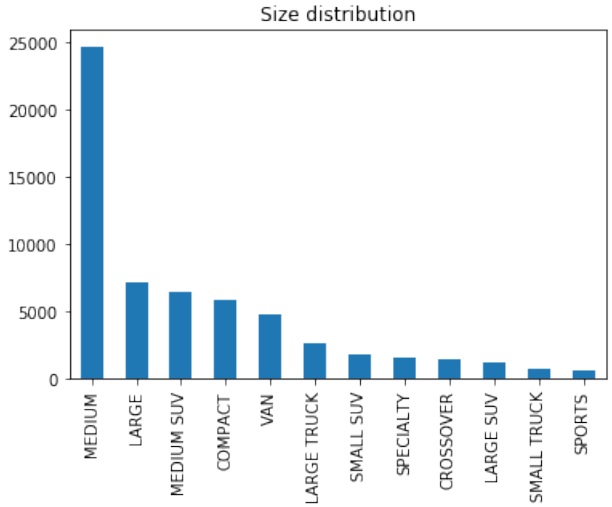


Size. È predominante il valore "MEDIUM" per il 42,21%, seguito da "LARGE" (12,19%), "MEDIUM SUV" (10,96%), COMPACT (9,87%) e VAN (8,01%).

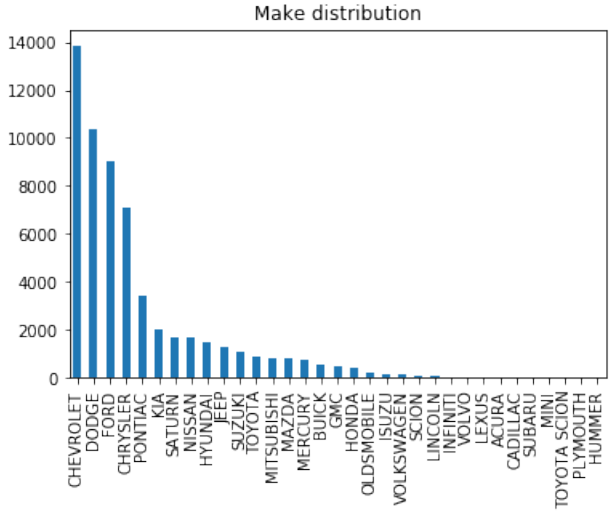
Make: Chevrolet è la marca più frequente.
Le auto sono per il 23,71% Chevrolet, seguito da Dodge (17,74%), Ford (15,41%), Chrysler (12,15%).



Model, Trim, Submodel. Questi tre attributi sono sufficienti ad identificare univocamente un veicolo. Come già anticipato, *Model* e *SubModel* contengono un certo grado di rumore, includendo informazioni non strettamente collegate al modello del veicolo. Queste informazioni, a volte incomplete, causano una duplicazione nei possibili valori di questi attributi. Grazie all'uso di espressioni regolari è stato possibile ripulire la colonna *Model* estraendo una nuova chiamata *ModelSimple* che riduce il range di possibili valori da 1006 a 266.



La correzione degli errori sintattici per l'attributo *Trim* ha ristretto i possibili valori distinti a 130. Tra questi *Bas* è il valore più frequente. Il trim è il livello di equipaggiamento di una macchina, dopo una serie di ricerche possiamo affermare che *Bas* è l'equipaggiamento Base. Per quanto riguarda tutti gli altri valori, è difficile trovare un significato preciso perchè ogni casa automobilistica adotta la sua nomenclatu-

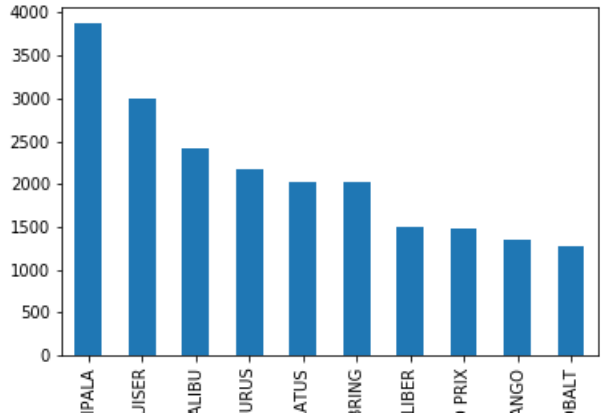


ra, che varia anche da un modello all'altro.

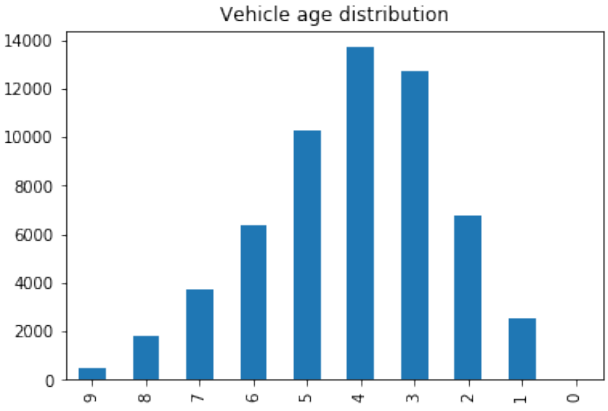
MODEL SIMPLE	
IMPALA	3879
PT CRUISER	3002
MALIBU	2407
TAURUS	2175
STRATUS	2027
SEBRING	2019
CALIBER	1504
GRAND PRIX	1478
DURAGO	1355
COBALT	1281

Così come l'attributo *Model*, anche *SubModel* conteneva informazioni non legate al modello della macchina e abbiamo provato ad estrarle ottenendo diverse colonne:

- Iniezione di carburante: può avere valori di tipo stringa come *SFI, SPI, EFI, MFI.* Solo per poco più di 7000 record è stato possibile estrarre questa informazione e di conseguenza non è stata utilizzata.
- Numero di porte: 2 o 4. È stato possibile estrarre questa informazione per il 90% dei record del dataset. La maggior parte dei veicoli ha 4 porte, come mostra il grafico in figura.
- Trazione: è stato possibile estrarre l'informazione sul numero di ruote motrici solo da poco più di 2000 records e pertanto questa informazione non è stata utilizzata.

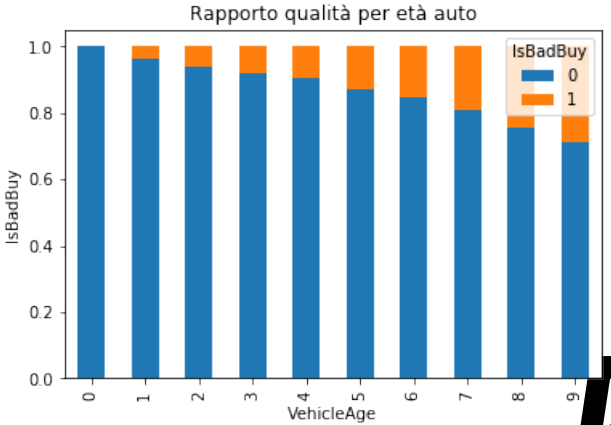


VehicleAge, VehYear. Anche questi due attributi sono ridondanti in quanto uno indica l'età della macchina e l'altro il suo anno di produzione. L'età più frequente è 4 anni e, come logicamente ci si aspetterebbe, più il veicolo è giovane e più e probabile che sia un *Good Buy.*



Nationality. Le auto sono per l'83,58% *AMERICAN*, seguite da *OTHER ASIAN* e *TOP LINE ASIAN*, che sono stati uniti in *ASIAN* . Una piccolissima minoranza di 152 record si riferisce ad auto europee (Volkswagen, Mini, Volvo) rispettivamente di nazionalità *TEDESCA, INGLESE* e *SVEDESE*, che abbiamo deciso di specificare.

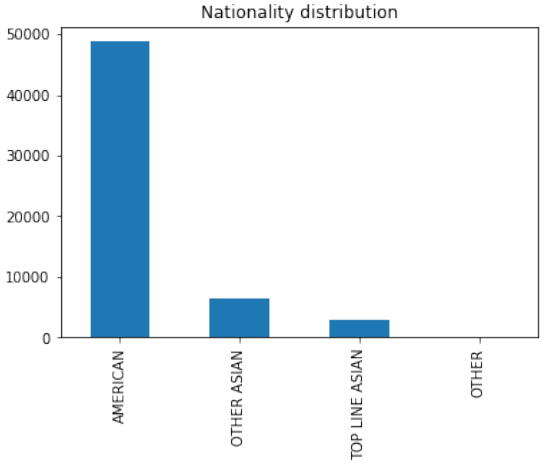
Color: i colori più frequenti sono argento, bianco e blu
Il valore più frequente è "silver" (20,23%), seguito da "white" (16,73%), "blue" (14,17%), "grey" (10,76%) e "black" (10,39%)

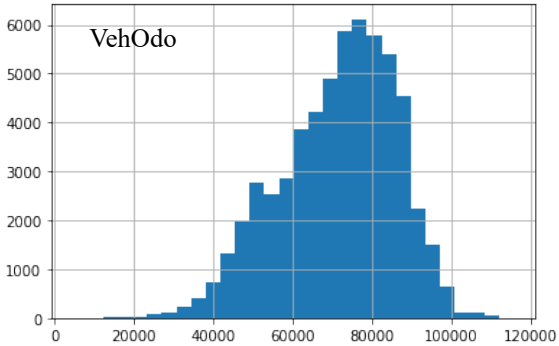


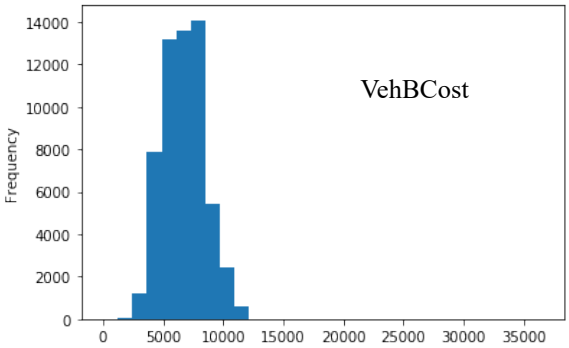
VehOdo. la media del chilometraggio è di 71478 km e la maggior parte dei valori si concentra tra i 60.000 e gli 80.000 km. Dal grafico in figura possiamo notare una distribuzione poissoniana asimmetrica negativa a sinistra.

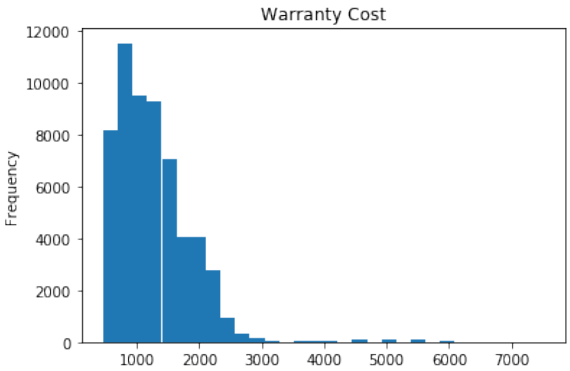
WarrantyCost. Il costo di garanzia medio è 1276 dollari e si concentra tra 462 $ e 3000 $, con una distribuzione asimmetrica a destra.

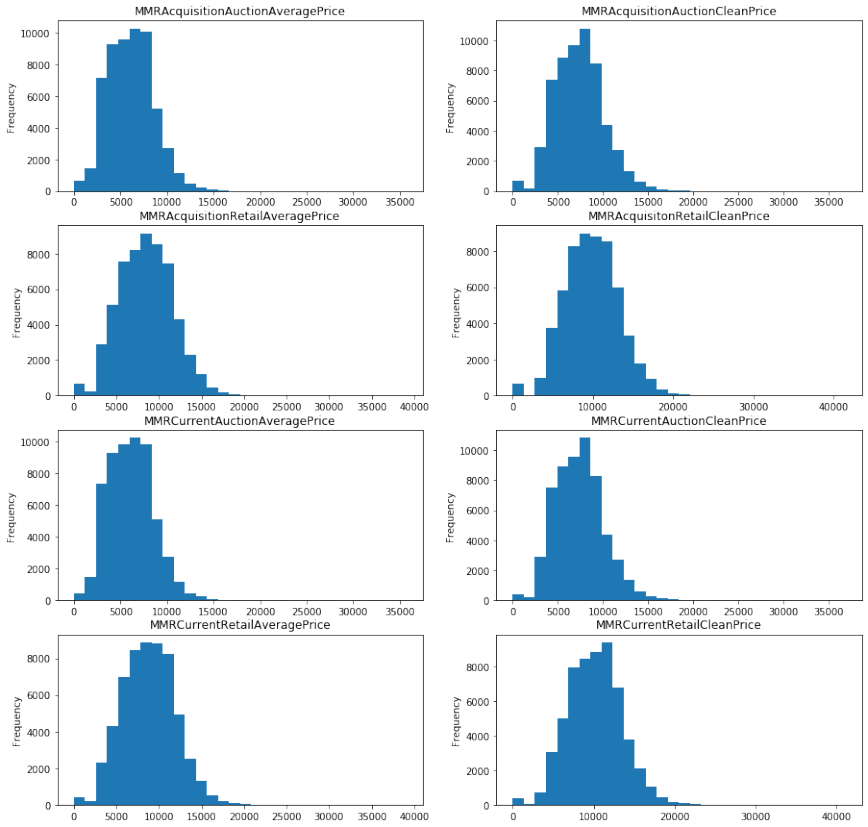
MMRs, VehBCost. Tutte le distribuzioni dei prezzi hanno una distribuzione log-normal, come è possibile evincere dai grafici in figura.











3 DATA PREPARATION

3.1 Eliminazione degli attributi non significativi o ridondanti.

La comprensione della semantica degli attributi, insieme con il calcolo della *Pearson's Correlation* per gli attributi numerici e della *Chi-Squared Correlation* per quelli categorici ci ha portato a formulare le seguenti considerazioni e a cancellare determinate colonne.

	VehOdo	MMRA-AAP	MMRA-ACP	MMRA-RAP	MMRAR-CP	MMR-CAAP	MMR-CACP	MMR-CRAP	MMR-CRCP	VehB-Cost	Warran-tyCost
VehOdo	1	-0.03	0.01	0.03	0.6	-0.04	-0.01	0.01	0.04	-0.06	0.45
MMRAAAP	-0.03	1	0.99	0.90	0.89	0.95	0.94	0.87	0.87	0.82	0.004
MMRAACP	0.01	0.99	1	0.89	0.90	0.93	0.94	0.86	0.87	0.83	0.04
MMRARAP	0.03	0.90	0.89	1	0.99	0.85	0.85	0.92	0.91	0.77	-0.01
MMRARCP	0.06	0.89	0.90	0.99	1	0.85	0.85	0.91	0.92	0.78	0.02
MMRCAAP	-0.04	0.95	0.93	0.855	0.85	1	0.99	0.91	0.90	0.80	-0.001
MMRCACP	-0.01	0.94	0.94	0.85	0.85	0.99	1	0.90	0.91	0.81	0.03
MMRCRAP	0.01	0.87	0.86	0.92	0.91	0.91	0.90	1	0.99	0.78	-0.01
MMRCRCP	0.04	0.87	0.87	0.91	0.92	0.90	0.91	0.99	1	0.78	0.02
VehBCost	-0.06	0.82	0.83	0.77	0.78	0.80	0.81	0.78	0.78	1	0.03
WarrantyCost	0.45	0.004	0.04	-0.01	0.02	-0.001	0.03	-0.01	0.02	0.03	1

- VehYear e VehicleAge sono ridondanti e hanno la stessa semantica e distribuzione, per questo abbiamo deciso di mantenere solo VehicleAge come attributo categorico.
- WheelType e WheelTypeID sono ridondanti e per questo è stato deciso di mantenere solo WheelType.
- Gli attributi VNZIP1 e VNST sono ridondanti e inoltre VNZIP1 ha troppi valori distinti. Abbiamo raggruppato i valori di VNST per area geografica derivando una nuova colonna chiamata USArea con 4 possibili valori: Sud, NordEst, Ovest e Centro-Ovest.
- *BYRNO, RefId* e *PurchDate* sono stati eliminati perché giudicati non interessanti ai fini dell'analisi.
- MMRs e *VehBCost* mostrano un'elevatissima correlazione e a causa di questo abbiamo deciso di mantenere solamente l'attributo *VehBCost* che ricordiamo indica il prezzo pagato per l'acquisto del veicolo all'asta. Crediamo inoltre che, basandoci sulla semantica degli 8 MMRs, solamente MMRAcquisitionAuctionAveragePrice possa essere veramente significativo per il dataset.

3.2 Gestione di missing values e outliers.

Missing values.

Come già anticipato, il dataset Carvana presenta un consistente numero di missing values che sono stati trattati come descritto di seguito.

Dato il ridotto numero di valori mancanti per TopThreeAmericanName, Nationality e Size è stato possibile correggerli in maniera esatta controllando l'attributo Make, grazie alla nostra conoscenza del dominio e brevi ricerche su Google.

L'attributo Trim presentava 1911 valori mancanti che abbiamo portato a 525 raggruppando i records per ModelSimple e SubModel. Abbiamo fatto dei controlli a campione per verificare la correttezza dei dati, con l'ausilio del sito cars.com. I records rimasti sono stati eliminati dal dataset.

Il sito cars.com non ha aiutato per correggere i missings di Transmission in quanto tutti gli 8 veicoli sono presenti sul mercato sia con il cambio manuale che con quello automatico.

Basandoci sulla distribuzione dei records raggruppati per ModelSimple, abbiamo corretto assegnando il valore AUTO che di certo non altera la distribuzione originale.

In base alla nostra conoscenza del dominio, abbiamo assunto che il tipo di cerchioni di un veicolo faccia parte del suo livello di equipaggiamento. Quindi per la correzione dei missing

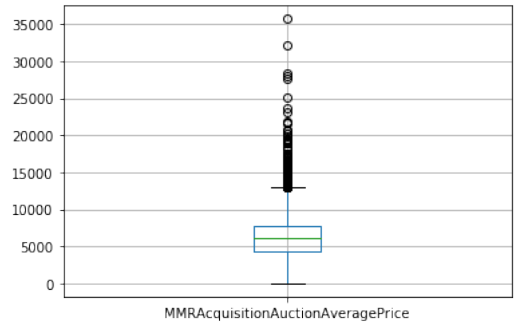
values in WheelType abbiamo raggruppato i records in base all'attributo Trim prima di effettuare la sostituzione. Così facendo abbiamo rispettato la distribuzione dei valori iniziali e corretto tutti i missing values tranne 1. Il sito cars.com ci ha aiutato a correggere quest'ultimo missing value, relativo ad una Suzuki Grand Vitara1.
Per la sostituzione dei missing values (hidden e non) negli attributi di prezzo (MMRs e VehBCost) abbiamo tentato due strade: raggruppamento per ModelSimple con riempimento usando la media e regressione lineare multivariata. Abbiamo scelto di utilizzare i risultati ottenuti con la prima soluzione perchè hanno preservato meglio la distribuzione iniziale, inoltre non siamo riusciti ad ottenere un accuracy sufficientemente alta per la regressione. Di seguito riportiamo le differenze nella distribuzione dei valori prima e dopo la correzione dei missing values.

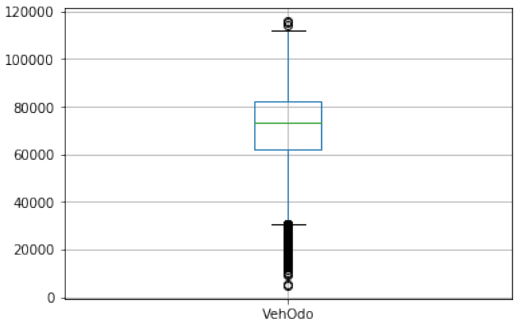
[PRIMA]	MMRAAAP	MMRAACP	MMRARAP	MMRARCP	MMRCAAP	MMRCACP	MMRCRAP	MMRCRCP
count	57208.00	57208.00	57208.00	57208.00	57230.00	57230.00	57230.00	57230.00
mean	6189.40	7446.95	8585.07	9953.50	6165.92	7431.54	8828.42	10205.64
std	2373.57	2606.66	3029.53	3223.84	2379.11	2612.98	3004.09	3200.72
min	884.00	1076.00	1455.00	1662.00	369.00	494.00	899.00	1034.00
25%	4327.00	5467.00	6358.00	7550.00	4303.00	5447.00	6576.00	7829.25
50%	6124.00	7340.00	8481.00	9837.00	6074.00	7324.00	8756.00	10118.00
75%	7786.00	9042.00	10677.00	12111.00	7749.00	9025.00	10922.00	12321.00
max	35722.00	36859.00	39080.00	40308.00	35722.00	36859.00	39080.00	40308.00

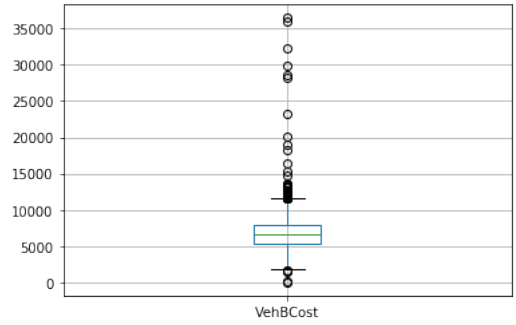
[DOPO]	MMRAAAP	MMRAACP	MMRARAP	MMRARCP	MMRCAAP	MMRCACP	MMRCRAP	MMRCRCP
count	57863.00	57863.00	57863.00	57863.00	57864.00	57864.00	57864.00	57864.00
mean	6192.27	7450.88	8589.02	9958.49	6165.93	7432.46	8829.12	10207.20
std	2366.39	2599.27	3018.80	3212.97	2372.17	2605.71	2994.52	3190.95
min	884.00	1076.00	1455.00	1662.00	369.00	494.00	899.00	1034.00
25%	4342.00	5485.00	6368.50	7571.00	4311.00	5462.00	6587.00	7840.75
50%	6129.00	7353.00	8488.00	9849.00	6078.00	7330.44	8756.00	10127.00
75%	7782.00	9044.00	10665.50	12106.50	7746.00	9025.00	10918.00	12316.00
max	35722.00	36859.00	39080.00	40308.00	35722.00	36859.00	39080.00	40308.00

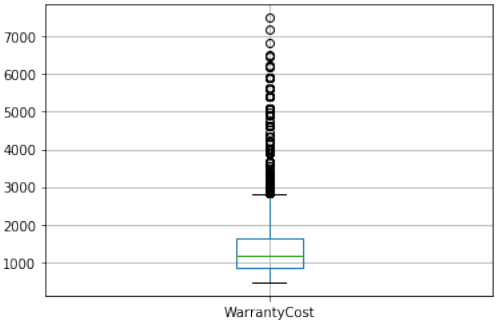
Outliers.

Oltre agli outliers categorici, ovvero i record con caratteristiche che hanno una frequenza estremamente più bassa rispetto al resto dei records, il dataset Carvana presenta anche una modesta quantità di outliers per gli attributi numerici.
Dai boxplots in figura è possibile notare come la maggior parte degli outliers siano sopra il 3° quartile
È stato deciso di non eliminare gli outliers a priori. Nelle sezioni riguardanti l'analisi verranno proposte alcune soluzioni tramite il metodo IQR.







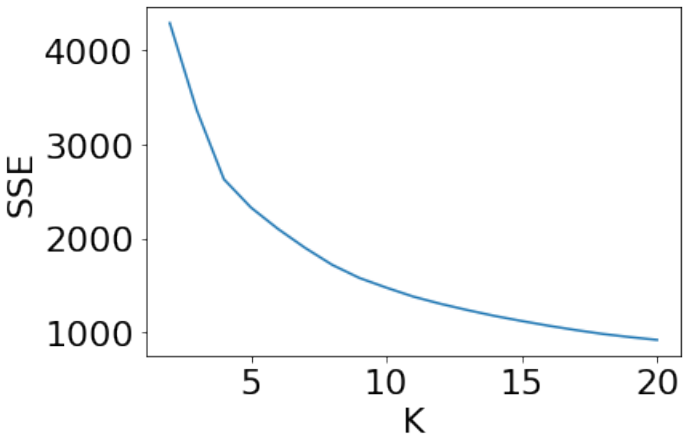


4 CLUSTERING

Data la complessità del dataset, causata principalmente dal consistente numero di attributi presenti, l'operazione di clustering è stata importante principalmente per la comprensione dei dati e per la valorizzazione della distribuzione dei record.
Al fine di applicare i diversi algoritmi per il clustering sono state effettuate sul dataset alcune modifiche preliminari: sono stati rimossi tutti gli attributi categorici; per tutti gli attributi ridondanti o estremamente correlati si è tenuto esclusivamente quello più significativo; si sono normalizzati gli attributi rimanenti con il metodo Min-Max.
In particolare per gli 8 attributi di prezzo "MMRs" e *VehBCost*, la cui correlazione era prossima ad 1, si è deciso di tenere esclusivamente quest'ultimo che indica il prezzo pagato per l'acquisto del veicolo.

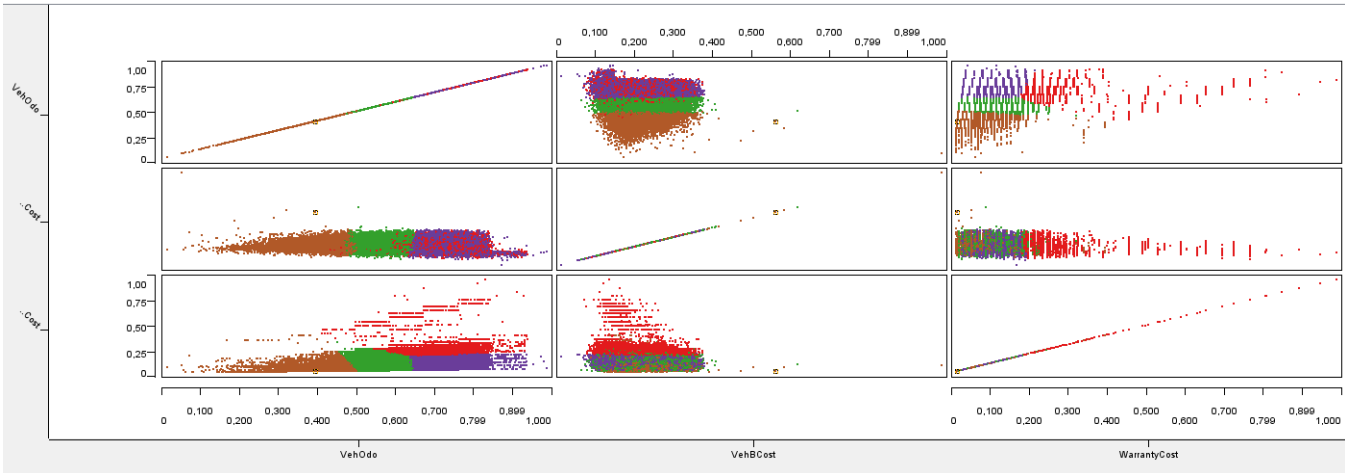
4.1 K-means

Essendo l'algoritmo K-means suscettibile alla presenza di outliers, abbiamo inizialmente provato ad eseguire l'algoritmo in assenza di questi. La rimozione degli outliers è stato effettuata con il metodo IQR (interquartile range) e diversi valori di cut-off K. Contrariamente alle nostre aspettative, i risultati così ottenuti non sono stati ottimali: i valori di SSE troppo alti e l'assenza di clustering visibilmente significativi ci hanno spinto a preferire i risultati ottenuti in presenza degli outliers.



Abbiamo valutato come ottimale la configurazione dell'algoritmo con distanza euclidea come funzione di distanza e con obiettivo l'individuazione di soltanto 4 cluster. Queste impostazioni, scelte sia tramita l'analisi del decrescere del SSE all'aumentare del numero di clustering, sia tramite diversi tentativi, ci ha permesso di ottenere una rappresentazione grafica dei cluster più comprensibile. Oltre a questo sono stati testati i valori di K: 5, 6, 7 e 10.
Avendo un numero molto elevate di record, la suddivisione in quattro cluster risulta molto generica: i risultati ottenuti sono variegati e poco caratterizzanti. Possiamo, tuttavia, tracciare delle leggere linee di demarcazione:

- Cluster 0: Contiene principalmente record che ben rappresentano la media del dataset. È inoltre il cluster più grande, con quasi 20.000 record.



- Cluster 1: In media, le auto al suo interno hanno un costo di garanzia molto più alto rispetto alla media complessiva. Hanno anche un chilometraggio leggermente più alto rispetto alla media, simile al chilometraggio medio del cluster 3.

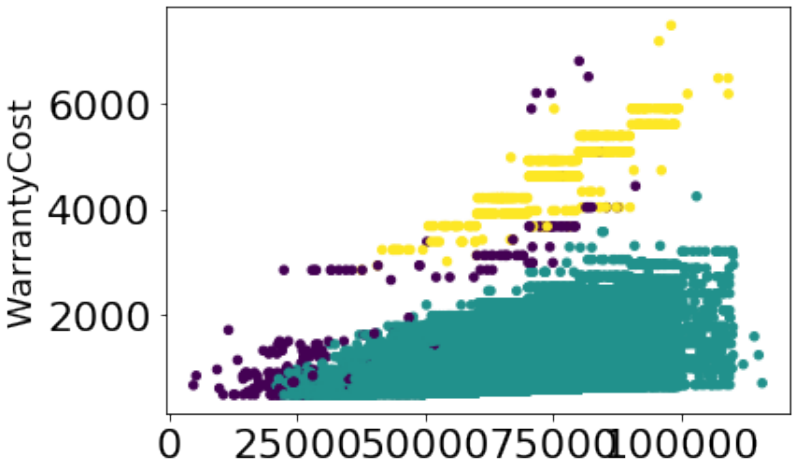
- Cluster 2: il chilometraggio di queste auto si discosta in negativo di circa 5000 dalla media del dataset, e, poco meno, dalla media degli altri cluster. Probabilmente anche per questa ragione registrano un costo di garanzia leggermente inferiore a quello della media generale, sebbene simile a quello dei cluster 0 e 3.
  Rispetto alla distribuzione generale, questo cluster contiene una maggioranza di veicoli Chrysler piuttosto che Grand Motor.

- Cluster 3: il terzo cluster presenta una media di prezzo d'acquisto inferiore a quella generale e un chilometraggio medio leggermente superiore alla media generale. Possiamo supporre che appartengano a questo gruppo principalmente auto in condizioni d'acquisto mediocri, molto utilizzate.

4.2 DBSCAN

Per l'algoritmo DBSCAN non sono state effettuate operazioni preliminari aggiuntive rispetto a quelle elencate nell'introduzione. A causa della complessità del calcolo, la computazione delle distanze fra tutti i punti del dataset ai fini della selezioni del miglior valore di "eps" è stata effettuata su un sottogruppo del dataset generato casualmente e contenente il 65% dei valori. L'algoritmo in sé è stato invece eseguito su tutto il dataset.
La funzione di distanza che ha dato risultati migliori sul dataset è stata la *cosine distance*. Mentre la euclidea riusciva ad individuare soltanto un cluster all'interno del dataset - a causa dell'aspetto "continuo" e regolare di questo - con l'individuazione di più o meno outliers al variare delle impostazioni, la cosine invece, al costo di qualche punto nel valore di silhouette, è riuscita a estrapolare informazioni più interessanti.

Come parametri si è selezionato 0.15 come "eps", ovvero come distanza entro il cui studiare la densità di item intorno ad un punto, e 30 come limite minimo di item per identificare una zona ad alta densità (ovvero "min_sup"). Sono state testate diverse combinazioni, come: eps 0.0003, k=5; eps 0.0002, k= 15; eps 0.2 k= 30; eps 0.05 k= 30.
La combinazione scelta è quella che riportava il più alto valore di validazione.
Con questi parametri siamo riusciti ad identificare 2 cluster e 207 outliers o noise points. Tuttavia soltanto il cluster 1 è interessante ai fini di una analisi, in quanto il cluster 0 contiene al suo interno il 99% dei record del dataset e quindi non contiene alcuna informazione aggiuntiva.
Il cluster 1 conta soltanto 448 records, caratterizzati principalmente da un elevatissimo costo di garanzia, circa 3.000$ dollari in più rispetto alla media generale, e da un costo d'acquisto leggermente inferiore a quello medio, più o meno intorno ai 1.000$.
Si può osservare come una prima metà di queste auto siano prodotte dalla *Chevrolet*, in linea quindi con la distribuzione generale del dataset, e come invece l'altra metà siano invece prodotte dalla *Pontiac* e dalla *Buick*, case di produzione fortemente minoritarie nel dataset completo. Fanno parte di questo cluster tutte le auto "*Venture FWD V6*" e "V*enture FWD V6 3.4L*" presenti nel dataset e, più in generale, ne fanno parte quasi esclusivamente van, furgoncini, o comunque, auto medio-grandi.

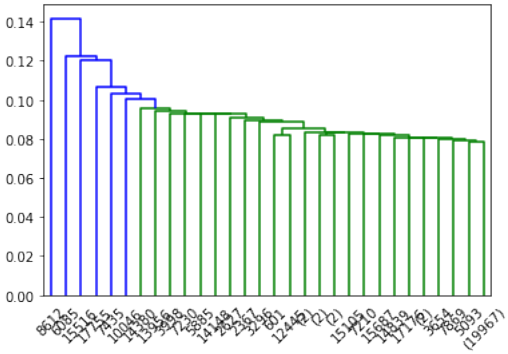


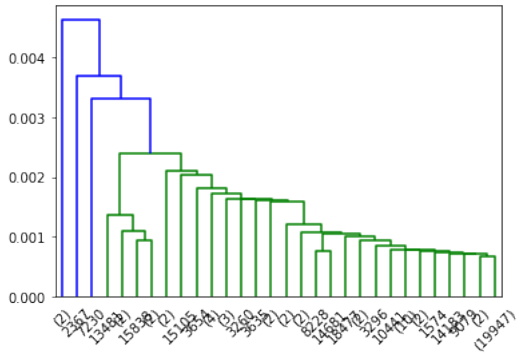
4.3 Gerarchico

Essendo anche questo algoritmo sensibile alla presenza di outliers, in fase preliminare si è deciso di rimuovere preventivamente gli outliers sempre attraverso il metodo IQR, oltre ad eliminare attributi categorici e ridondanti.
Per l'esecuzione dell'algoritmo abbiamo scelto di utilizzare sia la funziona di distanza "cosine" che "euclidean", ciascuna con le metriche "single link", "complete link" ed "average". In aggiunta per l'euclidean si è utilizzata anche la "ward", impossibile da applicare sulla cosine.

In generale osserviamo che con il single link si sono ottenuti risultati poco interessanti: notiamo che i record tendono velocemente ad agglomerarsi in un unico cluster che, al crescere della distanza, ingloba lentamente tutti gli altri item del dataset. Il che è spiegabile con la sostanziale continuità dei record, che sono quindi molto vicini tra di loro.

Single euclidean

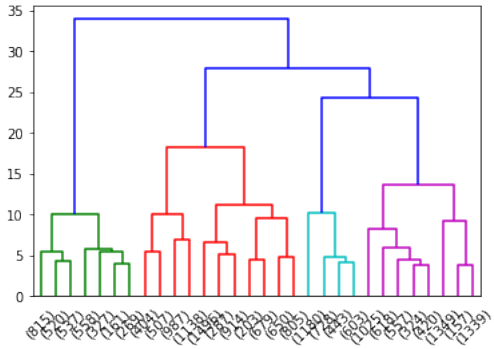




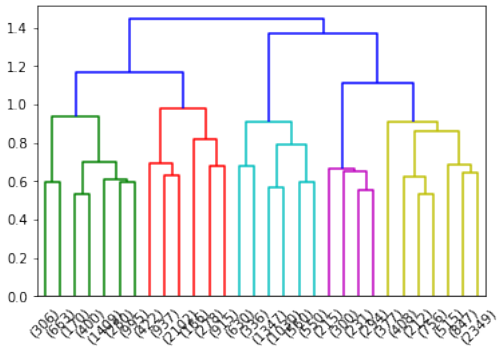
Single cosine

Il metodo ward, che tende a minimizzare la varianza all'interno dei cluster, sembra ricalcare in parte il risultato ottenuto tramite K-means, con la formazione di 4 cluster di cui uno leggermente più grande degli altri.

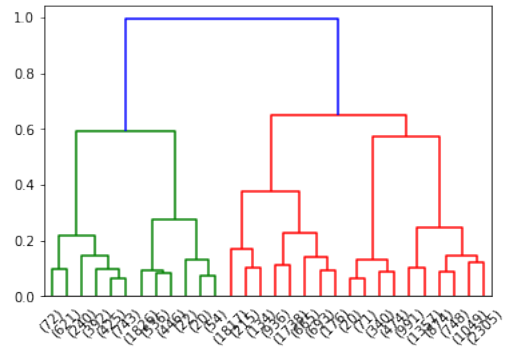
La complete link ha portato due risultati diversi per l'euclidean e la cosine. Nella prima a distanza 1 notiamo la formazione di 5 cluster, di simili dimensioni. Nella seconda invece il dataset si divide in due metà già a 0.7.



Ward



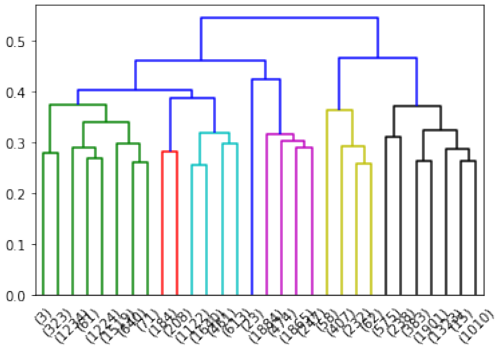
Complete euclidean

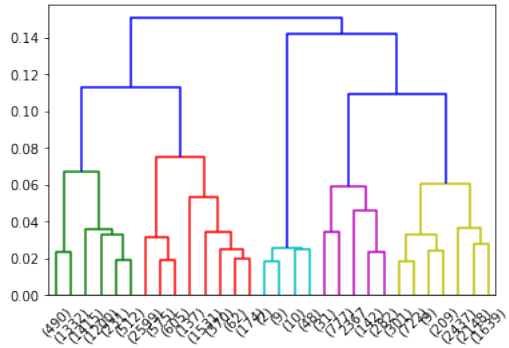


Complete cosine

L'*average,* infine, ha prodotto un risultato interessante con la *cosine*, con la formazione di cinque cluster ben definiti. Mentre con *l'euclidean* i risultati sono riconducibili a quelli ottenuti con la complete link, con la formazione di diversi cluster già a brevi distanze.

Average euclidean





Average cosine

4.4 Considerazioni finali

Le operazioni di clustering effettuate non hanno, in generale, soddisfatto le nostre aspettative iniziali. Sebbene in presenza di valori di validazione piuttosto alti, il dataset risulta troppo uniforme per offrire una suddivisione in cluster soddisfacente: tutte le suddivisioni analizzate risultano o troppo labili e poco definite, come nel caso del K-means, o troppo ristrette e poco interessanti, come nel caso del DBSCAN.

Come anticipato, la divisione in 4 cluster individuata attraverso l'algoritmo K-means risulta la più calzante per una rappresentazione generale della composizione del dataset, incapace tuttavia di aggiungere informazioni aggiuntive rispetto a quanto analizzato nella fase di data understanding.
I cluster individuati così risultano inoltre labili, poco definiti, con valori di silhouette piuttosto bassi: 0.27 per il cluster 0; 0.26 per il cluster 1; 0.43 per il cluster 2; 0.21 per il cluster 3; con una media generale di 0.29.

Numericamente parlando, il risultato ottenuto tramite DBSCAN ha un valore generale di silhouette nettamente migliore, pari a 0.48. Tuttavia i due cluster ottenuti sono troppo sproporzionati: il secondo contiene solo 207 elementi, pari allo 0.3% del dataset originale, mentre il primo ne contiene il 99% (il restante 0.7% è stato definite noise dall'algoritmo) .

Simili i risultati ottenuti con hierarchical.
Il miglior valore di silhouette è stato ottenuto con il metodo ward, ovvero 0.28. Anche in questo la divisione ottenuta sembra ricalcare quella del K-means. Più bassi i valori ottenuti con la complete link, intorno allo 0.17. I risultati ottenuti con la single link sono invece da scartare in quanto, come detto precedentemente, evidenziano la formazione di un unico cluster.

5. ASSOCIATION RULES

Nella formulazione delle regole associative, l'elevato numero di attributi altamente correlati, se non persino ridondanti, avrebbe comportato l'individuazione di un elevato numero di regole già assodate, e dunque di poco valore per la nostra analisi.
Per migliorare l'analisi, abbiamo dunque deciso di rimuovere i seguenti attributi:

- "TopThreeAmericanName" e "VNST" in quanto contenenti informazioni geografiche già presenti in "USregion".
- "Model" e "Submodel" in quanto ridondanti con "ModelSimple".
- Gli 8 valori "MMR", rappresentanti stime di prezzo fortemente correlate al prezzo d'acquisto dell'auto, ovvero "VehBCost".

Inutili al fine dell'algoritmo anche "RefId", in quanto identificativo, e "PurchDate", che invece registra la data d'acquisto.
I restanti attributi sono stati puliti dagli outliers attraverso il calcolo dei quartili e, per quanto riguarda quelli continui, discretizzati in modo da ottenere dei range di valori numerici su cui applicare l'algoritmo.
Infine il dataset è stato suddiviso in un training e in un test.

Essendo "IsBadBuy", ovvero il target dell'analisi, molto sbilanciato, abbiamo notato che "cattivo acquisto", il valore meno ricorrente, non compariva mai nelle regole individuate. Abbiamo dunque deciso di proporre l'analisi sia con la distribuzione originale dei valori sia con una versione bilanciata al 50 e 50 del dataset.

5.1 Dataset Originale

Pattern

Sono stati presi in considerazione tre diverse tipologie di pattern: "frequent", ovvero tutti i pattern che appaiono all'interno del dataset un numero di volte pari o superiore ad un limite stabilito, cioè superiori al "min_sup"; "closed", cioè quei frequent pattern con una frequenza

superiore a qualsiasi altro "superset" che li contenga; "maximal", cioè quei frequent pattern che non hanno alcun "superset" che li contenga.
Per ognuna di queste tipologie abbiamo eseguito l'algoritmo con 5 diverse soglie di frequenza, cioè con tre diversi "min_sup": 1%, 2%, 5%, 10% e 20%.

With types a we found (values are % of support)
{1: 41054, 2: 13953, 5: 3027, 10: 853, 20: 208}

With types c we found (values are % of support)
{1: 36954, 2: 13446, 5: 3018, 10: 853, 20: 208}

With types m we found (values are % of support)
{1: 4779, 2: 1865, 5: 475, 10: 158, 20: 45}

Come è logico supporre, i "frequent pattern" sono di numero maggiore rispetto ai "closed frequent itemset", che a loro volta sono maggiori rispetto ai "max frequent itemset". In particolare possiamo osservare come i pattern ottenuti con i "closed frequent itemset" siano solo di poco inferiori ai "frequent": la differenza è intorno al 7% con i valori di supporto più bassi, mentre è praticamente nulla già al 5% di supporto. I "max frequent itemset" invece sono molto inferiori rispetto ai "closed": all'incirca del 80%.
Da queste informazioni abbiamo conferma di quanto il dataset sia variegato. I pattern individuati sono generalmente piccoli e pochissimi "superset" sono più o ugualmente frequenti rispetto ai rispettivi sottogruppi.

Anche all'aumentare del valore di "min_sup" ovviamente il numero dei pattern trovati diminuisce, con cambiamenti vicini al 70% per ogni gradino.

La maggior parte dei pattern individuati contiene valori che sono molto frequenti nel dataset, come ad esempio:

- 4 Porte, cambio automatico;
- Buon acquisto, cambio automatico;
- Macchina americana, cambio automatico;
- 4 Porte, buon acquisto;
- Ecc.

Altre evidenziano patten ovvi, già individuati in fase di data understanding:

- Macchina americana, Chevrolet;
- Un modello di macchina, la rispettiva casa di produzione;
- Ecc.

Soltanto a frequenza più bassa troviamo informazioni interessanti, come:

- Chilometraggio tra 40.000 e 50.000, costo garanzia tra 500 e 1100.
  Cioè le auto con un chilometraggio inferiore rispetto alla media hanno un costo di garanzia più basso.
- Van, costo d'acquisto tra 3.000 e i 6.000. Cioè poco inferiore alla media.
- Compact car, costo d'acquisto tra 3.000 e i 6.000.
- Compact car, costo di garanzia tra i 500 e 1100.

- MEDIUM SUV, costo d'acquisto TRA I 6.500 e 9.500, costo di garanzia tra 1.100 e 1.900
- Ecc.

Regole

Per la formulazione delle regole abbiamo scelto di tenere gli item con una frequenza superiore al 20% del dataset, in quanto minori di numero e più significativi. Abbiamo eseguito l'algoritmo tenendo in considerazione diversi livelli minimi di confidenza, ovvero la probabilità di trovare un item B sapendo che l'item A è presente. Abbiamo impostato come valori di confidence: 1, 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 e 100. All'aumentare del valore di "min_conf" riscontriamo un calo del numero delle regole individuate, con particolare focus tra il 20 e il 30% di confidence, dove abbiamo una diminuzione del 45% e tra il 90 e 100% di confidenze, dove abbiamo un picco del 98%: sulle 610 con confidenza superiore al 90%, sono 8 hanno confidenza pari al 100%.

With confidence 1, support 20, we found 51230 rules
With confidence 2, support 20, we found 38197 rules
With confidence 5, support 20, we found 23839 rules
With confidence 10, support 20, we found 17022 rules
With confidence 20, support 20, we found 9010 rules
With confidence 30, support 20, we found 4838 rules
With confidence 40, support 20, we found 3730 rules
With confidence 50, support 20, we found 2731 rules
With confidence 60, support 20, we found 1657 rules
With confidence 70, support 20, we found 1131 rules
With confidence 80, support 20, we found 1021 rules
With confidence 90, support 20, we found 610 rules
With confidence 100, support 20, we found 8 rules

Ritenendo le regole ottenute con confidenza 90% poco informative, osserviamo le regole più interessanti ottenute con confidence 80%:
(Le regole hanno il seguente formato (lift)(items antecedenti)(conseguenza))

- (1.0457939570042045)('OTHER_Auction', 'Good Buy', 'AUTO_Transmission') ---> 4D
- (1.0178138990874082)('CHEVROLET_Make', 'Good Buy') ---> AUTO_Transmission
- (1.0153902125615344)('ALLOY_WheelType', 'South', '4D') ---> AUTO_Transmission
- (1.0465983436000288)('3_Age',) ---> Good Buy
- (1.031258600656461)('CHEVROLET_Make', 'AUTO_Transmission') ---> Good Buy
- (1.0292970383525062)('COVERS_WheelType', 'MANHEIM_Auction') --->Good Buy

Abbiamo utilizzato le prime tre per il riempimento di valori mancanti, mentre le ultime due per predirre il target nel nostro test set. Otteniamo rispettivamente: 96% di accuracy sulla prima, 98% sulla seconda e sulla terza, 92% sulla quarta, 91% sulla quinta e 90% sull'ultima.

5.2 Dataset Bilanciato

Pattern

Per il dataset bilanciato sono state prese in considerazione le medesime tipologie viste per il dataset sbilanciato. Osserviamo che i rapporti tra queste tre tipologie sono simili a quelli visti in precedenza: la differenza tra il numero di pattern ottenuti con "frequent" e con "closed" è

minima, ancora minore a quella ottenuta nel dataset sbilanciato; molto grande invece quella con "maximal".

With types a we found (values are % of support)
{1: 49770, 2: 14971, 5: 2867, 10: 719, 20: 156}

With types c we found (values are % of support)
{1: 40631, 2: 13971, 5: 2847, 10: 719, 20: 156}

With types m we found (values are % of support)
{1: 7724, 2: 2862, 5: 657, 10: 184, 20: 46}

Avendo deciso di eseguire l'algoritmo su un dataset bilanciato per individuare dei pattern contenenti il valore "Bad Buy", vediamone qualcuno:

- 5 anni, cattivo acquisto;
- chilometraggio tra gli 80.000 e i 90.000, cattivo acquisto;
- costo garanzia tra 11.000 e 18.000, cattivo acquisto
- ecc.

Gli altri pattern restano simili a quelli individuati precedentemente:

- 3 anni, buon acquisto;
- Americana, Chevrolet;
- ecc.

Regole
Anche in questo caso per la formulazione delle regole abbiamo scelto di tenere gli item con supporto maggiore del 20%. Abbiamo poi eseguito l'algoritmo con i valori di confidenza 1, 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100.
Osserviamo le regole più interessanti ottenute con confidenza 80%.

- (1.0814969930788634)('[462.0, 680.9)_Warranty',) ---> 4D
- (4.364421812148303)('LS_Trim',) ---> CHEVROLET_Make
- (1.053859954079775)('GREY',) ---> 4D

Utilizzandole per correggere i valori mancanti otteniamo un'accuratezza del 94% con la prima, del 93% con la seconda e del 96% con la terza. Non abbiamo ottenuto regole con la classe "isBadBuy" come target.

6. CLASSIFICATION

6.1 Preparazione dataset

Dalla matrice di correlazione e dall'esperienza fatta sul dataset nei punti precedenti, abbiamo notato che VehBCost e tutti gli attributi di prezzo MMR sono molto correlati tra di loro, quindi tra questi abbiamo deciso di tenere solo VehBCost. Inoltre l'attributo Model è stato scartato in quanto ModelSimple è la sua versione semplificata e corretta e al posto di VNST usiamo USRegion, che ci dice in quale parte di America risiede quello stato, riducendo di molto il numero di valori univoci. RefID, PurchDate e IsOnlineSale sono irrilevanti o difficili da gestire (discretizzazione di

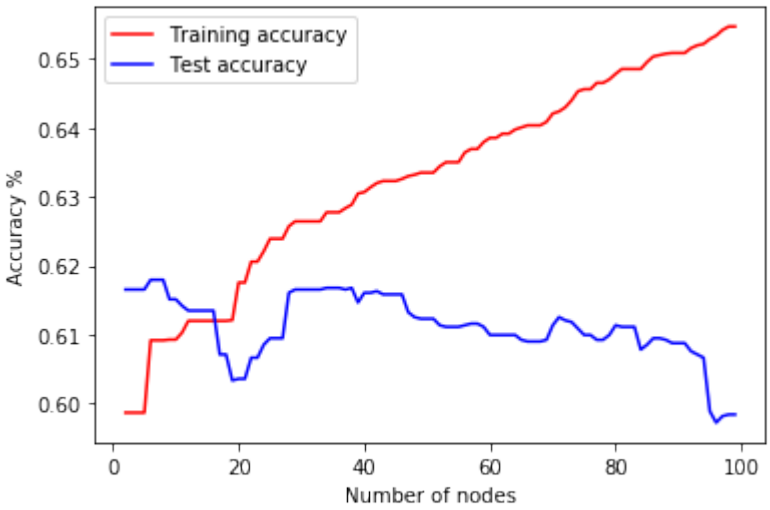
PurchDate). Infine abbiamo scartato anche SubModel, in quanto l'elevato numero di valori unici rallentava l'esecuzione degli algoritmi e il dataset è troppo piccolo rispetto a questi valori affinchè siano rilevanti.
Dato che su python i DecisionTree non supportano direttamente gli attributi categorici, abbiamo inizialmente usato un LabelEncoder, il quale tenta di dare un valore numerico ad ogni valore unico, ma dato che funziona bene solo sugli attributi ordinali e i nostri sono semplicemente categorici, abbiamo optato successivamente per l'OneHotEncoding, ossia abbiamo aggiunto al dataset un numero di colonne pari al numero di valori unici e ogni attributo aggiunto è binario, quindi passiamo da 17 colonne a 427.
Infine abbiamo creato 4 diversi dataset: uno sbilanciato (originale), uno bilanciato tramite undersampling sulla classe maggioritaria (7080 GoodBuy, 7080 BadBuy), uno solo numerico normalizzato sbilanciato e uno solo numerico normalizzato bilanciato (per testare le SVM, che faticano con un numero elevato di dimensioni e quindi non possiamo fare OneHotEncoding). Abbiamo inoltre applicato l'holdout sul caravana_training e le dimensioni dei training set e dei test set è rispettivamente (0.7, 0.3)

6.2 Validazione di vari alberi

Per trovare i migliori parametri per gli alberi, siamo partiti guardando le curve di training e test error rispetto al numero di nodi, per avere una stima del valore ottimale e cercare di ridurre il numero di parametri per la GridSearch, dalla curva ottenuta abbiamo stimato che il valore ottimale dovrebbe essere circa intorno a 20 nodi.

Partendo da questa informazione, abbiamo optato per una RandomizedSearchCV, la quale esplora lo stesso spazio della GridSearch, ma in modo computazionalmente più veloce e inoltre applicando CrossValidation per ottenere un migliore ValidationScore. I parametri testati sono stati:

- Profondità dell'albero [2, 20]
- Numero minimo di samples per dividere un nodo = [2, 5, 10, 20, 30, 50, 100]
- Numero minimo di samples per creare una foglia = [1, 5, 10, 20, 30, 50, 100]
- Massimo numero di foglie (intorno alle 20) = [10, 15, 20, 25, 30]



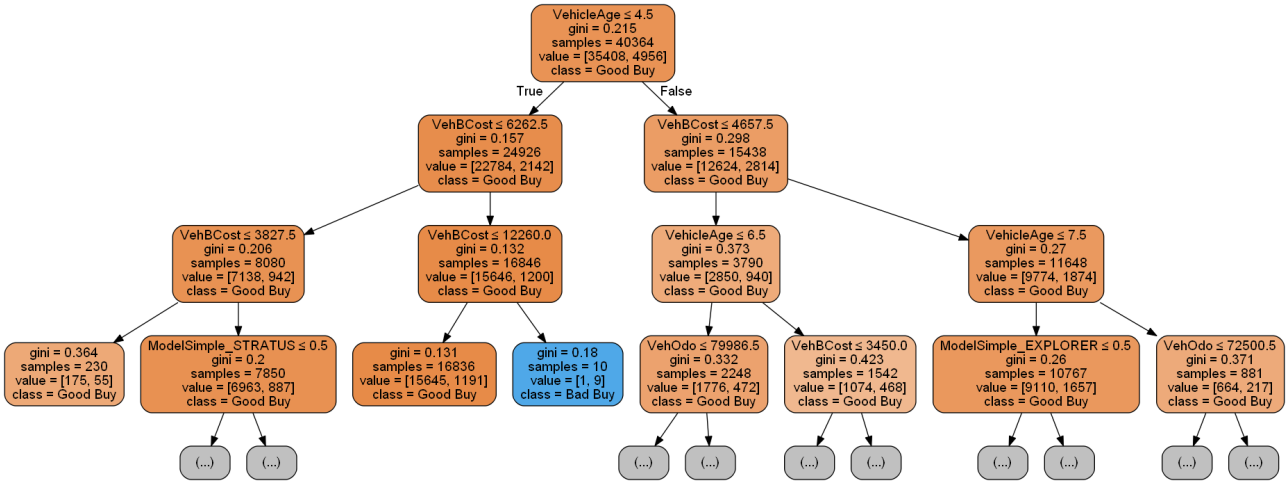
Abbiamo quindi provato questi valori testando le misure Gini e Entropy sia con il dataset bilanciato che quello sbilanciato e abbiamo ottenuto ciò:

	Accuracy	Recall	F1
Gini Bilanciato	0.6079 (+/- 0.03)	0.6634 (+/- 0.12)	0.6053 (+/- 0.03)
Gini Sbilanciato	0.8772 (+/- 0.00)	0.9991 (+/- 0.00)	0.9351 (+/- 0.00)
Entropy Bilanciato	0.6068 (+/- 0.03)	0.6151 (+/- 0.09)	0.6062 (+/- 0.03)
Entropy Sbilanciato	0.8772 (+/- 0.00)	0.9991 (+/- 0.00)	0.9342 (+/- 0.00)

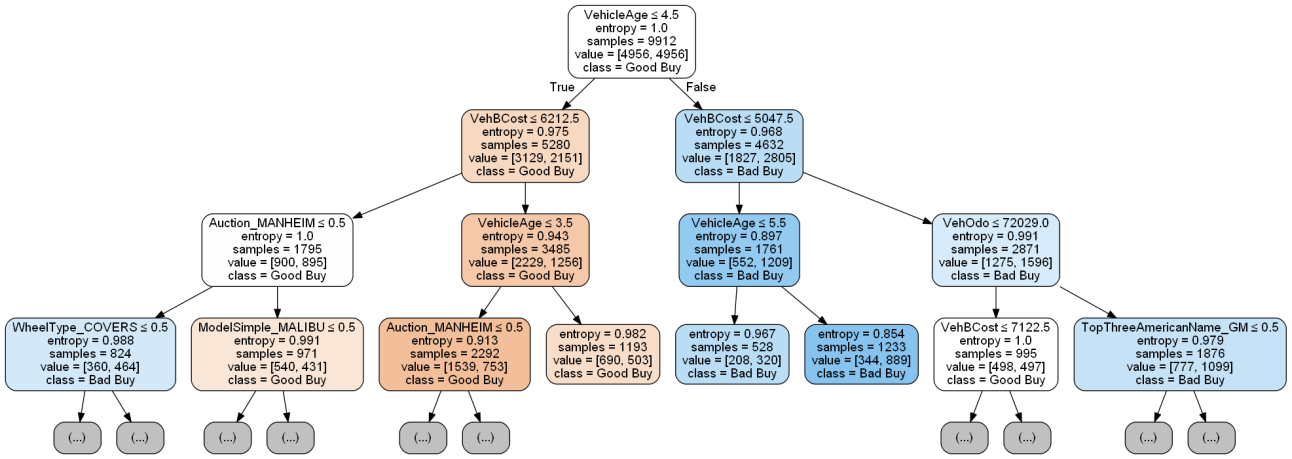
6.3 Interpretazione di vari alberi

Possiamo fare alcune considerazioni da questi risultati: il nostro dataset ha una classe maggioritaria che copre circa l'87% dei record, quindi il dataset è parecchio sbilanciato. Notiamo che i modelli con accuracy più bassa hanno un valore di recall e F1 simili a l'accuracy.
Mentre gli alberi risultati da Gini ed Entropy sono molto simili, gli alberi che si creano con i due diversi dataset sono parecchio diversi, quindi scegliamo il Gini per l'approccio col dataset bilanciato e la Entropy per l'approccio col dataset sbilanciato.

Dataset Sbilanciato



Dataset Bilanciato



Notiamo subito che mentre il primo albero è molto più incline a catalogare i record come Good Buy (perchè è la classe più presente nel dataset), il secondo tende a mettere a destra i BadBuy e a sinistra i GoodBuy usando come criterio principale VehicleAge, che già nella fase di DataUnderstanding sembrava essere molto importante, infatti se andiamo a vedere quanto ogni attributo è importante per il GainRatio, notiamo che l'età del veicolo sembra essere la prima cosa da notare, a seguire il costo, poi l'odometria e infine una serie di attributi categorici.

Dataset Sbilanciato		Dataset Bilanciato	
Attributo	Importanza	Attributo	Importanza
VehicleAge	0.5858	VehicleAge	0.5074
VehBCost	0.2705	VehBCost	0.2008
VehOdo	0.0644	VehOdo	0.0626
ModelSimple_EXPLORER	0.0316	Auction_MANHEIM	0.0585
ModelSimple_SORENTO	0.02758	WheelType_COVERS	0.0497
ModelSimple_STRATUS	0.01983	Make_CHEVROLET	0.0188

6.4 Pruning

La decisione di tagliare il decision tree può essere presa prevalentemente per prevenire l'overfitting del modello. Il pruning dell'albero tuttavia comporta un notevole svantaggio in questo caso: le previsioni di cattivi acquisti diminuiscono sensibilmente, diventando solo 6 rispetto al modello non prunato che ne conta più di 50. Viceversa, il pruning dell'albero offre un miglio-

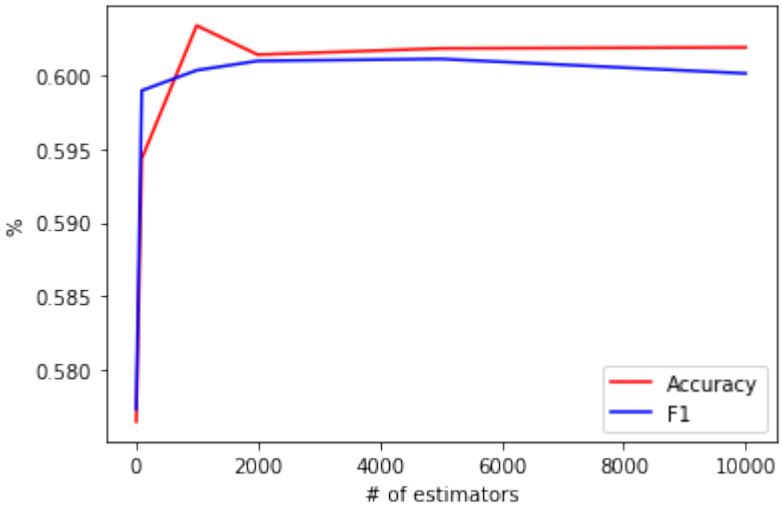
ramento dell'accuratezza del modello dello 0.2% (raggiunge 0.8796).

6.5 Altri metodi di classificazione

Altri metodi che abbiamo usato includono RandomForest, KNN, NaiveBayes e SVM

Random Forest

Abbiamo provato sia col dataset bilanciato che col dataset sbilanciato variando il numero di classificatori tra [10, 100, 1000, 2000, 5000, 10000]. Con il dataset bilanciato abbiamo ottenuto la curva riportata a destra facendo 2-fold cross validation (in quanto i calcoli erano già abbastanza pesanti).



Dalla curva possiamo osservare come l'accuracy e l'F1 non cambiano molto rispetto al numero di stimatori, e che un numero di stimatori intorno a 1000 probabilmente è il preferibile. A destra i risultati.
Per quanto riguarda il dataset sbilanciato, i risultati ottenuti variando il numero di stimatori è molto piccola, quindi abbiamo provato con 2000 stimatori (seconda tabella).

	Accuracy	F1
Giny	0.6042 (+/- 0.02)	0.6004 (+/- 0.03)
Entropy	0.5999 (+/- 0.03)	0.6013 (+/- 0.04)

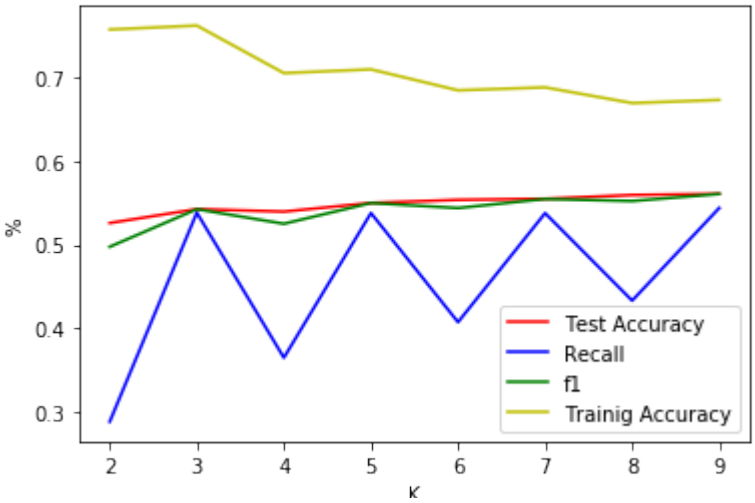
Dataset bilanciato

	Accuracy	F1
Giny	0.8718 (+/- 0.00)	0.4894 (+/- 0.01)
Entropy	0.8718 (+/- 0.00)	0.4888 (+/- 0.01)

Dataset sbilanciato

KNN

Per quanto riguarda il K-Nearest-Neighbor, abbiamo provato a variare il valore di K tra 2 e 10, in quanto provando con valori superiori abbiamo ottenuto risultati molto peggiori. (Secondo grafico).



Notiamo che il valore di recall varia molto dal fatto che K sia pari o dispari, che l'accuracy e l'F1 sono abbastanza costanti e la training accuracy tende a decrescere all'aumentare di K, quindi il valore che ottimizza tutte le misure è K=3 che con la 10-Fold-Validation ci dà i seguenti risultati:

Accuracy	Recall	F1
0.6073	0.6073	0.6073

Naive Bayes

Anche qua abbiamo provato con dataset bilanciato e dataset non bilanciato, quindi nel primo caso il prior delle due classi e 0.5 e 0.5 mentre nel secondo caso è 0.87 e 0.13. Questi sono i risultati che abbiamo ottenuto:

	Accuracy	Recall	F1
Dataset bilanciato	0.6079	0.4910	0.5423
Dataset sbilanciato	0.8669	0.0590	0.5132

SVM

Il tempo di addestramento per le Support Vector Machine supera di gran lunga le nostre aspettative, tanto che siamo riusciti a trainare un solo modello con kernel lineare e dataset bilanciato il quale ci ha dato i seguenti risultati

Accuracy	Recall	F1
0.6018 (+/- 0.02)	0.6143 (+/- 0.03)	0.6018 (+/- 0.02)

Scelta del miglior modello e testing

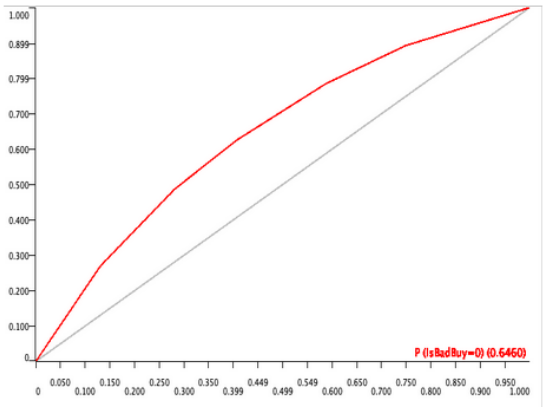
Come miglior modello, scegliamo quello che ci ha dato il punteggio F1 più alto degli altri, ovvero il decision tree basato sul gini index, quindi carichiamo il test set e lo prepariamo in modo che abbia le stesse feature del training set, e questi sono i risultati ottenuti:

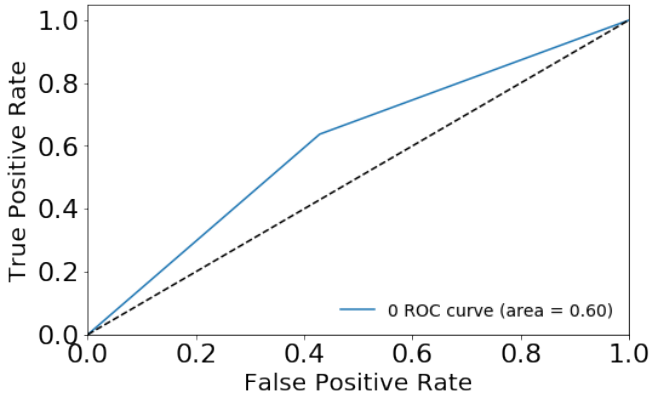
	Precision	Recall	F1	Support
0 (Good buy)	0.881	0.995	0.935	12829
1 (Bad buy)	0.471	0.032	0.06	1768

Accuracy	0.88
F1	0.93

	Predicted Good Buy	Predicted Bad Buy
Actual Good Buy	12681	65
Actual Bad Buy	1710	58

Se andiamo ad osservare la AUC e la ROC, vediamo che la AUC ottenuta con il dataset sbilanciato è 0.646, ovvero la curva in rosso nel grafico, mentre con il dataset sbilanciato quello che otteniamo è una AUC del 0.6





7. TEMPI DI TRAINING E TESTING

	Tempo di training	Tempo di testing
KNN	0.1715	0.6432
Decision Tree	1.0571	0.0269
NB	0.0877	0.0498
Random Forest (2000 stimatori)	53.0341	2.2454

I tempi sopra sono espressi in secondi. Possiamo notare che alcuni modelli hanno un tempo di training molto basso, come nel caso del KNN nel quale il training consiste solamente nell'aggiungere i dati in una tabella o nel Naive Bayes, in cui vengono calcolate le probabilità una sola volta per tutto il dataset, mentre Decision Tree impiega più tempo per calcolare i possibili guadagni di entropia/gini ad ogni nodo e random forest deve fare lo stesso con 2000 stimatori, anche se con dei DT molto più semplici. Nel caso del testing, KNN impiega un tempo maggiore in quanto deve calcolare tutte le distanze dagli altri attributi per trovare quelli più vicini, i DT hanno la proprietà di essere molto veloci nel testing in quanto per testare un attributo, questo deve passare in un numero di nodi che è al più uguale al numero dell'altezza dell'albero. Naive Bayes si tratta semplicemente di applicare la formula di probabilità condizionata e il tempo maggiore risulta essere la random forest, probabilmente per il numero abbastanza elevato di stimatori che devono poi essere combinati insieme.

8. EM CLUSTERING

Abbiamo provato ad applicare l'EM clustering usando il BayesianGaussianMixture, il quale tenta di stimare il numero migliore di cluster, possibilmente eliminandone alcuni durante l'esecuzione dell'algoritmo. Abbiamo provato con i diversi tipi di covarianza ['full', 'tied', 'diag', 'spherical'] e un numero di componenti compreso tra 2 e 20, poi abbiamo selezionato solo i modelli che hanno dato una convergenza e ne abbiamo analizzato il SilhouetteScore, ottenendo i seguenti risultati, che risultano comunque peggiori degli approcci usati sopra (il numero dopo il tipo di covarianza indica il numero di componenti ottenute:

Full 2	Tied 2	Tied 3	Tied 4	Tied 6	Spherical 4
0.2848	0.2969	0.2475	0.2625	0.1738	0.2854