

Data Science

Analyse eines E-Commerce Datensatzes

eingereicht von:

Luca Rettenberger, Mario Amann, Felix Hamburger, Tobias
Schmähling, Martin Lanz

July 28, 2021

Dozent: Prof. Dr. Höpken

Contents

1 Business Understanding	3
1.1 Motivation	3
1.2 Zielsetzung	3
2 Data Understanding	5
2.1 EDA	5
2.2 Probleme und notwendige Transformationen	14
2.3 Tabellarische Darstellung der Attribute	16
3 Kausalmodell	17
3.1 Korrelationsmatrix	19
4 Data Preparation	21
5 Modellierung	22
5.1 Supervised Learning	22
5.1.1 Naïve Bayes	22
5.1.2 Support Vector Machine	31
5.1.3 Neuronale Netze	37
5.1.4 Lineare Regression	40
5.1.5 Polynominal Regression	42
5.1.6 Logistische Regression	43
5.1.7 C4.5 Decision Tree Algorimus	45
5.1.8 Classification and Regression Trees (CART)	46
5.1.9 Random Forest	49
5.1.10 Rule Induction	50
5.2 Unsupervised Learning	53
5.2.1 Apriori-Algorithmus	53
5.2.2 Frequent Pattern Growth	56
5.2.3 K-Means	58
5.2.4 K-Medoid	62
5.2.5 Kernelized K-Means	66
5.2.6 Agglomerative Clustering	69
5.2.7 Density-Based Spatial Clustering of Applications with Noise	71
5.2.8 Self-organizing Map	74
6 Zusammenfassung	77
Literatur	79

1 Business Understanding

Um als Unternehmen konkurrenzfähig zu bleiben ist es heutzutage unvermeidbar Geschäftsprozesse zu überwachen, Kundendaten zu sammeln und auf diese umfassende Analysen anzuwenden. Hierdurch lassen sich mögliche Probleme erkennen und zukünftige Entwicklungen voraussagen.

Das Projekt wird anhand des CRISP-DM[WH00] Models erläutert und die verschiedenen Unterthemen analytisch aufgearbeitet.

1.1 Motivation

Für das Projekt wird ein Datensatz zur Verfügung gestellt, der Informationen zu Kunden einer Online-Plattform, demographischen Eigenschaften, vergangenem Kaufverhalten und Kündigungen enthält.

Durch eine Analyse der Daten können Verhaltensmuster bestimmter Kunden respektive Kundengruppen erkannt werden. Zudem lassen sich Trends und Prognosen feststellen, auf die das Unternehmen frühzeitig reagieren kann um eine optimale User Experience zu gewährleisten.

Verschiedenste Methodiken geben Aufschluss über versteckte Informationen in den Daten. Hierfür kommen verschiedenste Mittel zum Einsatz:

- Regression
- Classification
- Clustering

1.2 Zielsetzung

Im Laufe der Arbeit sollen die durch das Kausalmmodell - welches in späteren Kapiteln behandelt wird - definierten Zielvariablen **Churn**, **Complain** und **Cash-back Amount** vorhergesagt und analysiert werden.

Über die vorhergesagten Werte sollen dann im späteren Verlauf der Arbeit folgende Fragen betrachtet und beantwortet werden.

- Welche Gründe gibt es für die Abwanderung aus dem Unternehmen?
- Gibt es Kundengruppen die sich besonders häufig beschweren?
- Welche Kundengruppen sind dem Unternehmen sehr lange treu?
- Welche Attribute sind von größter Bedeutung für einen hohen Cashback Amount?

- Gibt es interessante Zusammenhänge um auf Kundenverhalten zu schließen?

2 Data Understanding

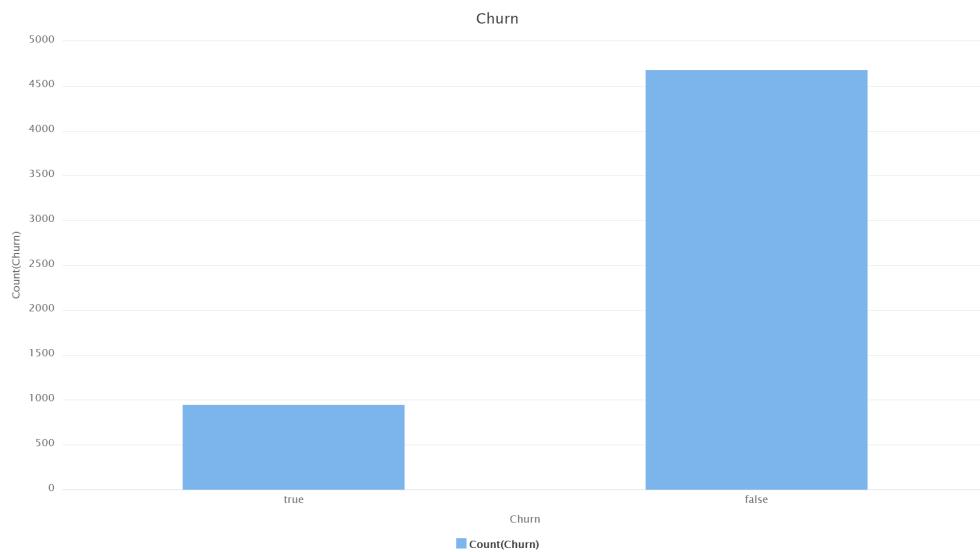
Im Datensatz befinden sich 5630 Datenpunkte mit je 20 Attributen. In der Tabelle 1 werden die 20 Attribute näher beschrieben indem der Datentyp, eine Beschreibung der Attribute und die Anzahl ihrer Fehlwerte aufgeführt werden.

2.1 EDA

Im Folgenden wird näher auf die Attribute und deren Verteilungen eingegangen. Fehlwerte oder fehlende Einheiten werden vorerst nicht beachtet und später im Kapitel Probleme und notwendige Transformationen weiter behandelt.

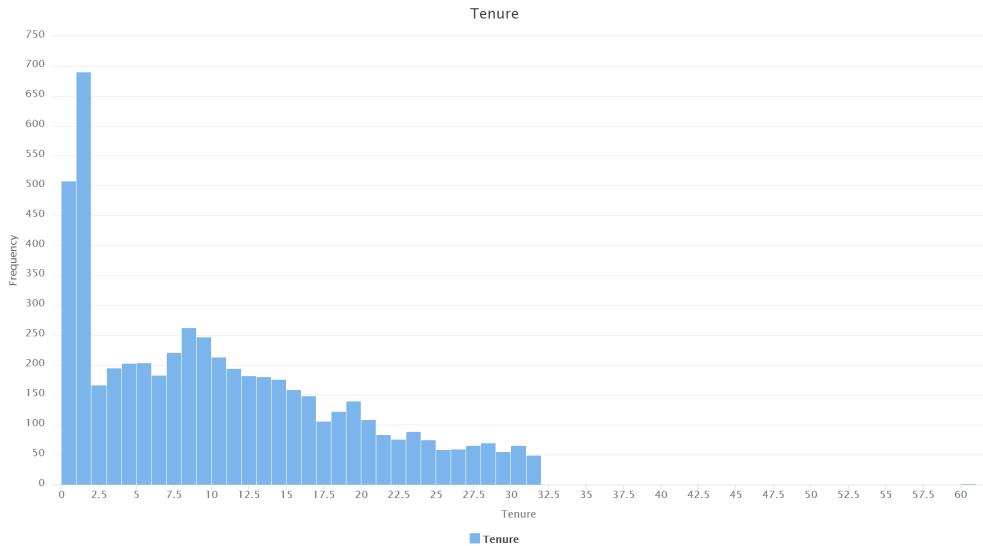
CustomerID Die CustomerID ist die einzigartige Identifikationsnummer eines Kunden. Es liegt nahe, dass die Identifikationsnummer keinen Einfluss auf das Kaufverhalten des einzelnen Kunden besitzt. Aus diesem Grund wird dieses Attribut bei der Modellierung nicht verwendet.

Churn Das Churn Attribut ist ein binominales Attribut, welches angibt, ob ein Kunde abgewandert ist. Im vorhandenen Datensatz ist hierbei eine deutliche Mehrheit der nicht abgewanderten Kunden sichtbar. Deren Anzahl liegt mit 4682 Kunden deutlich höher als die abgewanderten Kunden mit 948.

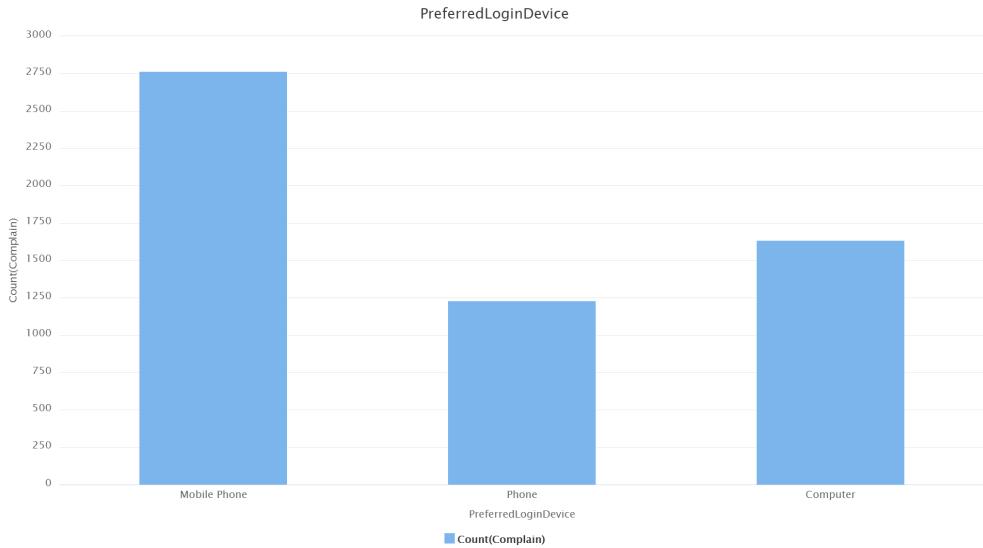


Tenure Tenure gibt die Zeit an, die ein Kunde dem Unternehmen treu ist. Es handelt sich hierbei um einen Integerwert, welcher im Datensatz eine Spannweite von 0 bis 61 besitzt. Der Durchschnitt ist 10.190 mit einer Standardabweichung von 8.557. In der Grafik wird deutlich, dass ein Fünftel der Datenpunkte im Bereich 0 und 1 sind und diese zwei Werte damit die zwei Mengen mit den meisten

Datenpunkten sind. Mit Blick auf die Visualisierung zeigt sich, dass alle Datenpunkte außer zwei in der Spannweite von 0 bis 31 sind. Diese zwei Datenpunkte liegen bei 61 und sind somit als Ausreißer zu betrachten.

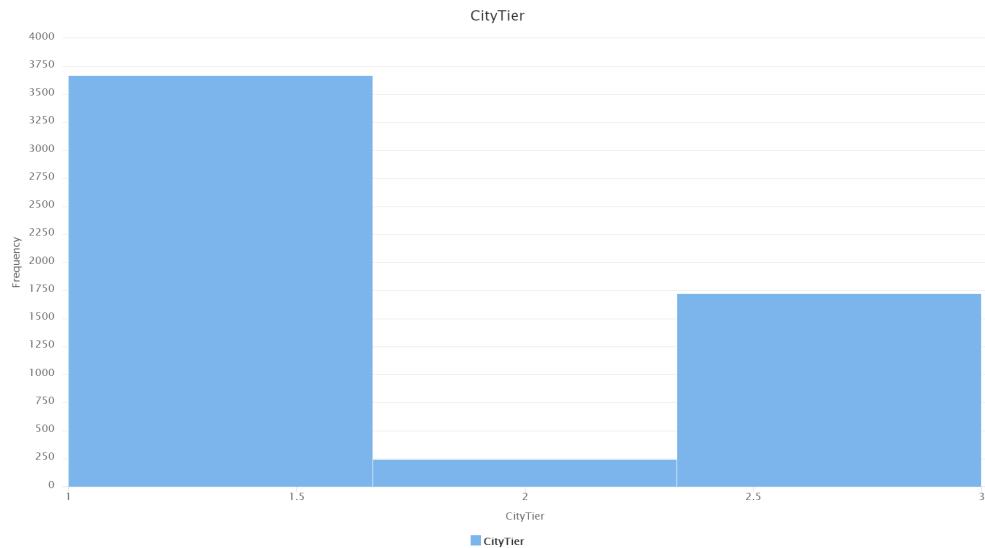


PreferredLoginDevice Das Attribut PreferredLoginDevice ist ein nominal und gibt das präferierte Einlog Device eines Kunden an. Hierbei handelt es sich um die drei Klassen Mobile Phone, Computer und Phone. In der Visualisierung wird deutlich, dass fast 50 Prozent der Kunden der Klasse Mobile Phone angehören, etwa 30 Prozent der Klasse Computer und etwa 20 Prozent der Klasse Phone.

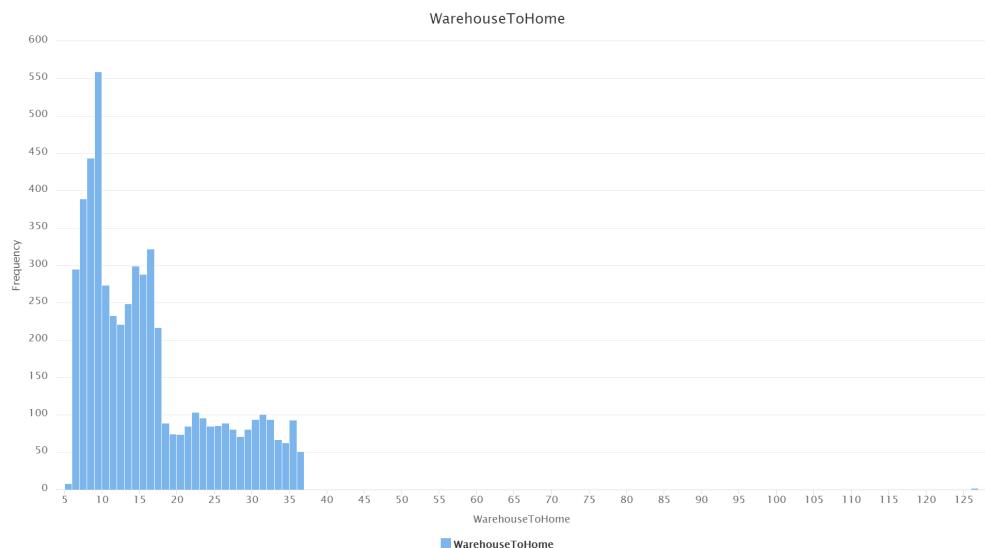


CityTier Im Attribut CityTier werden die Städte des Kunden nach Größe und Wohlstand in Klassen aufgeteilt. Die Visualisierung zeigt deutlich, dass mit 3666

Kunden die meisten der Klasse 1 und mit 1722 Kunden die zweit meisten der Klasse 3 zugeordnet werden. Die wenigensten werden der Klasse 2 zugeordnet.

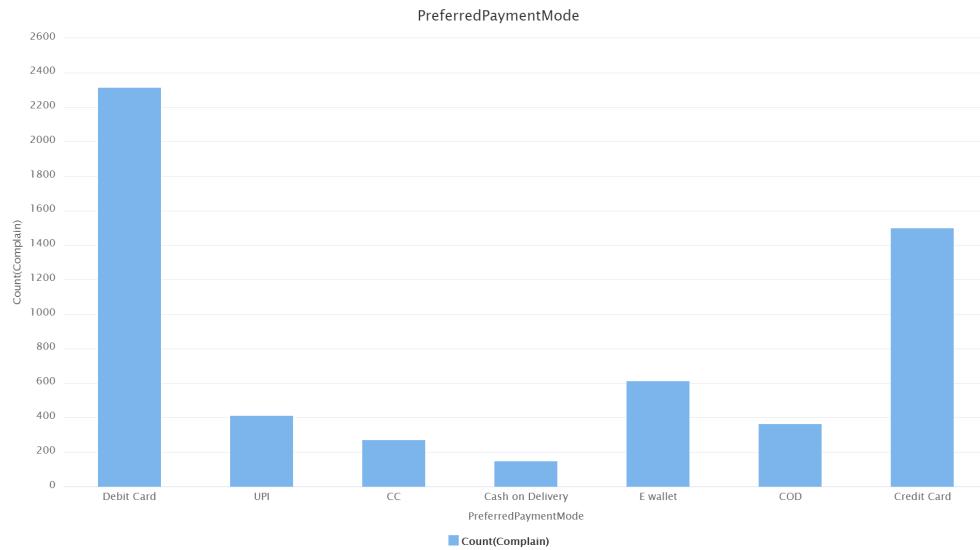


WarehouseToHome Mit WarehouseToHome wird die Distanz eines Kunden zum Warenhaus angegeben. Die kürzeste Distanz liegt hierbei bei 5 und die größte bei 127. Die durchschnittliche Distanz liegt bei 15.640 mit einer Standardabweichung von 8.531. In der Visualisierung wird deutlich, dass die Distanz zum Warenhaus der meisten Kunden unterhalb des Wertes 37 liegt und somit die Kunden im Bereich 127 als Ausreißer im Datensatz zu betrachten sind.

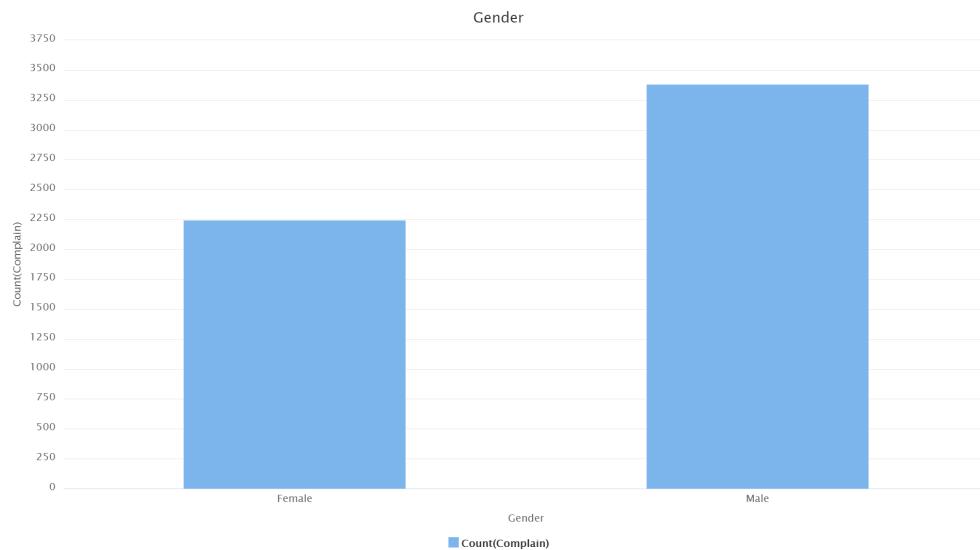


PreferredPaymentMode Ein weiteres nominales Attribut ist die präferierte Zahlmethode eines Kunden. Die Klasse der Debit Card besitzt hierbei über 41 Prozent

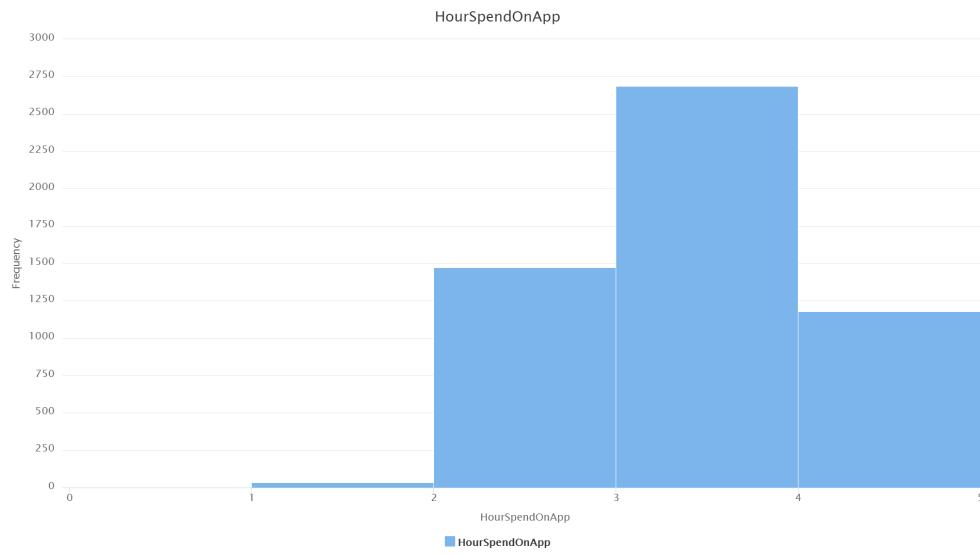
der Kunden. Danach beinhaltet die Klasse Credit Card mit etwa 27 Prozent die zweitmeisten Kunden. Die anderen fünf Klassen beziehen maximal 11 Prozent der Kunden ein. Diese Verteilung der präferierten Zahlmethode wird in der Visualisierung deutlich.



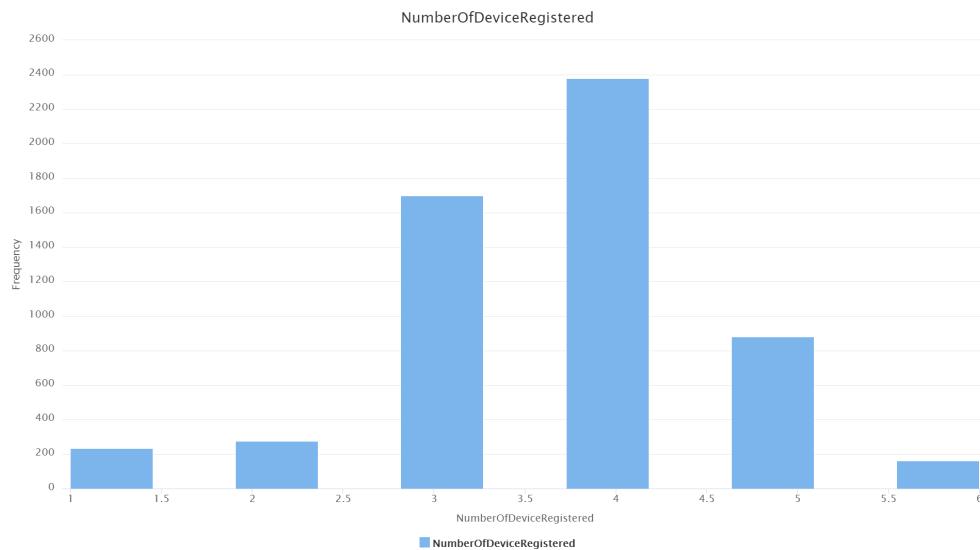
Gender Das Geschlecht eines Kunden wird im Attribut Gender aufgeführt. Mit 3384 sind unter den Kunden mehr Männer als Frauen anzutreffen.



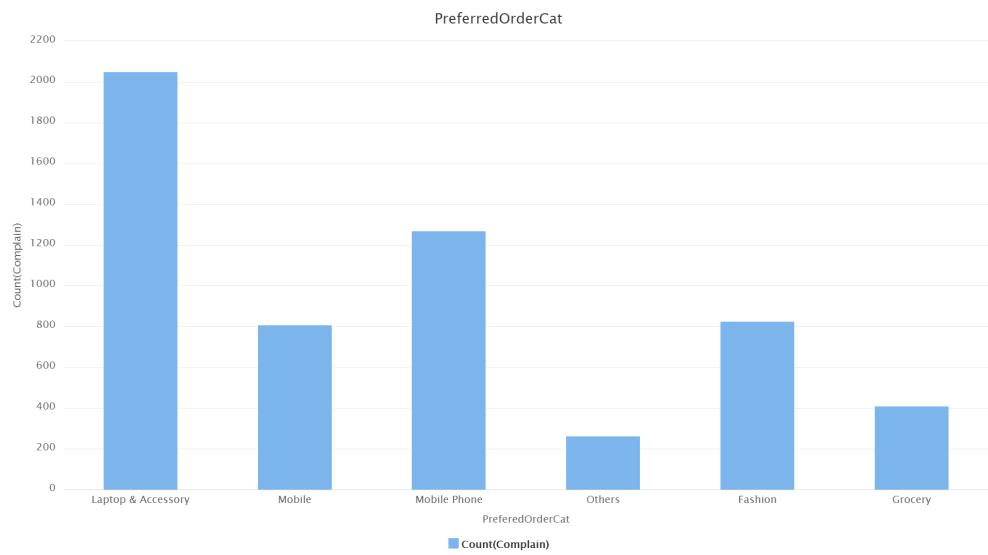
HourSpendOnApp Die Anzahl an Stunden, die ein Kunde auf der App verbracht hat wird im Attribut HourSpendOnApp aufgeführt. Durchschnittlich verbringen Kunden 2.932 Stunden auf der App, wobei der Kunde mit der längsten Verweildauer 5 Stunden auf der App verbracht hat.



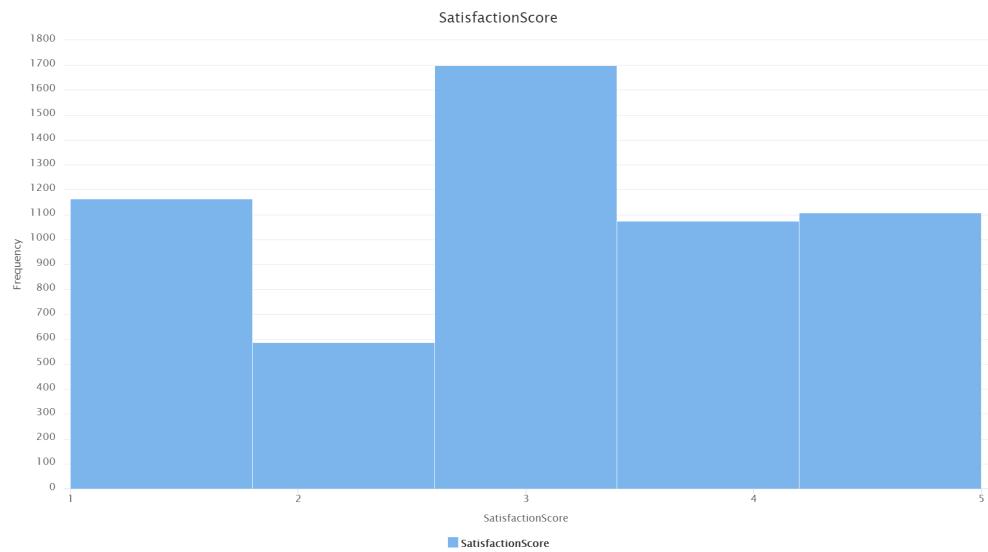
NumberOfDeviceRegistered Das Attribut NumberOfDeviceRegistered gibt an, wie viele Devices ein Kunde registriert hat. Durchschnittlich hat ein Kunde 3.689 Devices eingetragen. Die meisten Kunden haben sich mit 3 oder 4 Devices angemeldet.



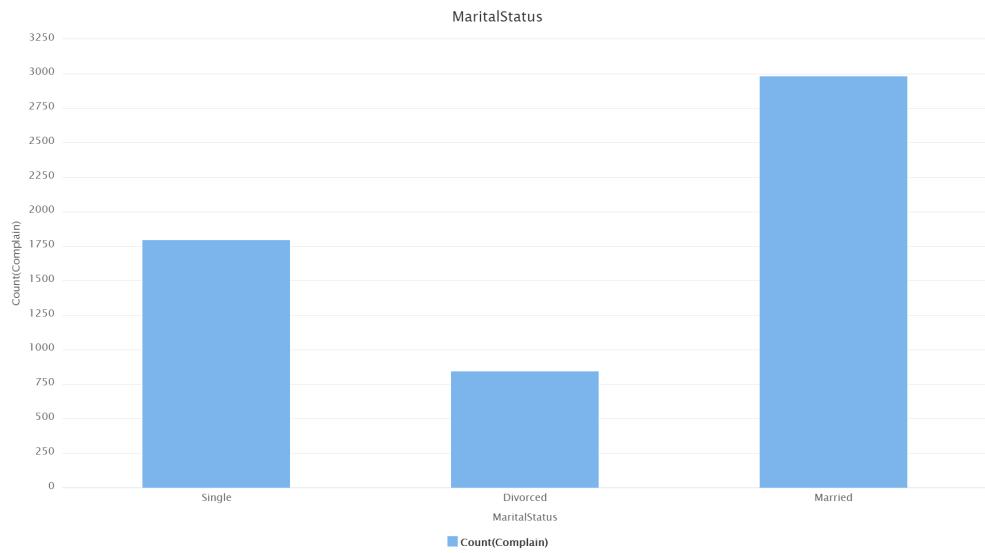
PreferredOrderCat Die präferierte Bestellkategorie ist ein weiteres nominales Attribut. Mit 36 Prozent ist die am meisten präferierte Kategorie Laptop & Accessory. Die Kategorie Mobile Phone ist die zweitmeiste Bestellkategorie mit etwa 23 Prozent. Unter 5 Prozent der Kunden präferieren die Kategorie Others.



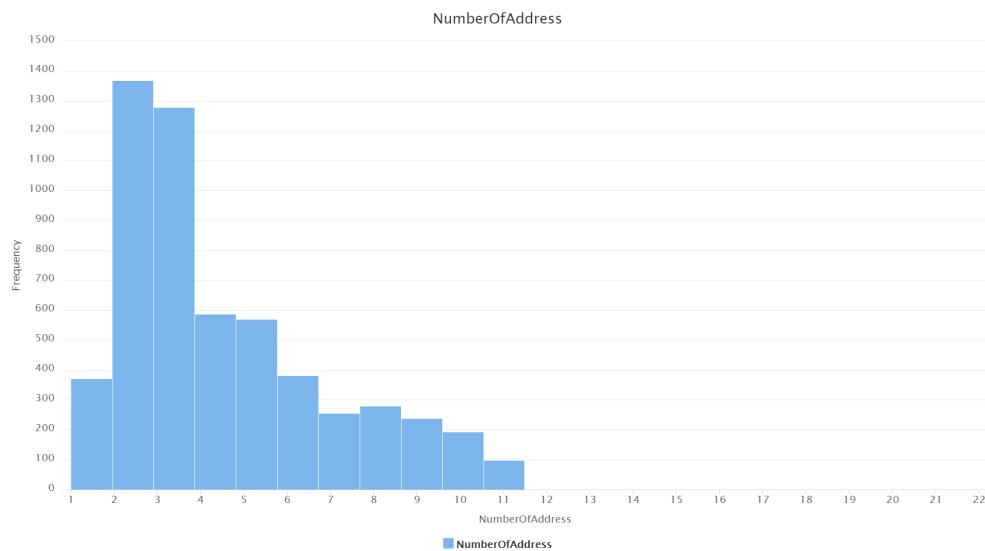
SatisfactionScore Der SatisfactionScore gibt die Zufriedenheit eines Kunden an. Der Wertebereich dieses Attributs reicht von 1 bis 5. Der Durchschnitt liegt bei 3.067. In der Visualisierung ist zu erkennen, dass es mehr Datenpunkte mit einem SatisfactionScore von 3 gibt und die wenigsten mit einem Score von 2. Die Klassen 1, 4 und 5 sind dagegen sehr ausgewogen.



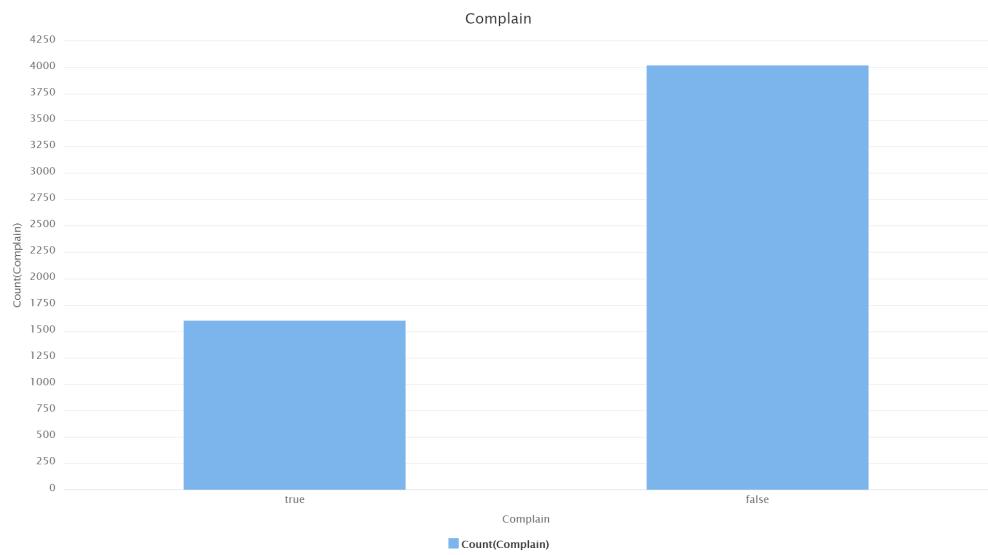
MaritalStatus Hinter dem MaritalStatus Attribut verbirgt sich der Beziehungsstatus eines Kunden. Es ist ein nominales Attribut mit den nominalen Werten Married, Single und Divorced. Die Mehrheit der Datenpunkten gehört zur Klasse Married mit 53 Prozent. Etwa 32 Prozent der Kunden sind Single und 15 Prozent geschieden.



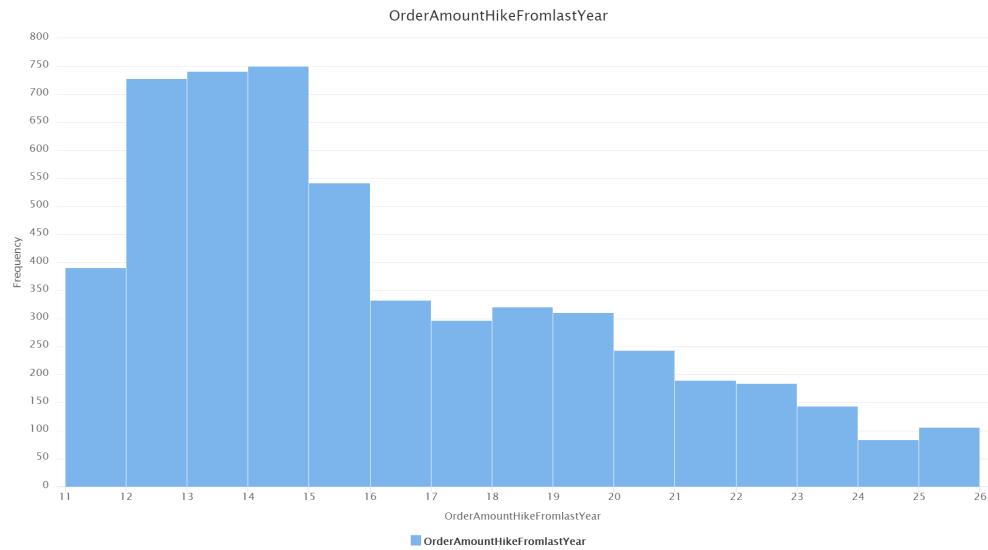
NumberOfAddress Die Anzahl an Adressen eines Kunden hat einen Wertebereich von 1 bis maximal 22 in diesem Datensatz. Der Durchschnitt beträgt 4.214. Mit Blick auf die Visualisierung sieht man deutlich, dass 2 und 3 Adressen am häufigsten vorkommen. Es haben beinahe alle Kunden 11 oder weniger Adressen. Somit sind die vereinzelten Werte um 20 Adressen als Ausreißer anzusehen.



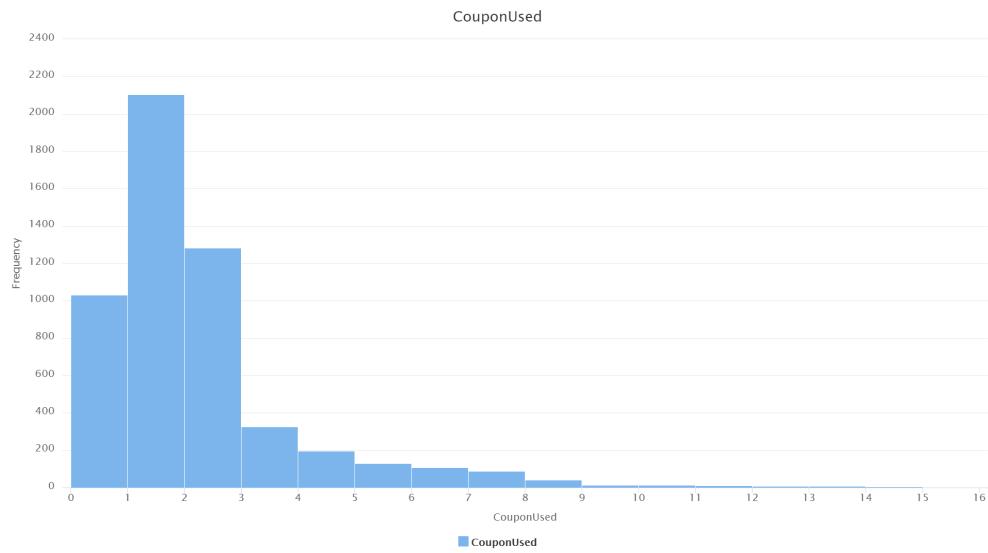
Complain Das Attribut Complain gibt an, ob sich ein Kunde im letzten Monat beschwert hat. Etwa 28 Prozent der Kunden haben sich im letzten Monat beschwert.



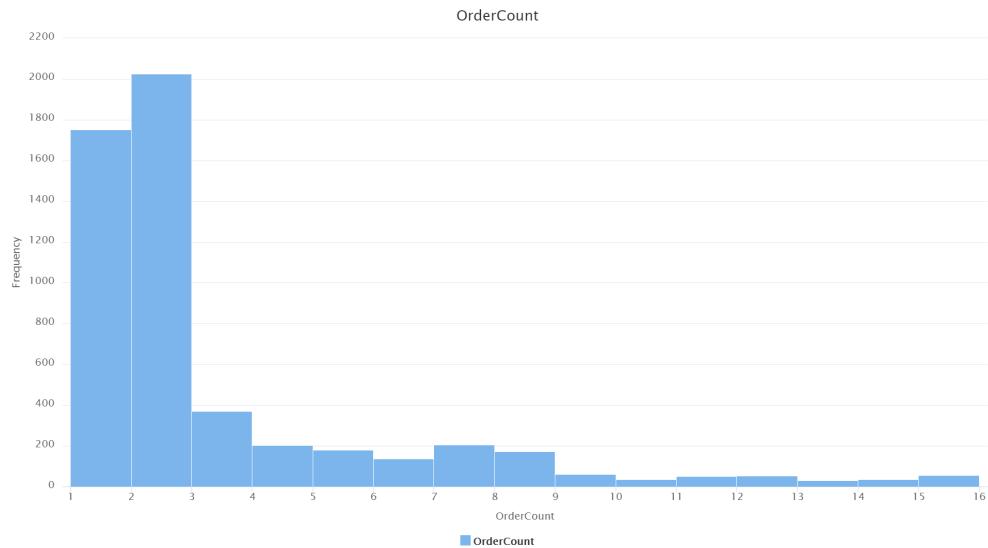
OrderAmountHikeFromLastYear Dieses Attribut gibt die prozentuale Steigerung der Kaufaufträge im Vergleich zum Vorjahr an. Das Minimum war hierbei 11 Prozent und das Maximum bei 26 Prozent. Durchschnittlich wurde eine Steigerung von 15.708 Prozent gemessen.



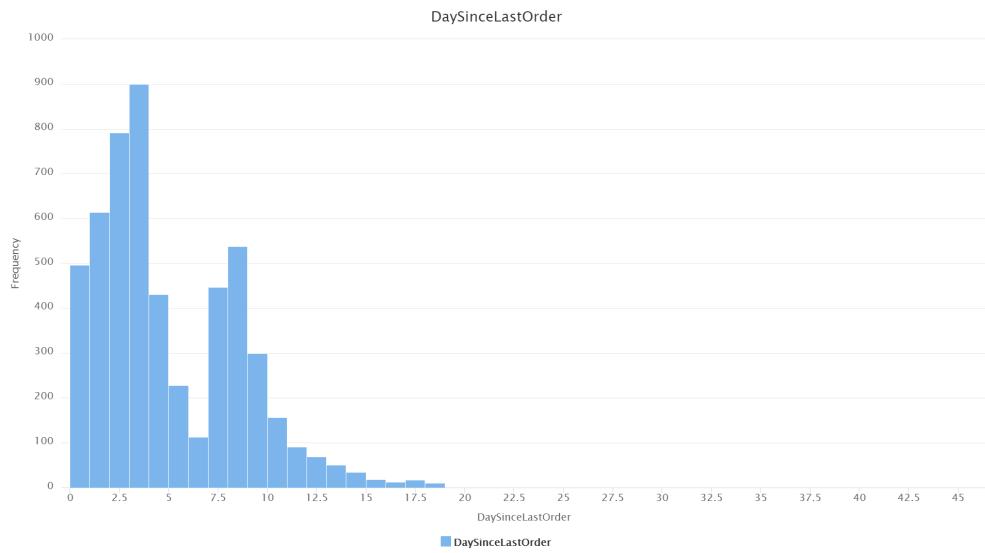
CouponUsed Die Anzahl an genutzten Coupons hat einen Wertebereich von 0 bis 16. Wobei die meisten Kunden keinen bis drei Coupons genutzt haben. Dies wird auch durch den Durchschnitt von 1.751 und der Standardabweichung von 1.895 deutlich.



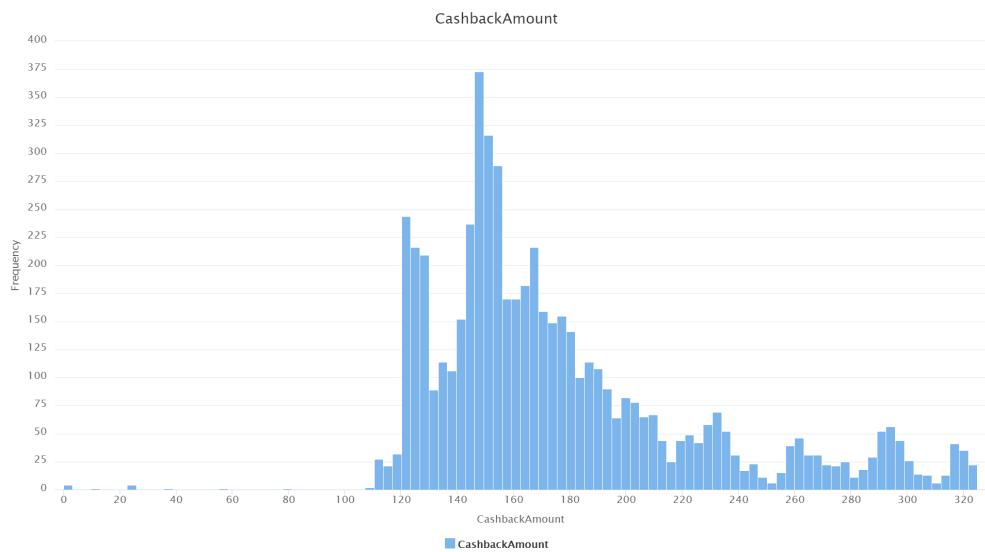
OrderCount OrderCount gibt die Anzahl an Bestellungen an. Wie in der Visualisierung zu sehen hat die Mehrheit der Kunden 1 oder 2 Bestellungen aufgegeben. Der Durchschnitt der Bestellungen liegt bei 3.008 mit einer Standardabweichung von 2.940



DaySinceLastOrder Das Attribut DaySinceLastOrder gibt die Tage seit der letzten Bestellung an. Zwar ist der maximale Wert 46, allerdings sind fast alle Datenpunkte unter 18. Somit sind die Datenpunkte um den Wert 46 als Ausreißer anzusehen. Die Visualisierung zeigt eine überraschende Kurve. Zunächst besitzt das Histogramm einen maximalen Wert um den Wert 3, dann bei 6 ein lokales Minimum und bei 8 ein lokales Maximum.



CashbackAmount Der CashbackAmount gibt den durchschnittlichen Cashback eines Kunden im letzten Monat an. Dieser Wert ist das einzige Attribut mit dem Datentyp Real. Der niedrigste Wert ist 0 und der maximale Cashback eines Kunden liegt bei 324.990, wobei fast alle Kunden einen Cashback oberhalb von 109 erreichten. Die meisten Kunden befinden sich im Wertebereich von 119 bis etwa 215 mit einem lokalen Maximum bei etwa 123 und einem globalen Maximum bei 148.



2.2 Probleme und notwendige Transformationen

Es wurde zunächst untersucht, welche Datenpunkte Fehlwerte beinhalten. Durch die ähnliche Anzahl an Fehlerten pro Attribute lag die Annahme nahe, dass einige wenige Datenpunkte geben könnte, die viele Attribute mit Fehlerten

besäßen. Nach der Untersuchung der Fehlwerte wurde diese Annahme jedoch nicht nur widerlegt, sondern jeder Datenpunkt mit Fehlwert hat genau nur ein Fehlwert, weshalb von den insgesamt 5630 Datenpunkten 1856 Datenpunkte Fehlwerte aufweisen. Mit Blick auf die Behandlung dieser Fehlwerte ist daher eine Entfernung eine unpraktische Methode, da somit ein Drittel der Informationen des Datensatzes verloren gehen würden. Die genauen Transformierungen und Preprocessingschritte für diese Fehlwerte wird im Kapitel der allgemeinen Preprocessing behandelt.

Ein weiteres Problem sind die fehlenden genauen Beschreibungen der einzelnen Attribute. Es werden bei einer Vielzahl an Attributen keine Einheiten angegeben. Grundsätzlich ist dies für die Modellierung weniger von belangen, da dort sowieso ein Datenpreprocessing den Einfluss von Einheiten - wenn nötig - eliminiert. Allerdings sind durch die fehlenden Einheiten logische Fehler im Datensatz nur schwer zu erkennen (Beispiel: Als Distanz eines Kundenzuhause zum Warenhaus sind 50000 Kilometer Entfernung unrealistisch, während 50000 Meter eine realistische Entfernung wäre). Dieses Problem beinhaltet folgende Attribute:

- Tenure: Im Blick auf die Visualisierung und auf die Skala der Werte von 0 bis 61 gehen wir hierbei von Monaten aus.
- CityTier: Die Skala der Werte bei diesem Attribut ist von 1 bis 3. Bei den meisten CityTier Listen wird die 1 als die größere Stadt mit einem größeren Wohlstand kategorisiert.
- WarehouseToHome: Die meisten Werte bei diesem Attribut liegen zwischen 0 und 36. Der maximale Wert bei 127. Dies schließt darauf, dass diese Werte Kilometer, Meilen oder Einheiten mit ähnlicher Maßeinheit sein könnte. Wir werden im Folgenden davon ausgehen, dass damit Kilometer gemeint ist.
- CashbackAmount: Es ist nicht ersichtlich, welche Einheit hier verwendet wird. Die Größenordnung der Zahlen lässt zumindest vermuten, dass Euro und Dollar eher unrealistisch wären.

Ebenso ist die fehlende Beschreibung des Datensatzes ein weiteres Problem. Es wäre gut zu wissen, in welchem Zeitraum die Daten aufgenommen wurden. Wurde der Datensatz beispielsweise in einem Monat aufgenommen, in dem es Ausfälle des Systems gab würde dies die Attribute beeinflussen. Zum Beispiel würde dies vermehrt zu Beschwerden von Kunden führen und das Attribut Complain hätte ein zusätzliches Bias. Dies würde dann die Analysen beeinflussen und gegebenenfalls zu falschen Schlüssen führen.

Des Weiteren hat die Data Understanding gezeigt, dass in den polynomiellen Attributen PreferredPaymentMethod, PreferredOrderCat und PreferredLoginDevice Abkürzungen und teilweise sprachliche Synonyme auftauchen. Dies beinhaltet die Ähnlichkeit von "Mobile" und "Mobile Phone" in der präferierten Kaufkat-

egorie und im präferierten Einlog Device mit "Mobile Phone" und "Phone". Noch deutlicher ist dies in der präferierten Zahlmethode, dort gibt es die nominellen Werte "CC" und "Credit Card", sowie "COD" und "Cash on Delivery". Die Abkürzungen könnten darauf hinweisen, dass mit diesen nominellen Werten die gleiche Zahlmethode gemeint ist. Im Gespräch mit Herrn Professor Höpken wurde dies zwar als naheliegend, allerdings nicht ausreichend eingeordnet, da bei einer falschen Zusammenführung Informationen zum Datensatz hinzugefügt werden würden, die nicht der Wahrheit entsprechen könnten. Daher werden die einzelnen nominellen Werte belassen wie sie sind.

2.3 Tabellarische Darstellung der Attribute

tab. 1 Tabellarische Darstellung der Attribute

Attribut	Datentyp	Beschreibung	Fehlwerte
CustomerID	Integer	Einzigartige Identifikationsnummer eines Kunden	-
Churn	Binominal	Abwanderung eines Kunden	-
Tenure	Integer	Zeit, die ein Kunde dem Unternehmen treu ist	264
PreferredLoginDevice	Polynominal	Präferiertes Einlog Device	-
CityTier	Integer	Städte nach Größe und Wohlstand in Gruppen aufgeteilt (1 groß, 3 klein)	-
WarehouseToHome	Integer	Distanz Kundenzuhause zu Warenhaus	251
PreferredPaymentMode	Polynominal	Präferierte Zahlmethode	-
PaymentModeGender	Binominal	Geschlecht des Kunden	-
HourSpendOnApp	Integer	Anzahl Stunden spendiert auf der App	255
NumberOfDeviceRegistered	Integer	Anzahl registrierter Devices	-
PreferredOrderCat	Polynominal	Präferierte Bestellungskategorie eines Kunden im letzten Monat	-
SatisfactionScore	Integer	Zufriedenheit eines Nutzers am Service	-
MaritalStatus	Polynominal	Beziehungsstatus eines Kunden	-
NumberOfAddress	Integer	Anzahl Adressen eines Kunden	-
Complain	Binominal	Kunde hat sich im letzten Monat beschwert	-
OrderAmountHikeFromLastYear	Integer	Prozentuale Steigerung der Kaufaufträge im Vergleich zum Vorjahr	265
CouponUsed	Integer	Anzahl genutzter Coupons eines Kunden im letzten Monat	256
OrderCount	Integer	Anzahl an Kaufaufträgen eines Kunden im letzten Monat	258
DaySinceLastOrder	Integer	Tage seit dem letzten Kaufauftrag	307
CashbackAmount	Real	Durchschnittlicher Cashback eines Kunden im letzten Monat	-

3 Kausalmodell

Für das Kausalmodell wurden alle Attribut untereinander angesehen und auf kausale Zusammenhänge untersucht. Hierbei kam das unten angefügte Kausalmodell zustande. Als Zielvariablen wurden hierbei die binominalen Attribute Churn und Complain ausgewählt und das Attribut CashbackAmount. Diese werden in den späteren Modellen verwendet und weiter untersucht.

Attribute, bei denen kein kausaler Zusammenhang zu den anderen Attribute vermutet werden:

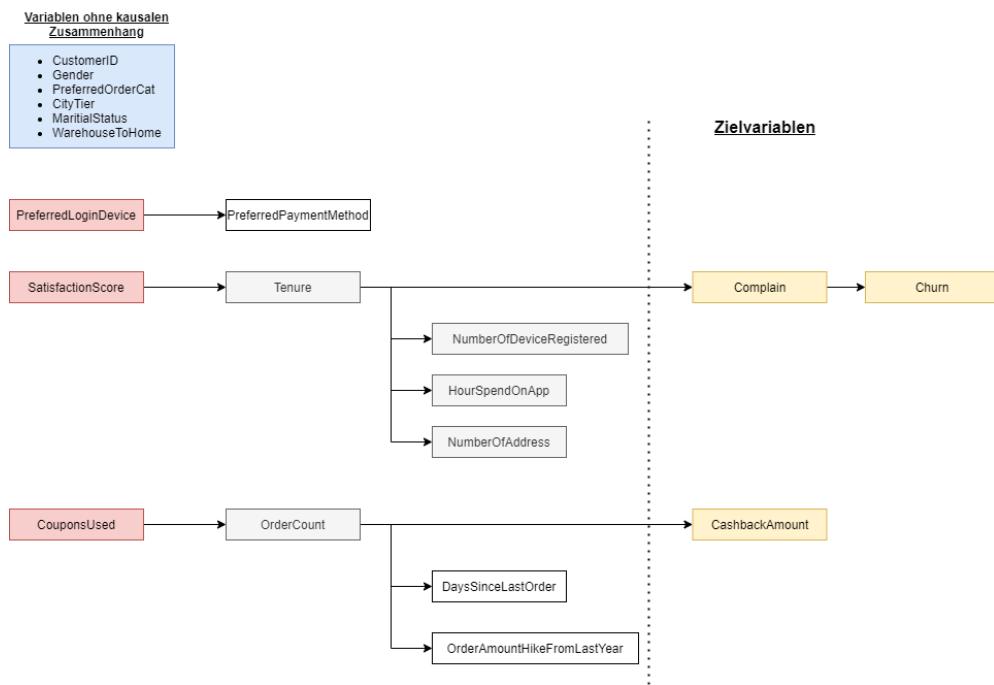
- CustomerID
- Gender
- PreferredOrderCat
- CityTier
- MaritalStatus
- WarehouseToHome

Die erste erstellte Kausalkette betrachtet das präferierte Einlog Device und die präferierte Bezahlmethode. Hierbei wird ein kausaler Zusammenhang vermutet, da ein Computer eher geeignet für Zahlmethoden mit größerem Aufwand ist als ein Mobiltelefon. Die Vermutung basiert auf den kleineren Bildschirm und die eher einfacheren Bedienungen eines Mobiltelefons.

Die größte Kausalkette betrachtet den SatisfactionScore. Die Zufriedenheit eines Kunden wirkt sich dabei auf die Zeit, die ein Kunde dem Unternehmen treu ist aus. Die Annahme ist hierbei, dass zufriedenere Kunden länger einem Unternehmen treu sind als unzufriedene Kunden. Dieses Attribut wirkt sich dann auf die Anzahl an registrierte Adressen, Anzahl an spendierten Stunden und Anzahl an registrierten Geräten aus. Hierbei ist der Gedankengang, dass je länger ein Kunde einem Unternehmen treu ist, desto wahrscheinlicher ist es, dass ein Kunde umgezogen ist oder in der Zwischenzeit ein neues Gerät gekauft hat oder länger die App benutzt hat. Dazu wird vermutet, dass je länger ein Kunde dem Unternehmen treu ist, desto unwahrscheinlicher beschwert er sich. Ob ein Kunde sich beschwert wirkt sich dann auf den Abgang des Kunden aus. Die Annahme lautet dabei, dass Kunden, die unzufrieden sind und sich beschweren wahrscheinlicher dem Unternehmen den Rücken kehren als Zufriedene, die sich nicht beschwert haben.

Die letzte erstellte Kausalkette beginnt mit dem Attribut CouponUsed. Die Vermutung ist hierbei, dass je mehr Coupons ein Kunde benutzt, desto mehr Bestellungen gibt er auf. Die darauffolgende Annahme ist, dass sich die Anzahl an

Bestellungen im letzten Monat sich auf den CashbackAmount im letzten Monat und auf die prozentuale Steigerung der Kaufaufträge im Vergleich zum Vorjahr auswirkt. Und eine hohe Anzahl an Bestellungen im letzten Monat lässt die Vermutung nahe, dass der Kunde in einer höheren Frequenz einkauft und somit die Tage seit der letzten Bestellung geringer ist als bei Kunden, die weniger Bestellungen im Monat aufgeben.



3.1 Korrelationsmatrix

Bei der Korrelationsmatrix werden Attribute paarweise analysiert. Zunächst werden die Korrelationen zu den Zielvariablen Churn, CashbackAmount und Complain untersucht. Für die Korrelation wird der bereinigte Datensatz verwendet.

Churn Wie bereits in dem Kausalmodell erwartet, hat das Attribut Tenure mit -0.336 eine negative Korrelation zu Churn. Kunden, die länger einem Unternehmen treu sind, wandern auch weniger ab. Ebenfalls erwartet war die Korrelation zwischen Churn und Complain mit 0.250. Beschwert sich jemand, so ist es wahrscheinlicher, dass dieser Kunde auch abwandert.

Complain Wie bereits in Churn beschrieben, hat Complain zu diesem Attribut eine positive Korrelation. Ansonsten hat Complain keine hoch korrelierten Attribute. Aus dem Kausalmodell wurde angenommen, dass Tenure sich auf Complain auswirkt. Würde dies stimmen, so müsste dies auch in der Korrelationsmatrix zu sehen sein. Dies lässt darauf schließen, dass die erwartete Kausalität zwischen diesen Attributen nicht besteht.

CashbackAmount Der CashbackAmount hat eine hohe Korrelation zum Attribut Tenure mit 0.462. Die zweithöchste Korrelation teilen sich das Attribut DaySinceLastOrder und OrderCount mit jeweils 0.345 zur Zielvariablen CashbackAmount. Das Attribut CouponUsed hat ebenfalls eine positive Korrelation mit 0.219 zum Attribut CashbackAmount.

Des weiteren gibt es Korrelationen zwischen Inputattributen. In der nachfolgenden Liste wurden alle Korrelationen zwischen Inputattributen größer 0.2 aufgezählt.

- HourSpendOnApp - NumberOfDeviceRegistered - 0.305
- NumberOfAddress - Tenure - 0.233
- OrderCount - CouponUsed - 0.649
- OrderCount - DaySinceLastOrder - 0.451
- CouponUsed - DaySinceLastOrder - 0.314

Attribut...	Churn	Complain	Tenure	CityTier	Wareho...	Gender	HourSp...	Number...	Satisfia...	Number...	OrderA...	Coupon...	OrderC...	DaySinc...	Cashba...	Custom...
Churn	1	0.250	-0.336	0.085	0.074	0.029	0.019	0.108	0.105	0.044	-0.011	-0.001	-0.028	-0.156	-0.154	-0.019
Complain	0.250	1	-0.021	0.003	0.028	-0.040	0.007	0.003	-0.031	-0.026	-0.005	-0.008	-0.019	-0.043	0.001	-0.010
Tenure	-0.336	-0.021	1	-0.059	-0.018	-0.046	-0.021	-0.023	-0.014	0.233	0.010	0.098	0.176	0.177	0.462	0.030
CityTier	0.085	0.003	-0.059	1	0.010	-0.025	-0.010	0.028	-0.012	-0.029	-0.032	0.023	0.033	0.019	0.056	0.003
Warehou...	0.074	0.028	-0.018	0.010	1	-0.001	0.057	0.018	0.008	-0.011	0.036	-0.002	0.001	0.016	-0.011	0.056
Gender	0.029	-0.040	-0.046	-0.025	-0.001	1	-0.018	-0.022	-0.035	-0.031	0.000	-0.035	-0.031	-0.020	-0.025	0.004
HourSpe...	0.019	0.007	-0.021	-0.010	0.057	-0.018	1	0.305	0.031	0.140	0.102	0.187	0.103	0.072	0.114	0.580
Number...	0.108	0.003	-0.023	0.028	0.018	-0.022	0.305	1	-0.017	0.085	0.069	0.162	0.100	0.021	0.137	0.411
Satisfact...	0.105	-0.031	-0.014	-0.012	0.008	-0.035	0.031	-0.017	1	0.054	-0.027	0.017	0.019	0.031	0.003	-0.033
Number...	0.044	-0.026	0.233	-0.029	-0.011	-0.031	0.140	0.085	0.054	1	0.016	0.037	-0.007	-0.062	0.187	0.161
OrderAm...	-0.011	-0.005	0.010	-0.032	0.036	0.000	0.102	0.069	-0.027	0.016	1	0.034	0.024	0.010	0.025	0.116
Coupon...	-0.001	-0.008	0.098	0.023	-0.002	-0.035	0.187	0.152	0.017	0.037	0.034	1	0.649	0.314	0.219	0.233
OrderCo...	-0.028	-0.019	0.176	0.033	0.001	-0.031	0.103	0.100	0.019	-0.007	0.024	0.649	1	0.451	0.345	0.136
DaySinc...	-0.156	-0.043	0.177	0.019	0.016	-0.020	0.072	0.021	0.031	-0.062	0.010	0.314	0.451	1	0.345	0.113
Cashbac...	-0.154	0.001	0.462	0.056	-0.011	-0.025	0.114	0.137	0.003	0.187	0.025	0.219	0.345	0.345	1	0.217
Custom...	-0.019	-0.010	0.030	0.003	0.056	0.004	0.580	0.411	-0.033	0.161	0.116	0.233	0.136	0.113	0.217	1

Fig. 1 Korrelationsmatrix

4 Data Preparation

Beim Data Preparation werden die Daten für die spätere Modellierung aufbereitet. Die Aufbereitung der Daten findet in der Regel in zwei Phasen statt. In der ersten Phase werden die Daten allgemein angepasst. Es werden die Schritte durchgeführt, die für jedes Modell notwendig sind. Die zweite Phase ist sehr stark mit der Modellierung verbunden. Zum Beispiel ist die Normalisierung der Daten nur bei machen Modellen notwendig.

Die Daten vom Projekt benötigen keine größere allgemeine Aufbereitung, es sind zum Beispiel keine Datumswerte vorhanden, die umgewandelt werden müssen. Dennoch wurden kleine Anpassungen der Daten durchgeführt. Für ein besseres Verständnis werden die Attribute Churn, Complain von Numerical in Binominale umgewandelt. Damit es später zu keinem Missverständnis kommt was 1 bzw. 0 bedeutet. Die Werte True und False sind hingegen eindeutig. Auch werden Fehlwerte behandelt, da die meisten Modelle nicht mit Fehlwerten umgehen können. Jedoch wurde auch der Datensatz mit fehlenden Werten für den Fall behalten, dass ein Modell mit Fehlwerten zurecht kommt.

Für das behandeln der Fehlwerte gibt es im Prinzip zwei Möglichkeiten. Zunächst könnte jeder Datenpunkt, der Fehlwerte hat, entfernt werden. Die zweite Möglichkeit ist, die Fehlwerte durch sinnvolle Werte z.B. den Mittelwert ersetzt werden. Durch das Entfernen aller Datenpunkten mit Fehlwerten, würden fast ein Drittel der Daten verloren gegangen und es bestünde die Gefahr, dass sich dadurch die Verteilung der Daten ändert, wenn die Fehlwerte nicht zufällig verteilt sind. Daher haben wir die fehlenden Daten durch andere Werte ersetzt.

In dem Datensatz sind Fehlwerte bei: Tenure, WarehouseToHome, HourSpendOn-App, OrderAmountHikeFromLastYear, CouponUsed, OrderCount und DaySince-LastOrder vorhanden. Für das Ersetzen der fehlenden Werte gibt verschiedene Möglichkeiten. Zum Beispiel können die Werte auf null gesetzt werden oder auf den Maximalwert. Wir habe uns für alle Attributen außer CouponUsed jedoch für den Mittelwert entschieden, da dadurch die Datenverteilung sich am wenigsten verändert. Bei CouponUsed sind die meisten Werte bei 1. Der Mittelwert ist allerdings bei 1.7, da es einige große werte gibt. Daher haben wir die Fehlwerte durch den Median ersetzt, dieser liegt bei 1. Dadurch verändert sich der Mittelwert, allerdings wird die Datenverteilung weniger verändert.

5 Modellierung

5.1 Supervised Learning

5.1.1 Naïve Bayes

Funktionsweise

Naïve Bayes (NB) ist ein statistisches Klassifizierungsverfahren, mit starker Annahme von Unabhängigkeit zwischen den einzelnen Attributen. NB basiert auf dem Satz von Bayes, welcher besagt dass,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (5.1)$$

Hier sind A und B Wahrscheinlichkeitsereignisse und $P(X|Y)$ ist die bedingte Wahrscheinlichkeit.

Die Klassifizierung eines Samples $X = \{x_0, x_1, \dots, x_n\}$ zu einer Klasse $C = \{c_0, c_1, \dots, c_m\}$ kann als ein Wahrscheinlichkeitsereignis gesehen werden in dem die Klasse c_i mit der höchsten Wahrscheinlichkeit die richtige Klasse bezüglich X ist:

$$\arg \max_{i \in 0,1,\dots,m} P(c_i|x_0, x_1, \dots, x_n). \quad (5.2)$$

Mit Gleichung 5.1 kann Gleichung 5.2 umgeformt werden:

$$\arg \max_{i \in 0,1,\dots,m} \frac{P(x_0, x_1, \dots, x_n|c_i)P(c_i)}{P(x_0, x_1, \dots, x_n)}. \quad (5.3)$$

Der naive Teil von NB ist, dass angenommen wird alle Attribute x_i wären konditional Unabhängig, was bedeutet, dass:

$$P(x_0, x_1, \dots, x_n|c_i) = P(x_0|c_i), P(x_1|c_i), \dots, P(x_n|c_i). \quad (5.4)$$

Wenn man nun Gleichung 5.1 und Gleichung 5.4 kombiniert und den Zähler ignoriert (denn er ist konstant was für die Klassifizierung keinen Unterschied macht) erhält man folgendes Maximierungsproblem:

$$\arg \max_{i \in 0,1,\dots,m} P(c_i) \prod_{j=0}^n P(x_j|c_i), \quad (5.5)$$

was der Formel für die NB Klassifizierung entspricht.

Training

- NB kann sowohl mit nominalen- und numerischen Attributen sowie mit Fehlenden Werten (die einfach ignoriert werden) umgehen und es muss kein zusätzliches Datenpreprocessing durchgeführt werden.
- Da NB für die Klassifizierung verwendet werden kann, benutzen wir den Algorithmus als Klassifizierer für die Zielattribute *Churn* und *Complain*.
- Für alle Tests wird wie vorgegeben eine 10-Fache automatische Kreuzvalidierung durchgeführt.
- Als Metriken für die Klassifizierung wird Accuracy, AUC, Precision, Recall und Kappa verwendet.

Ablationsstudie

Da bei NB durch die Dichtefunktion (Wahrscheinlichkeitsverteilung) der Einfluss jedes Attributs bestimmt werden kann wird zusätzlich zur Klassifikation untersucht, welche Attribute den größten Einfluss auf die Entscheidungsfindung von NB haben und wie sehr sich die Performance verändert, wenn unwichtigen Attribute nicht für die Erstellung des Models mit einbezogen werden.

Churn

Dichtefunktionen der einzelnen Attribute:

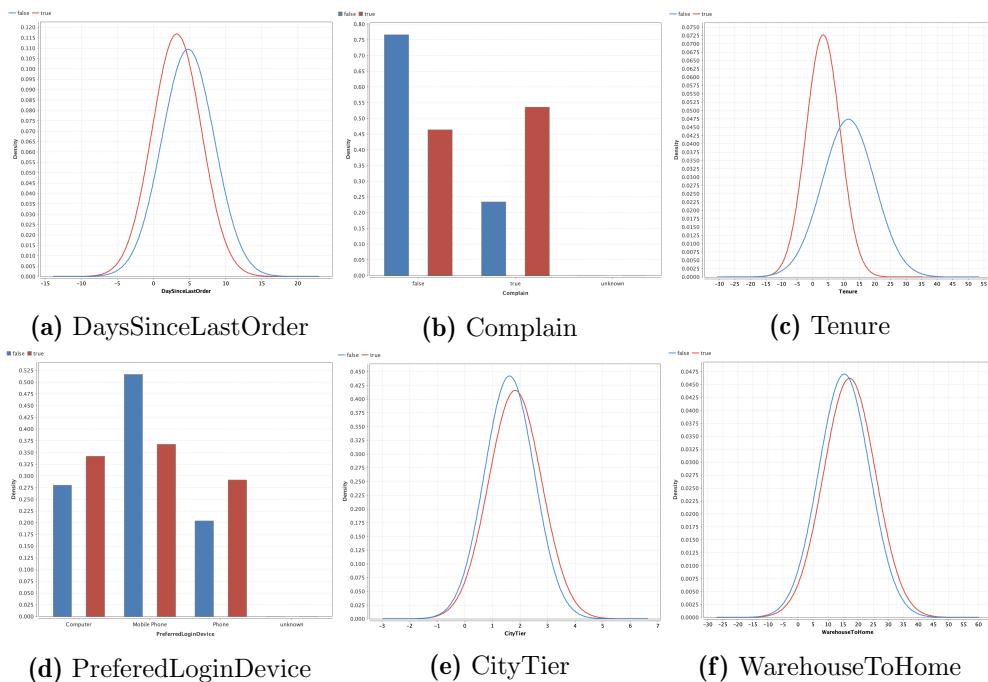


Fig. 2 Dichtefunktionen Churn 1/2

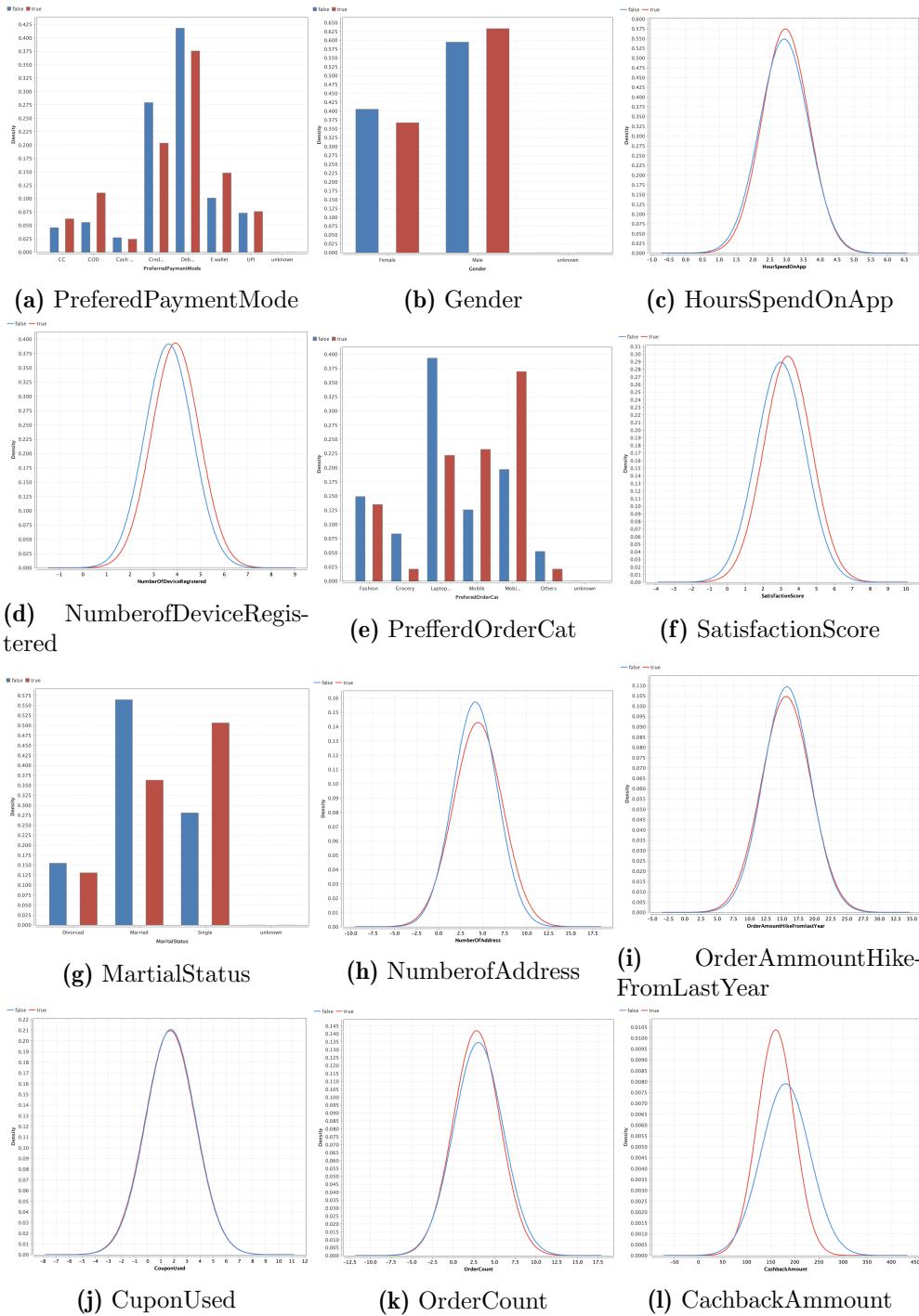


Fig. 3 Dichtefunktionen Churn 2/2

Aus den Dichtefunktionen lässt sich erkennen, dass viele Attribute isoliert gesehen nur einen sehr kleinen Einfluss auf die Qualität der Klassifizierung von NB haben werden. So sind *CityTier*, *WarehouseToHome*, *Gender*, *HourSpendOnApp*, *NumberOfDeviceRegistered*, *SatisfactionScore* und *NumberOfAddress* durch ihre große Überlappungen für die NB Klassifizierung zu vernachlässigen.

Die Attribute, sortiert nach geschätzter Wichtigkeit:

1. Complain
2. Tenure
3. PrefferdOrderCat
4. PreferredPaymentMethod
5. PrefferdLoginDevice
6. SatisfactionScore
7. MartialStatus
8. CashbackAmmount
9. DaysSinceLastOrder
10. Gender
11. NumberOfDeviceRegistered
12. CityTier
13. WarehouseToHome
14. NumberOfAddress
15. OrderCount
16. HourSpendOnApp
17. OrderAmmountHikeFromLastYear
18. CuponUsed

Jedoch bilden selbst *Complain* und *Tenure* keine weit getrennten Verteilungen, was keine gute Klassifikation mit NB erwarten lässt.

In der Ergebnistabelle sind die oben genannten Metriken gegen die Anzahl der verwendeten Attributen (# Attribute) gezeigt.

# Attribute	Accuracy	Kappa	AUC	Precision	Recall
18	83.91%	0.465	0.827	51.94%	61.50%
14	84.33%	0.473	0.832	53.04%	61.29%
10	82.61%	0.428	0.811	48.66	59.18%
6	85.35%	0.425	0.818	58.57%	45.15%
2	85.65%	0.393	0.822	62.18%	37.88%

Wie erwartet zeigt das Ergebnis, dass viele der Attribute für NB nicht nützlich sind, folglich sinkt die Performance nur sehr langsam und selbst mit nur zwei Attributen (*Complain* und *Tenure*) kann vergleichbare Performance erreicht werden. Das zeigt, dass NB für die Klassifikation von *Churn* nicht geeignet ist, da vermutlich eine große Abhängigkeit zwischen den Attributen besteht.

Ergebnisse: Complain

Dichtefunktionen der einzelnen Attribute:

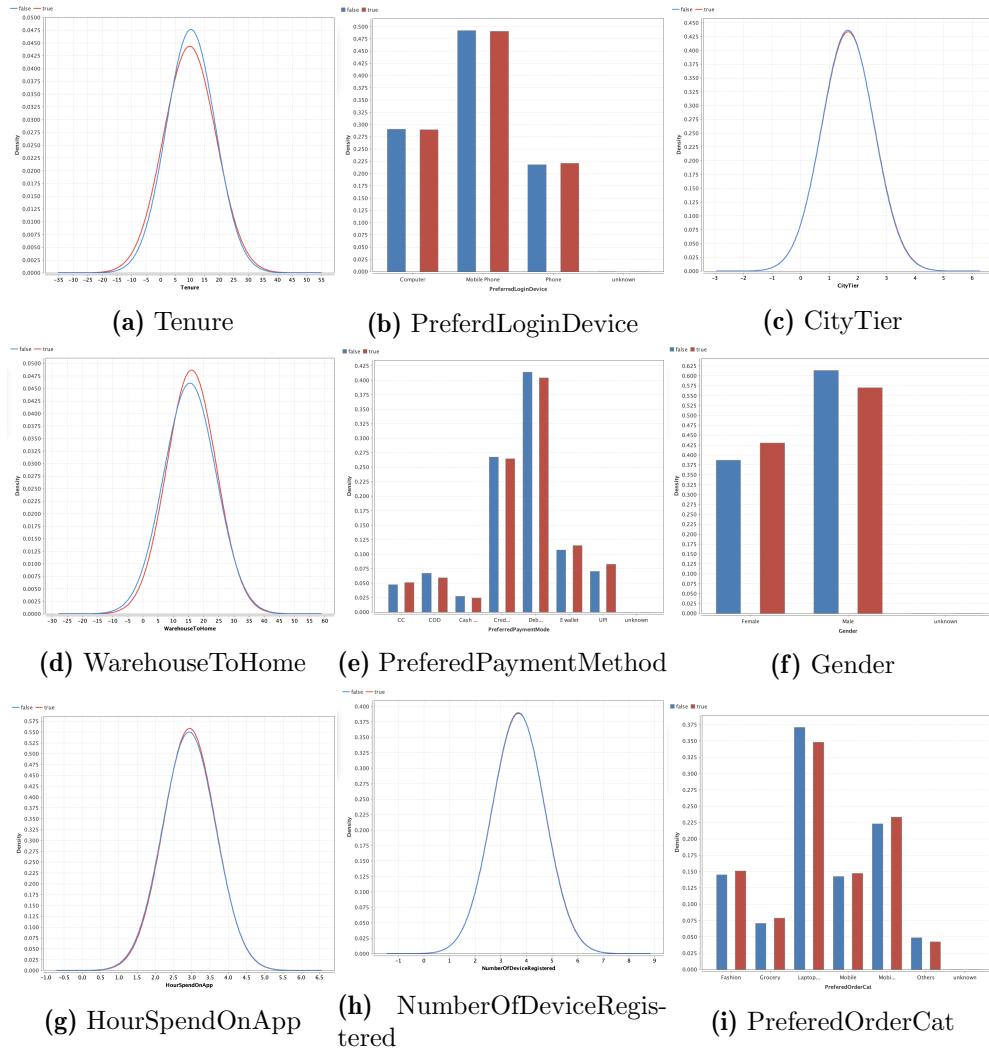
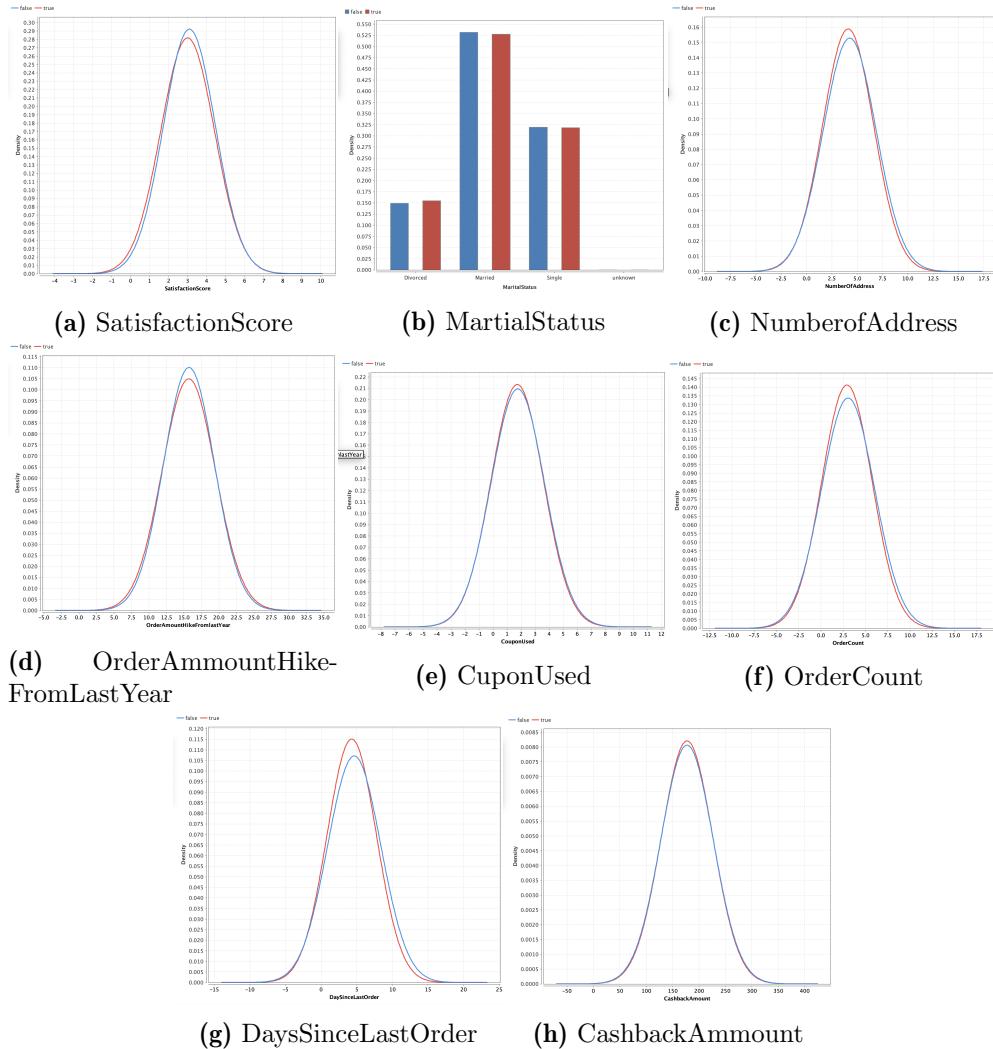


Fig. 4 Dichtefunktionen Complain 1/2

**Fig. 5** Dichtefunktionen Complain 2/2

So wie bei der Klassifikation von *Churn*, lässt sich erkennen, dass Die Attribute isoliert nur einen kleinen Einfluss haben. Die Aussagekraft der einzelnen Attribute scheint bei *Complain* aber noch kleiner zu sein. Die Attribute, sortiert nach geschätzter Wichtigkeit:

1. Tenure
2. OrderCount
3. DaysSinceLastOrder
4. NumberOfAddress
5. OrderAmmountHikeFromLastYear
6. SatisfactionScore

7. PreferdPaymentMethod
8. WarehouseTomHome
9. CuponUsed
10. CashbackAmmount
11. HourSpendOnApp
12. PreferedOrderCat
13. Gender
14. CityTier
15. NumberOfDeviceRegistered
16. MartialStatus
17. PreferedLoginDevice

So wie bei *Churn*, sind in der Ergebnistabelle sind die oben genannten Metriken gegen die Anzahl der verwendeten Attributen (# Attribute) gezeigt.

# Attribute	Accuracy	Kappa	AUC	Precision	Recall
17	71.47%	0.00	0.538	33.33%	0.12%
14	71.49%	0.01	0.545	40.00%	0.12%
10	71.51%	0.01	0.546	50.00%	0.12%
6	71.51%	0.01	0.545	50.00%	0.12%
2	71.51%	0.01	0.525	50.00%	0.12%

Das Ergebnis zeigt, dass BN nicht fähig dazu ist, *Complain* zu klassifizieren. Was zu erwarten war denn die Wahrscheinlichkeitsverteilungen zeigen bereits, dass die Attribute für sich gestellt nicht genug Aussagekraft haben um eine Klassifizierung zuzulassen.

5.1.2 Support Vector Machine

Die Support Vector Maschinen sind keine Maschinen im eigentlichen Sinne. Es handelt sich vielmehr um ein mathematisches Modell, welches für die binäre Klassifizierung wie auch für die Regressionsanalyse eingesetzt werden kann. Support Vector Maschinen wurden schon in den neunziger Jahren, hauptsächlich von Vapnik erfunden und haben laut [DL10] zu einem immensen Interesse an Techniken des maschinellen Lernens geführt. Laut dem Author basieren die meisten Algorithmen zur Klassifizierung auf traditionellen statistischen Methoden mit sehr guten Resultaten, falls die Eingabegröße, gegen unendlich konvergiert. Durch die begrenzte Anzahl an Rechenzeit und Speicherplatz, ist in der Praxis allerdings nur eine endliche Anzahl von Eingabegrößen möglich. Hier bietet die Support Vector Machine eine sehr effiziente Alternative und kann mithilfe des Kernel Tricks in hochdimensionalen Räumen angewandt werden.

Funktionsweise

Im zweidimensionalen Raum kann man sich das Modell als eine Methode, die eine Art Trennlinie zwischen die Daten, welche klassifiziert werden müssen, legt, vorstellen. Hierbei ist wichtig das die Gerade nicht willkürlich zwischen die Daten gelegt wird, sondern unter Berücksichtigung bestimmter Bedingungen. Eine grundlegende Bedingung ist, das der Abstand um die Gerade zu den nächstliegenden Datenpunkten, so groß wie möglich ist. Dies kann auf dem rechten Graphen von 11 betrachtet werden.

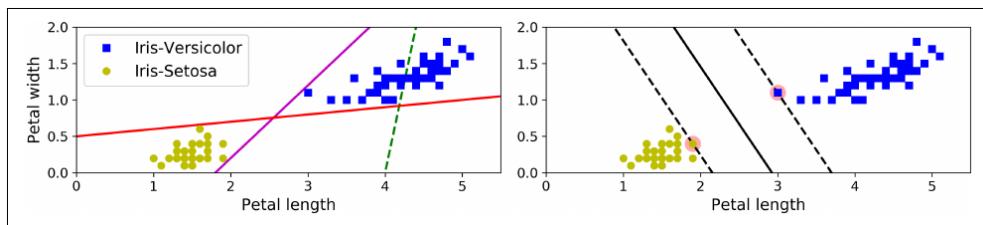


Fig. 6 Das linke Bild zeigt eher ungeeignete Trennlinien und das rechte eine Trennlinie mit zwei Stützvektoren, berechnet von einer Support Vektor Machine. Die Grafik wurde aus [Ger17] entnommen.

Falls die Daten nun nicht linear separierbar sind, gibt es verschiedene Methoden mit diesem Problem umzugehen. Eine Methode nennt sich **Large Margin Klassifizierung** und erlaubt das Eindringen von Datenpunkten in den Bereich um die Trennlinie. Das mathematische Optimierungsproblem kann wie folgt definiert werden

$$\min_{\theta} C \sum_{i=1}^m [y_i \text{cost}_1(\theta^T x_i) + (1 - y_i) \text{cost}_0(\theta^T x_i)] + \frac{1}{2} \sum_{j=1}^n \theta_j^2 \quad (5.6)$$

Mit dem Parameter C^1 kann nun eingestellt werden, wie breit der Bereich um Trennlinie sein soll. θ ist hier der Gewichtsvektor, welcher optimiert wird um die beste Trennlinie zu finden.

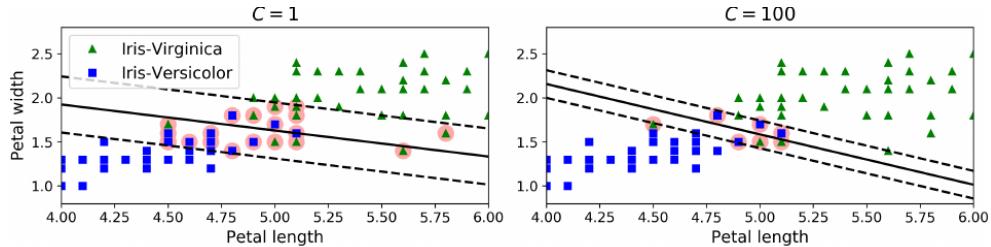


Fig. 7 Das linke Bild zeigt ein Modell welches einen großen Bereich und somit viele Verstöße zulässt, während im rechten durch einen schmäleren Bereich, weniger Verstöße zugelassen werden. Der Graph stammt aus dem Buch [Ger17]

Eine weitere Methode um mit nicht linear separierbaren Daten umzugehen, ist die Benutzung von Kernels. Im Prinzip geht es hierbei darum Daten wieder linear separierbar zu machen, indem man sie in einen höher dimensionalen Raum zu überführt. Das Prinzip ist in 8 mit einer Transformation von $R \rightarrow R^2$ beschrieben.

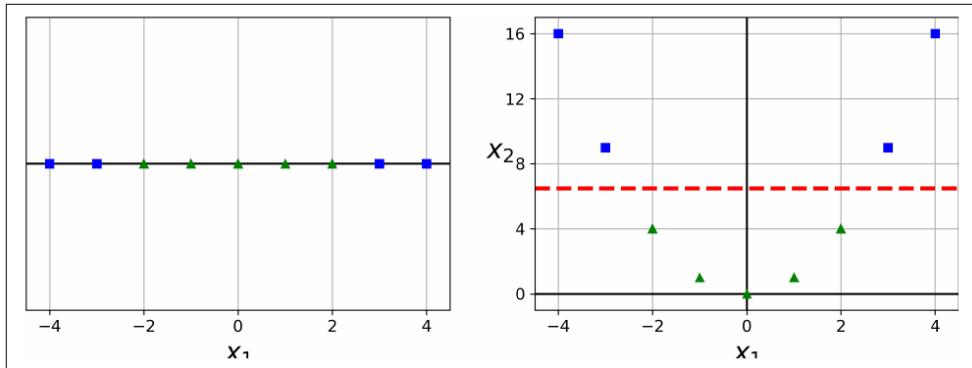


Fig. 8 Das linke Bild zeigt Daten im eindimensionalen Raum, welche nicht linear separierbar sind. Nach der Transformation (rechtes Bild) können die Daten durch eine Linie getrennt werden. Die Graphen stammen aus dem Buch [Ger17]

Da eine Transformation in einen hochdimensionalen Raum, mit enormem Rechenaufwand verbunden ist, kann man sich bei Benutzung der verschiedenen Kernels eines mathematischen Tricks bedienen, welcher es erlaubt, sich der Vorteile, hochdimensionaler Funktionen zu bedienen, ohne diese jemals berechnen zu müssen. Eine

¹Der Parameter C wirkt ähnlich wie der Regularisierungsparameter, allerdings umgekehrt. Falls das Modell auf die Trainingsdaten überangepasst ist (Overfitting), sollte der Parameter heruntergesetzt werden. Falls das Modell unterangepasst ist (Underfitting) sollte der Parameter erhöht werden.

detaillierte Beschreibung von Martin Hoffmann diesbezüglich, kann unter [Hof06] nachgelesen werden. Wie in dem Dokument von Herrn Hoffman beschrieben, geht es bei dem Kerneltrick im Prinzip um die Vereinfachung einer Multiplikation der Form

$$(x_1^2, \sqrt{2}x_2x_2, x_2^2)^T(z_1^2, \sqrt{2}z_1z_2, z_2^2) \quad (5.7)$$

auf lediglich

$$(x^T z)^2 \quad (5.8)$$

wobei x und z beides Vektoren sind. Um den effizientesten Kernel in Bezug auf das zu lösende Problem zu finden, kann eine Grid Suche durchgeführt werden.

Training

Da Support Vector Maschinen für die Klassifizierung wie auch für die Regression verwendet werden können, benutzen wir den Algorithmus hier als Klassifizierer für die Zielattribute *Churn* und *Complain*, und als Regressor für das Zielattribut *Cashback Ammount*. Da nach einigen Testdurchläufen sofort ersichtlich ist, dass die Performance durch eine höhere Anzahl von Eingabeattributen steigt, werden für die Voraussage der Zielvariablen, alle numerischen² Attribute als Eingabe verwendet, soweit dies unter Beachtung der kausalen Zusammenhänge erlaubt ist. Es ist zudem noch anzumerken, dass ein von fehlenden Daten bereinigter und normalisierter Datensatz verwendet wird.

Das Training für den Klassifizierer und den Regressor wird nach folgenden Schritten durchgeführt

1. Grid Optimierung für die Suche nach der besten Kombination von Parametern.
2. Evolutionary Optimierung
3. Vergleich von Modellen mit und ohne Smote Upsampling

Als Metriken für die Klassifizierung wird hier Accuracy, AUC, Precision, Recall und Kappa verwendet. Nachstehend eine kurze Erläuterung dieser Gütemaße:

- Accuracy ist der Prozentsatz der korrekt klassifizierten Beispiele - wahre Proben / Gesamtproben
- AUC bezieht sich auf die Fläche unter der Linie einer Receiver Operating Characteristic-Kurve (ROC-AUC). Diese Metrik ist gleich der Wahrscheinlichkeit, dass ein Klassifikator eine zufällige positive Probe höher einstuft als eine zufällige negative Probe.

²Support Vector Maschinen können nur numerische Attribute verarbeiten.

- Precision ist der Prozentsatz der vorhergesagten Positiven, die korrekt klassifiziert wurden.
- Recall ist der Prozentsatz der tatsächlichen Positiven, die richtig klassifiziert wurden - wahre Positive / wahre Positive + falsch Negative
- Kappa ist die Differenz zwischen der Accuracy des Null-Modells und der wirklichen Accuracy

Als Metriken für die Regression wird RMSE und MAE verwendet. Diese sind wie folgt definiert:

- RMSE (root mean squared error) MSE kann als absolute Güte für das Modell angesehen werden und wird wie folgt berechnet $MSE = \frac{1}{N} \sum_{i=1}^n (y_i - \hat{y}_i)^2$. RMSE ist dann lediglich die Quadratwurzel von MSE.
- MAE (mean absolute error) Hier wird anstatt der Summe der Quadrate, die Summe der absoluten Werte berechnet. Die Berechnung erfolgt mit $MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$. Intuitiv gibt MSE höhere Bestrafungen, um so höher der Schätzungsfehler ist.

Für alle Tests wird wie vorgegeben eine 10-Fache automatische Kreuzvalidierung durchgeführt.

Ergebnisse - Churn

Optimierungsschritt	Accuracy	Kappa	AUC	Precision	Recall	Optimierungsauswahl
Grid: Kernel	90.66%	0.58	0.95	93.10%	48.11%	Anova Kernel
Grid: Degree	97.34%	0.903	0.986	94.34%	89.56%	Degree: 8
Grid: C	97.35%	0.903	0.982	94.58%	89.45%	C: 9.1
Grid: Cache	97.35%	0.903	0.982	94.58%	89.45%	Cache: 100
Grid: Gamma, Smote	97.28%	0.901	0.985	97.99%	98.76%	Gamma: 30
Evolutionary: Degree, Gamma	97.32%	0.901	0.99	95.31%	88.51%	Degree: 5.61, Gamma: 6.85

Ergebnisse - Complain

Optimierungsschritt	Accuracy	Kappa	AUC	Precision	Recall	Optimierungsauswahl
Grid: Kernel	90.21%	0.76	0.93	82.93%	82.92%	Anova Kernel
Grid: Degree	92.61%	0.81	0.96	89.77%	83.66%	Degree: 8
Grid: Gamma	92.38%	0.79	0.98	98.84%	74.13%	Gamma: 30.0
Grid: C	93.34%	0.83	0.95	91.37%	84.66%	C: 74.5
Grid: Gamma, Smote	93.75%	0.84	0.97	96.77%	94.44%	Gamma: 40
Evolutionary: Degree, Gamma, C	94.30%	0.85	0.97	97.15%	82.42%	Degree: 5.66, Gamma: 86.34, C: 61.94

Ergebnisse - Cashback amount

Optimierungsschritt	RMSE	MAE	Optimierungsauswahl
Grid: Kernel	21.22	12.11	Anova Kernel
Grid: Degree	19.36	10.94	Degree: 4.8
Grid: C	20.22	11.73	C: 39.4
Grid: Gamma	18.92	10.54	Gamma: 40
Evolutionary: Degree, Gamma	19.82	11.34	Degree: 4.83, Gamma: 22.05
Evolutionary: Gamma, Degree, C	18.85	10.89	Gamma: 24.04, Degree: 4.03, C: 18.04

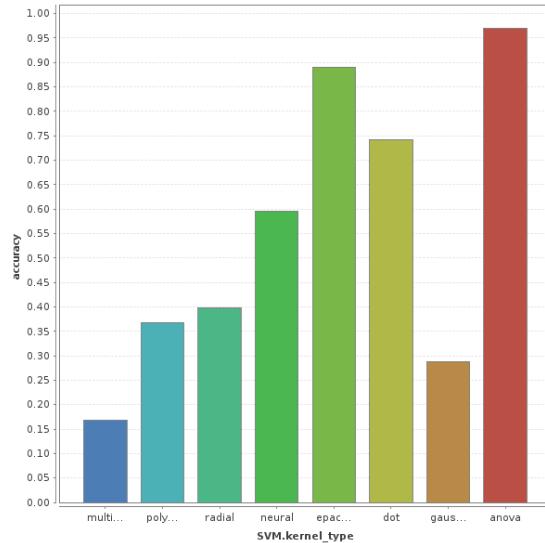


Fig. 9 Graph welcher die Optimierungsauswahl des Kernels mit der höchsten Accuracy bei der Klassifizierung des Zielattributes Churn, darstellt. Auch bei der Variable Complain und CashbackAmount, erzielt der Anova Kernel die besten Ergebnisse.

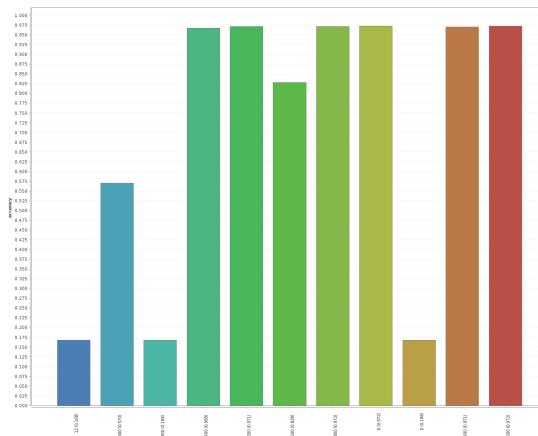


Fig. 10 Graph zeigt die Optimierungsauswahl des Grades eines Kernels. Hier kann man sehen das der Bereich des Grades von 2 bis 8, sehr gute Ergebnisse liefert.

Bei den Ergebnissen der Klassifizierung wie auch bei der Regression der **Support Vector Maschinen** wird bei einer Kernel Suche über die Grid Optimierung, mit

Abstand jeweils der Anova Kernel gewählt. Siehe hierzu 9. Das Kappa ist bei der Klassifizierung in diesem Schritt noch niedrig bei 0.58. Bei Erhöhung des Grades jedoch, springt dieser Wert auf bis zu 0.903. Eine Evolutionary Optimierung oder Grid Optimierung einiger Parameter wie Gamma, Cache oder C, hat hier nur zu einer leichten Verbesserung des Kappa geführt. Der Upsampling Operator Smote, welcher unausgeglichene Zielvariablen *upsamplen* kann, führt hier lediglich zu einer wesentlichen Verbesserung des Recall. Dies macht Sinn da Smote Werte gleichmäßig auf 50:50 verteilt. Bei der Voraussage des Zielattributes *Complain*, hat die Evolutionary Optimierung noch zu Verbesserungen des Kappa, wie auch der Accuracy geführt. Bei der Regression, brachte neben der Auswahl des Anova Kernels, eine Evolutionary Optimierung der Parameter Gamma, Grad und C, die besten Werte mit einem RMSE von 18.85.

5.1.3 Neuronale Netze

Laut [Wik21] geht es bei neuronalen Netzen weniger um das Nachbilden biologischer neuronaler Netze und Neuronen, sondern mehr um eine Modellbildung von Informationsverarbeitung. Das erste neuronale Netz wurde schon in den fünfziger Jahren entwickelt und ist unter dem Namen Perzeptron bekannt. Durch die enormen Verbesserungen in der Hardwareentwicklung, erleben laut dem oben genannten Artikel, neuronale Netzwerke seit 2009 eine Wiedergeburt mit einer enormen Verbesserung für bestimmte Anwendungen. Auch wenn neuronale Netze oft für hochdimensionale Problemstellungen wie für die Bilderkennung oder Textverarbeitung verwendet werden, könne diese auch für kleinere Datensätze benutzt werden.

Funktionsweise

Die prinzipielle Funktionsweise kann durch ein Modell mit einem gewichteten Eingabevektor an eine Übertragungs und Aktivierungsfunktion beschrieben werden, welches nach Vergleich mit den Zielwerten, die Gewichtsvektoren nach und nach anpasst, um den Fehler zu minimieren und bessere Voraussagen zu erhalten. Folgende Grafik veranschaulicht dieses Prinzip etwas genauer.

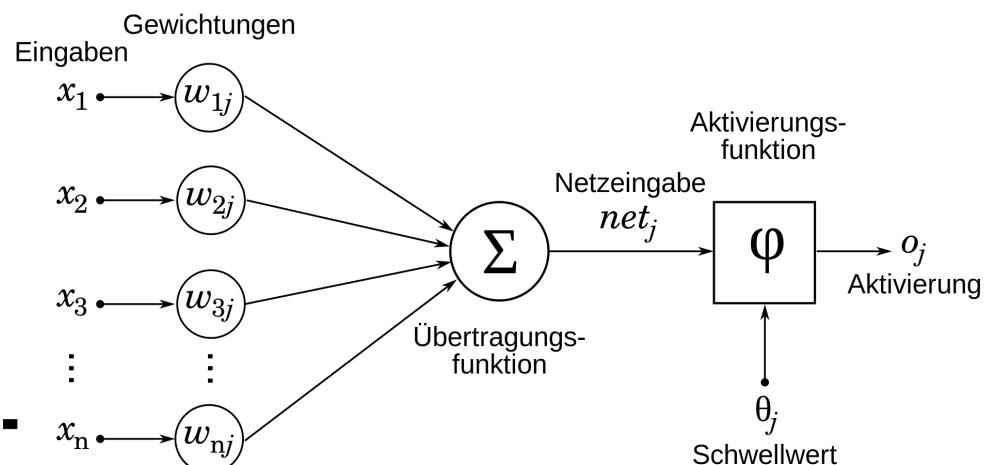


Fig. 11 Die Grafik zeigt ein Schema für ein künstliches Neuron und ist ursprünglich abgebildet auf [Wik21]

Training

Neuronale Netze können wie die Support Vector Maschinen als Klassifizierer oder Regressor verwendet werden. Auch hier wird für dieses Projekt als Zielattribute *Churn* und *Complain*, und für die Regression *Cashback amount* verwendet.

Das Training wird hier nach folgenden Schritten durchgeführt

1. Grid Optimierung für die Suche nach der besten Kombination von Parametern.

2. Evolutionary Optimierung

3. Vergleich von Modellen mit und ohne Smote Upsampling

Laut [GBC16] ist die Lernrate der wichtigste Parameter und sollte auf jeden Fall durch eine Grid oder Evolutionary Suche optimiert werden. Auch die Anzahl der verwendeten Layer oder der Epochen, kann hier über eine Parametersuche optimiert werden. Da ein Training oft sehr lange dauern kann, empfiehlt es sich verschiedene Parameter für eine kleine Anzahl an Epochen zu testen, und die besten Kombinationen im Anschluss für ein Modell mit vielen Epochen zu verwenden.

Als Metriken werden hier die gleichen wie in 5.1.2 ausführlich beschrieben, verwendet.

Für alle Tests wird wie vorgegeben eine 10-Fache automatische Kreuzvalidierung durchgeführt. Als Operator wird *Deep Learning* in Rapid Miner eingesetzt.

Ergebnisse - Churn

Optimierungsschritt	Accuracy	Kappa	AUC	Precision	Recall	Epochs	Activation	Optimierungsauswahl
Grid: Learning Rate	89.11%	0.61	0.91	67.35%	68.78%	100	Relu	Learning rate: 0.2
Grid: Learning Rate, Activation	91.58%	0.69	0.94	77.19%	71.74%	100	Tanh	Learning rate: 0.2, Activation: Tanh
Grid: Learning Rate	91.47%	0.68	0.93	76.41%	71.94%	100	Tanh	Learning rate: 0.62
Grid: Learning Rate, Momentum	92.10%	0.71	0.94	78.06%	73.94%	100	Tanh	Learning rate: 0.5, Momentum start: 1.0
CV	95.03%	0.82	0.97	86.58%	83.56%	3000	Relu	Adaptive rate
CV	95.68%	0.84	0.97	88.08%	86.28%	3000	Tanh	Adaptive rate
CV mit Smote Upsampling	90.87%	0.70	0.95	97.06%	91.80%	3000	Tanh	Adaptive rate

Ergebnisse - Complain

Optimierungsschritt	Accuracy	Kappa	AUC	Precision	Recall	Epochs	Activation	Optimierungsauswahl
Grid: Activation	82.63%	0.57	0.87	70.31%	68.58%	100	Tanh	Activation: Tanh
Grid: Learning Rate	83.29%	0.58	0.88	71.51%	69.07%	100	Tanh	Learning Rate: 0.2
CV	91.39%	0.78	0.95	87.09%	81.99%	3000	Tanh	Adaptive rate

Ergebnisse - Cashback amount

Optimierungsschritt	RMSE	MAE	Epochs	Activation	Optimierungsauswahl
Grid: Activation function	31.61	23.31	100	Tanh	Activation function: Tanh
CV	22.43	15.42	1000	Tanh	
CV	20.24	13.50	2000	Tanh	
CV	21.09	13.90	3000	Tanh	
Grid: Learning rate	22.21	15.27	3000	Relu	Learning rate: 0.9
Evolutionary: Learning rate	31.37	23.14	100	Relu	Learning rate: 0.07
Evolutionary: Learning rate, Epochs	22.86	16.01	968	Relu	Learning rate: 0.19, Epochs: 968

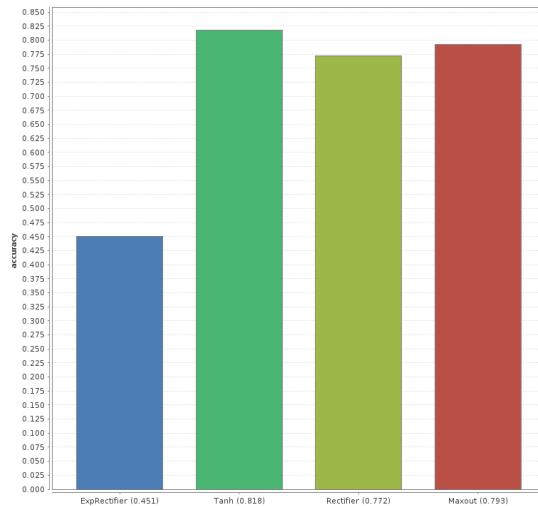


Fig. 12 Der Graph stellt eine Optimierungssuche nach der besten Aktivierungsfunktion dar. Hier schneidet die Funktion Tanh bei allen Zielvariablen am besten ab.

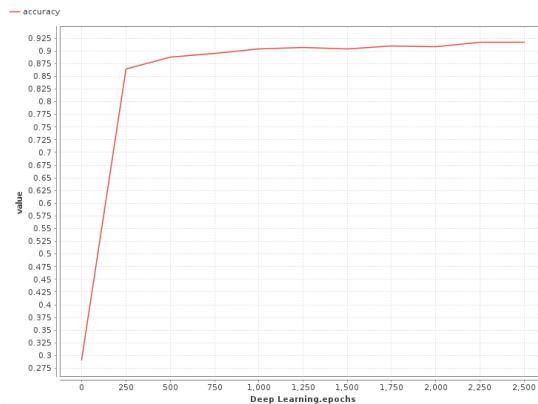


Fig. 13 Optimierungssuche für die Anzahl der Epochen. Es werden keine wesentliche Verbesserung ab einer Anzahl von ca. 2000 Epochen erzielt.

Bei den **Neuronalen Netzen** mit dem Operator Deep Learning liegen die besten Werte für Kappa bei der Klassifizierung für *Churn* bei 0.84 und bei *Complain* bei 0.78. Die Accuracy bei 95.68% und 91.39%. Ausschlaggebend für wesentliche Verbesserungen ist hier auf jeden Fall die Anzahl der Epochen. Siehe hierzu die Grafik 13. Vor Erhöhung der Epochen lohnt es sich die optimalen Parameter wie Lernrate, Momentum oder eine geeignete Aktivierungsfunktion zu finden. Prinzipiell ist laut [GBC16] die Lernrate der wichtigste Parameter bei Neuronalen Netzen. Bei Voraussage der Variable *Cashback amount*, konnte das beste Ergebnis (RMSE 21.09), lediglich durch die Erhöhung der zu trainierenden Epochen erreicht werden. Eine Evolutionary Optimierung der Parameter, hat hier zu keiner wesentlichen Verbesserung geführt.

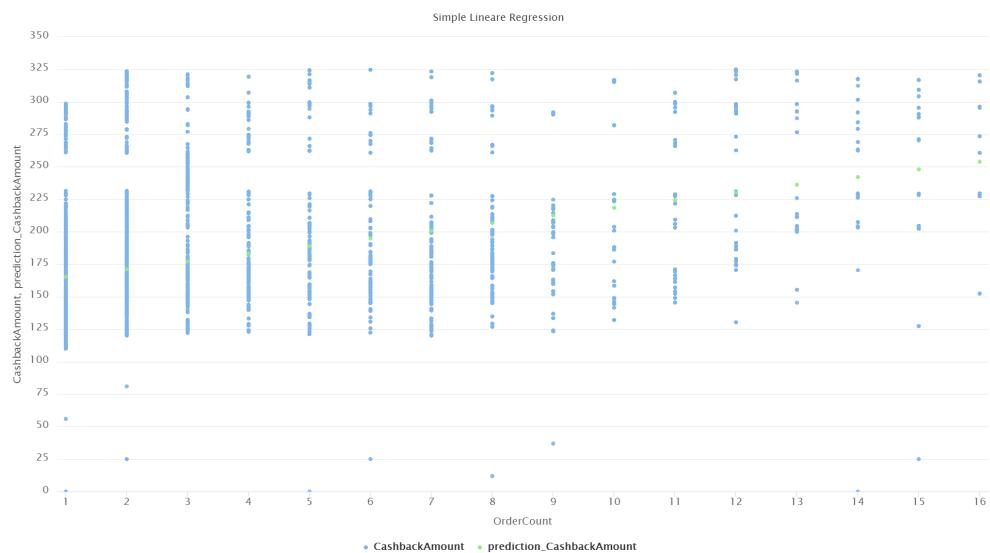
5.1.4 Lineare Regression

Die Regression ist die Schätzung einer numerische Zielvariablen. Bei der linearen Regression wird ein linearer Zusammenhang zwischen den Inputvariablen und der Zielvariablen angenommen. Um eine optimale Regressiongerade zu erreichen, muss die Summe der Residuen minimiert werden. Dafür wird die quadratische Lossfunktion verwendet, um größere Abweichungen zu bestrafen. Da bei der Linearen Regression keine Fehlwerte im Datensatz auftauchen sollten wird der Datensatz mit den bereinigten Fehlwerten verwendet.

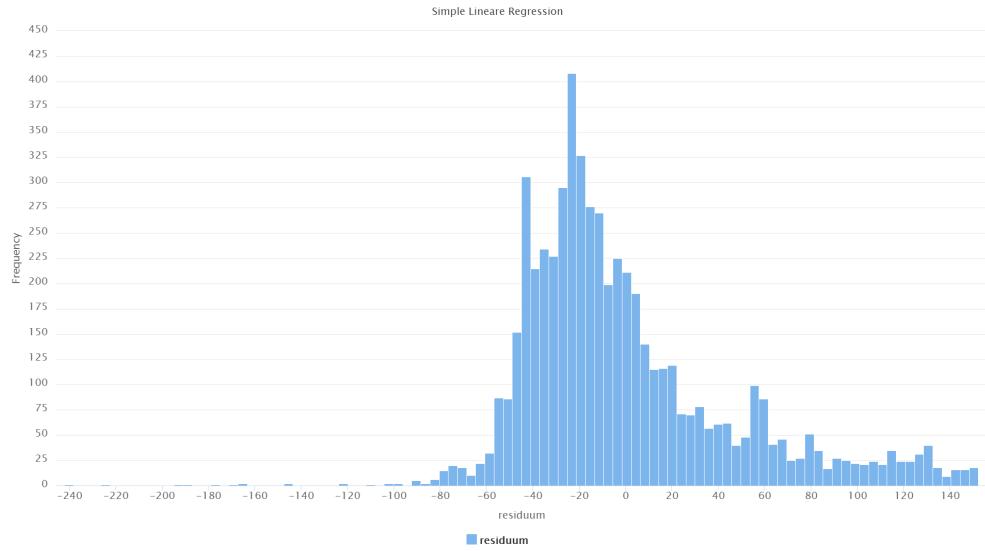
Simple Lineare Regression Zunächst wird eine simple lineare Regression mit der Zielvariablen CashbackAmount und der Inputvariable OrderCount umgesetzt. Der Koeffizient der Inputvariable OrderCount ist hierbei 5.904. Dies ist die Steigung der Geraden. Es ist natürlich sinnvoll, dass dies positiv ist, da das Attribut OrderCount eine positive Korrelation zur Zielvariablen besitzt. Der Bias, also der Schnitt der y-Achse, beträgt 159.495. Der p-Value ist 0, das bedeutet, dass das Ergebnis sehr signifikant ist. Dies wird in der folgenden Visualisierung deutlich. Die Performance Maßzahlen sind wie folgt:

- root mean squared error: 46.442
- absolute error: 34.913
- squared correlation: 0.112

Das Modell liegt somit im Schnitt 34.913 Punkte daneben. Der Erklärungsgehalt des Modells ist mit einem squared correlation von 0.112 sehr gering.



Die Visualisierung der Residuen lässt eine generelle Normalverteilung mit einer leichten Tendenz in den negativen Wertebereich erkennen.



Dieses Verfahren wurde auch mit den anderen Inputvariablen umgesetzt. Tenure erzielte dabei die besten Ergebnisse, allerdings wurden keine weiteren Erkenntnisse daraus gewonnen.

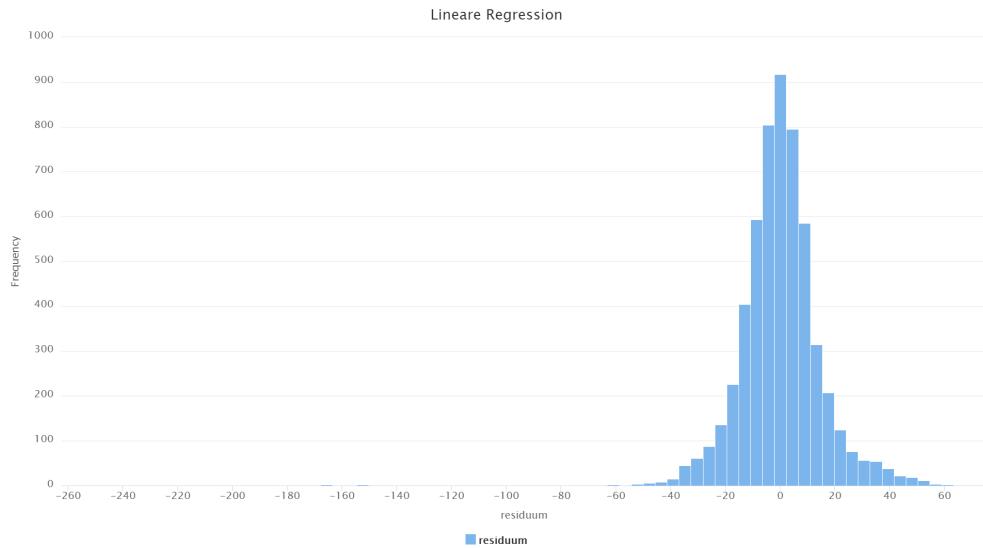
Lineare Regression Es werden alle Variablen als Inputattribute (Nominal Attribute zu numerische umgewandelt) außer dem Attribut CouponUsed verwendet, da dieses eine hohe Korrelation zum Inputattribut OrderCount besitzt. Alle numerischen Attribute werden mit der Z-transformation normalisiert. Das Modell mit einer minimalen Toleranz von 0.05 und einem ridge von $1.0 * 10^{-8}$ erreicht folgende Performance Maßzahlen:

- root mean squared error: 17.276
- absolute error: 10.688
- squared correlation: 0.877

Der root mean squared error und der absolute error sind im Vergleich zur Simple Linear Regression jeweils gesunken und die squared correlation gestiegen. Die squared correlation besagt, dass 87.7 Prozent der Schwankungen der Zielvariable um den Mittelwert erklärt werden können.

Einen signifikanten Einfluss hat das umgewandelte Attribut mit der präferierte Bestellkategorie. Anhand der Koeffizienten ist zu erkennen, dass die Bestellkategorien Fashion, Grocery und Others einen positiven Einfluss auf die Zielvariable ausüben und Laptop & Accessory, Mobile und Mobile Phone einen stark negativen. Dazu zeigen die präferierte Einlog Device, die Attribute HourSpendOnApp, NumberOfDeviceRegistered und NumberOfAddress, Churn, OrderAmountHike-FromLastYear und DaySinceLastOrder ebenfalls einen signifikanten Einfluss.

Die Visualisierung der Residuen zeigen eine deutlich bessere und annähernde Normalverteilung.

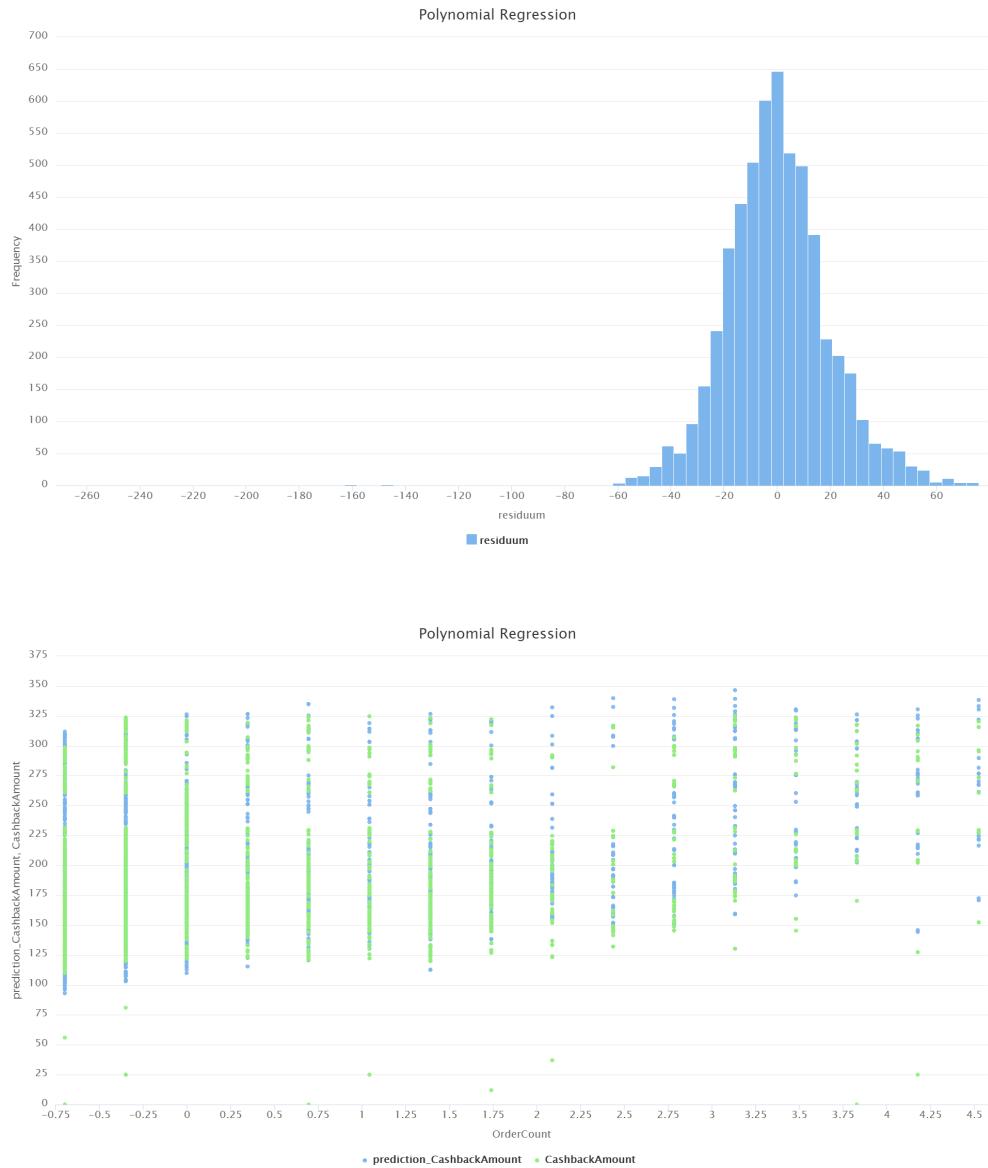


5.1.5 Polynomial Regression

Bei der polynomialen Regression können auch nichtlineare Zusammenhänge erkannt werden, indem Polynome n-ter Ordnung in der Regressionsgleichung vorkommen. Da bei der polynomialen Regression keine Fehlwerte im Datensatz auftauchen sollten wird der Datensatz mit den bereinigten Fehlwerten verwendet. Für das Preprocessing werden dieselben Schritte wie bei der linearen Regression angewandt. Mit der maximalen Anzahl an Iterationen von 20000, einem replication factor und einer maximalen Ordnung von 2 erzielt die polynomiale Regression folgende Performance Maßzahlen:

- root mean squared error: 37.030
- absolute error: 26.921
- squared correlation: 0.478

Die erzielten Ergebnisse sind somit schlechter als die der linearen Regression. Dies lässt sich mit Overfitting erklären, da sich das Modell zu sehr an die Trainingsdaten angepasst hat und dadurch eine schlechtere Generalisierung erzielt und somit schlechtere Ergebnisse auf den Testdatensatz erzielt.



5.1.6 Logistische Regression

Die logistische Regression ist im Gegensatz zur linearen und polynomialen Regression ein Verfahren für eine binäre Klassifikation. Dies bedeutet, dass die Zielvariable binomial ist. Bei diesem Verfahren wird zunächst eine linearen Regression durchgeführt, anhand dessen numerische Ausgabe im zweiten Schritt die Klasse bestimmt wird. Häufig wird hierfür die Sigmoid Funktion verwendet. Dabei wird die numerische Ausgabe der Sigmoid Funktion als Eingabe gegeben und diese mappt den numerischen Ausgabewert auf einen Wertebereich von 0 bis 1. Anschließend wird die Ausgaben der Sigmoid Funktion mit dem Schwellenwert von 0.5 verglichen. Ist der Ausgabewert größer 0.5, wird die Klasse eins vorhergesagt. Ist die Ausgabe kleiner dem Schwellenwert, wird die Klasse 0 vorhergesagt.

Die logistische Regression wird mit der Zielvariable Churn durchgeführt. Für das Preprocessing werden dieselben Schritte wie bei der linearen Regression angewandt. Der Algorithmus erzielt eine Accuracy von 88.63 Prozent. Dieses Ergebnis ist aber mit Vorsicht zu genießen, da der Recall mit 96.30 Prozent bei der Klasse der nicht abgewanderten Kunden zwar sehr hoch ist, allerdings bei den abgewanderten Kunden lediglich bei 50.35 Prozent liegt. Dazu erzielt das Modell ein kappa von 0.533 und ein squared correlation von 0.298.

5.1.7 C4.5 Decision Tree Algorithmus

Der C4.5 Decision Tree Algorithmus erstellt einen Entscheidungsbaum, indem die Entropie bei jeder Aufteilung minimiert wird. Im Anschluss wird eine Beschneidung (pruning) des Baums durchgeführt. Der Algorithmus ist für Klassifizierung geeignet.

Data Preprocessing Der Algorithmus benötigt kein besonderes Data Preprocessing und kommt auch mit Fehlwerten zurecht. Daher wurde vor dem Training nur das entsprechende Attribut als Label gesetzt. Möchte man das Attribut Complain vorhersagen, muss der kausale Zusammenhang beachtet werden und daher darf das Attribut Churn beim Training nicht verwendet werden.

Training Das Training wird für die Zielvariablen Churn und Complain durchgeführt. Als Metriken werden Accuracy, Kappa, Class Precision und Class Recall verwendet. Für alle Tests wird wie vorgegeben eine 10-Fache automatische Kreuzvalidierung durchgeführt.

Die zwei wichtigsten Parameter des Trainings:

- M, minimale Anzahl von Instanzen pro Blatt
- C, Schwellwert für das Beschneiden (pruning)

Die optimalen Parameter des Trainings wurden mit einem Grid Search ermittelt. Dabei soll der Kappa Wert möglichst groß sein.

Optimale Trainings Parameter

Label	M	C
Churn	1	0.698
Complain	1	0.607

Ergebnisse Für das Label Churn wurde eine Accuracy von 96.71% \pm 0.71% und ein Kappa von 0.883 \pm 0.030 erreicht. Ist das Label False beträgt der Class Recall 97.93% und Class Precision 98.12% und ist deutlich größer als wenn das Label True ist. Ist das Label True wird 90.72% bzw. 89.86% erreicht.

Für das Label Complain wurde eine Accuracy von 91.10% \pm 1.22% und ein Kappa von 0.781 \pm 0.030 erreicht. Ist das Label False beträgt der Class Recall 93.86% und Class Precision 93.70% und ist wieder deutlich größer als wenn das Label True ist. Ist das Label True wird 84.16% bzw. 84.53% erreicht.

Betrachtet man den erstellten Entscheidungsbaum für das Label Churn genauer, fällt auf, dass dieser mit 405 Blättern recht unübersichtlich ist. Die erste Aufteilung wird anhand von Tenure durchgeführt. Zu Beginn wird meistens anhand von Complain und CashbackAmount aufgeteilt. Die anderen Attribute werden erst

bei einer feineren Aufteilung verwendet. Es gibt ein sehr großes reines Blatt nach wenigen Aufteilungen mit insgesamt 661 Werten.

```
Tenure > 1
|   CashbackAmount > 124.79
|   |   Tenure > 21: false (661.0)
```

Das bedeutet, wenn ein Kunde länger als 21 Monate treu ist und ein Cashback-Amount von über 124.79 hat wandert er nicht ab. Für Churn gleich True, gibt es kein so großes Blatt. Die zwei größten Blätter haben 46 bzw. 30 Daten und sind auch sehr nah beieinander.

```
Tenure <= 1
|   Complain = true
|   |   NumberOfDeviceRegistered > 2
|   |   |   NumberOfAddress <= 3
|   |   |   |   DaySinceLastOrder <= 1
|   |   |   |   |   WarehouseToHome > 8
|   |   |   |   |   |   PreferredOrderCat = Mobile
|   |   |   |   |   |   |   DaySinceLastOrder <= 0: true (30.0)
|   |   |   |   |   |   |   PreferredOrderCat = Mobile Phone: true (46.0)
```

Zusammen repräsentieren diese zwei Blätter fast ein Zehntel der abgewanderten Kunden. Daher sind die Attributwerte ein Indiz dafür, dass ein Kunde eher abwandert.

Der Entscheidungsbaum für Complain als Zielvariable ist mit 1006 Blätter noch einmal größer und es gibt auch keine vergleichbaren großen Blätter. Dennoch wird wieder als erste Aufteilung Tenure verwendet. Die zwei kompletten Entscheidungsbäume können im Rapidminer unter
results
tree angeschaut werden

5.1.8 Classification and Regression Trees (CART)

Der CART Algorithmus erstellt Binärärbäume, d.h. bei jeder Spaltung werden die Daten in zwei Teile aufgeteilt. Für jede Aufteilung wird versucht die Reinheit der resultierenden Knoten zu maximieren. Zugleich wird versucht, dass jeder Knoten die Daten möglichst gleichmäßig aufteilt. Die Kombination davon ist das Gütemaß für die Aufteilung.

Data Preprocessing Der Algorithmus benötigt kein besonderes Data Preprocessing. Jedoch kommt er nicht mit Fehlwerten zurecht und somit wird der bereinigte Datensatz verwendet. Vor dem Training wurde nur die entsprechende Zielvariable als Label gesetzt. Möchte man das Attribut Complain vorhersagen, muss

der kausale Zusammenhang betrachtet werden und daher darf das Attribut Churn beim Training nicht verwendet werden.

Training Das Training wird für die Zielvariablen Churn und Complain durchgeführt. Als Metriken wird wieder Accuracy, Kappa, Class Precision und Class Recall verwendet. Für alle Tests wird wie vorgegeben eine 10-Fache automatische Kreuzvalidierung durchgeführt. Die Zwei wichtigsten Parameter des Trainings:

- M, minimale Anzahl von Instanzen pro Blatt
- N, Anzahl der Faltungen, die für das Pruning mit minimaler Kostenkomplexität verwendet werden

Die optimalen Parameter des Trainings wurden mit dem Grid Search Verfahren ermittelt. Dabei soll das Kappa möglichst groß sein.

Optimale Parameter des Trainings

Label	M	N
Churn	1	5
Complain	1	6

Für das Label Churn wurde eine Accuracy von 97.32% +/- 0.53 und ein Kappa von 0.905 +/- 0.019 erreicht. Ist das Label False beträgt der Class Recall 98.25% und Class Precision 98.52% und ist deutlich größer als wenn das Label True ist (92.72% bzw. 91.47%).

Für das Label Complain wurde eine Accuracy von 92.06% +/- 1.73% und eine Kappa von 0.806 +/- 0.041 erreicht. Ist das Label False beträgt der Class Recall 93.70% und Class Precision 93.86% und ist wieder deutlich größer als wenn das Label True ist (86.97% bzw. 85.43%).

Beim Blick auf den erstellten Entscheidungsbaum für die Zielvariable Churn fällt auf, dass die erste Aufteilung wieder nach dem Attribut Tenure durchgeführt wurde. Es gibt wieder einige reine Blätter. Diese entstehen aber erst nach häufigen Aufteilung der Daten. Das Größten Blatt hat z.B. eine Tiefe von zehn.

```

Tenure >= 1.5
| CashbackAmount >= 124.875
| | Complain!=(true)
| | | CouponUsed < 6.5
| | | | PreferredOrderCat!=(Fashion)
| | | | | OrderCount < 13.5
| | | | | | CashbackAmount < 323.39
| | | | | | | DaySinceLastOrder >= 1.5
| | | | | | | | PreferredPaymentMode!=(E wallet)|(Credit Card)
| | | | | | | | | SatisfactionScore < 4.5
| | | | | | | | | | NumberofAddress < 8.5: false(922.0/0.0)

```

Mit diesem einzelnen Blatt kann man keine allgemeine Aussage treffen, es ist aber ein genereller Indikator, dass ein Kunde eher nicht abwandert wenn Tenure ≥ 1.5 , CashbackAmount ≥ 124.875 und er sich nicht beschwert hat. Nach den ersten 3 Aufteilungen fällt auf, dass es einige solcher reinen Blätter gibt, die False sind und mehr als hundert Datenpunkte haben. Allerdings gibt es nur sehr vereinzelte Blätter die True sind. Diese haben auch nur ein paar wenige Werte. Alle Blätter für Churn gleich True und mehr als 20 Daten besitzen:

```

Tenure < 1.5
| Complain!=(true)
| | NumberofAddress >= 4.5
| | | NumberOfDeviceRegistered >= 2.5
| | | | OrderAmountHikeFromlastYear < 18.5
| | | | | MaritalStatus=(Single): true(46.0/0.0)
| | | | | MaritalStatus!=(Single)
| | | | | | DaySinceLastOrder >= 3.5: true(20.0/0.0)

Tenure < 1.5
| Complain!=(true)
| | NumberofAddress < 4.5
| | | DaySinceLastOrder < 1.5
| | | | PreferredPaymentMode=(Cash on Delivery)|(E
    wallet)|(COD)|(UPI)
| | | | | PreferredLoginDevice=(Computer)|(Mobile Phone)
| | | | | | HourSpendOnApp >= 2.5: true(28.0/0.0)

Tenure >= 1.5
| CashbackAmount < 124.875
| | Complain=(true)
| | | PreferredOrderCat=(Mobile
    Phone)|(Mobile)|(Others)|(Fashion)|(Grocery)
| | | | HourSpendOnApp >= 2.5
| | | | | PreferredPaymentMode=(UPI)|(CC)|(E wallet)|(Credit
    Card)|(Debit Card)|(COD)
| | | | | | NumberofAddress >= 1.5: true(28.0/0.0)

Tenure < 1.5
| Complain=(true)
| | NumberofAddress < 3.5
| | | DaySinceLastOrder >= 1.5
| | | | DaySinceLastOrder < 5.5
| | | | | MaritalStatus=(Single)
| | | | | | | WarehouseToHome >= 9.5
| | | | | | | | SatisfactionScore < 4.0: true(24.0/0.0)

```

Bei den Blättern sind kaum Gemeinsamkeiten zu erkennen. Selbst wenn eine Kunde sich nicht beschwert wandert er trotzdem manchmal ab. Jedoch jedes Blatt gibt eine Kombination von Attributen an, bei denen viele abwandern.

5.1.9 Random Forest

Random Forest ist ein Bagging Algorithmus der Entscheidungsbäume verwendet. Es werden viele verschiedene Entscheidungsbäume erstellt und daraus wird ein Ensemble Modell gebildet. Random Forest kann sowohl für Klassifikation, als auch für Regression verwendet werden.

Data Preprocessing Der Algorithmus benötigt kein besonderes Data Preprocessing und kommt auch mit Fehlwerten zurecht. Daher wird vor dem Training nur das entsprechende Attribut als Label gesetzt. Möchte man das Attribut Complain vorhersagen, muss der kausale Zusammenhang betrachtet werden und daher darf das Attribut Churn beim Training nicht verwendet werden.

Training Eine Klassifikation wird mit den Zielvariablen Churn und Complain und eine Regression mit der Zielvariable CashbackAmount durchgeführt. Bei der Klassifikation werden als Metriken Accuracy, Kappa, Class Precision und Class Recall verwendet. Bei der Regression werden als Metriken RMSE und MAE verwendet. Für alle Tests wird wie vorgegeben eine 10-Fache automatische Kreuzvalidierung durchgeführt. Die wichtigsten Parameter des Trainings für den Random Forest:

- Number of trees, die Anzahl an Bäumen
- Maximal depth, die maximale Tiefe jedes Baums
- Confidence, Wert für das Pruning
- Subset ratio, Verhältnis der zufällig ausgewählten Attribute, die getestet werden sollen
- criterion, Kriterium, nach dem jeder Baum erstellt wird.

Die Parameter des Trainings Maximal depth, confidence und subset ratio wurden mit dem Evolutionary Optimizer bestimmt. Dabei soll das Kappa bei der Klassifikation möglichst groß sein bzw. der RMSE bei der Regression möglichst klein. Die Number of trees wurde auf 200 gesetzt. Grundsätzlich werden die Ergebnisse nicht schlechter bei einer höheren Anzahl an Bäumen, es entsteht jedoch ein höherer Rechenaufwand. Für das Kriterium bei der Klassifikation wurde information gain, gain ratio, gini index und accuracy getestet und bei der Regression wird das least square als Kriterium verwendet.

Optimale Parameter des Trainings

Label	maximal depth	confidence	subset ratio	subset ratio
Churn	94	0.0051	0.6798	information gain
Complain	94	0.0227	0.6973	gini index
CashbackAmount	94	-	0.6974	least square

Ergebnisse Für das Label Churn wurde eine Accuracy von 97.69% +/- 0.69% und ein Kappa von 0.916 +/- 0.026 erreicht. Ist das Label False beträgt der Class Recall 99.12% und Class Precision 98.12% und ist deutlich größer als wenn das Label True ist.

Ergebnisse Churn

	true false	true true	class precision
pred.fasle	4641	89	98.12%
pred. true	41	859	95.44%
class recall	99.12%	90.61%	

Für das Label Complain wurde eine etwas schlechtere Accuracy von 95.03% +/- 1.31% und ein Kappa 0.873 +/- 0.035 erreicht. Ist das Label False beträgt der Class Recall 99.21% und Class Precision 93.86% und ist wieder deutlich größer als wenn das Label True ist.

Ergebnisse Complain

	true false	true true	class precision
pred.fasle	4548	434	91.29%
pred. true	134	514	79.32%
class recall	97.14%	54.22%	

Bei der Regression mit der Zielvariablen CashbackAmount wurde eine RMSE von 8.815 +/- 2.545 und MAE von 4.078 +/- 0.460 erreicht. Mit dem Random Forest wurden die besten Ergebnisse erreicht, allerdings ist es hier schwer möglich das Modell genauer zu untersuchen, da es aus 200 Entscheidungsbäume besteht und jeder einzelne Baum overfitted. Aber ein Ensemble Modell aus allen ergibt ein sehr gutes Modell.

5.1.10 Rule Induction

Bei der Rule Induction wird versucht für ein Teil der Daten eine Regel zu finden, sodass dieser Teil der Daten richtig klassifiziert wird. Im Anschluss werden diese Daten entfernt und es wird für die übrigen Daten wieder eine Regel gesucht. Diese wird solang gemacht bis keine Daten mehr übrigbleiben.

Data Preprocessing Der Algorithmus benötigt kein besonderes Data Preprocessing und kommt auch mit Fehlwerten zurecht. Daher wurde vor dem Training nur das entsprechende Attribut als Label gesetzt. Möchte man das Attribut Complain vorhersagen, muss der kausale Zusammenhang betrachtet werden und daher darf das Attribut Churn beim Training nicht verwendet werden.

Training Das Training wird mit den Zielvariablen Churn und Complain durchgeführt. Als Metriken werden Accuracy, Kappa, Class Precision und Class Recall verwendet. Für alle Tests wird wie vorgegeben eine 10-Fache automatische Kreuzvalidierung durchgeführt.

Die wichtigsten Parameter des Trainings:

- sample ratio, Stichprobenverhältnis der Trainingsdaten, die für das Wachsen und Beschneiden verwendet werden
- pureness, gewünschte Reinheit, d.h. wie korrekt eine Regel mindestens sein soll.
- minimal prune benefit, Schwellwert für die Bescheidung

Die optimalen Parameter des Trainings werden mit dem Evolutionary Optimizer bestimmt. Dabei soll das Kappa möglichst groß sein.

Optimalen Parameter des Trainings

Label	sample ratio	pureness	minimal prune benefit
Churn	0.9187	0.9753	0.8618
Complain	8974	0.9603	0.5164

Ergebnisse Für das Label Churn wurde eine Accuracy von 89.91% +/- 1.32% und ein Kappa von 0.587 +/- 0.058 erreicht. Für das Label Complain wurde eine etwas schlechtere Accuracy von 73.18% +/- 2.07% und ein Kappa 0.233 +/- 0.051 erreicht. Die erreichten Ergebnisse sind schlechter wie von den anderen Methoden. Allerdings sind die gefundenen Regeln leichter nachvollziehbar. Vor allem die ersten Regeln sind für die Interpretation interessant, da die unteren Regeln nur zutreffen, wenn alle Regeln darüber nicht zutreffen.

Die ersten fünf Regeln mit Churn als Zielvariable:

- if Tenure > 3.500 and Complain = false and WarehouseToHome \leq 13.500 then false (1371 / 30)
- if Tenure > 3.500 and PreferredOrderCat = Laptop & Accessory and MaritalStatus = Married then false (582 / 11)
- if Tenure > 3.500 and Complain = false and CityTier < 2.500 then false (710 / 34)

- if $\text{Tenure} > 3.500$ and $\text{NumberOfAddress} \leq 8.500$ and $\text{Tenure} > 13.500$ and $\text{Gender} = \text{Female}$ then false (192 / 2)
- if $\text{Complain} = \text{false}$ and $\text{DaySinceLastOrder} > 1.500$ then false (830 / 188)

Es wird deutlich, dass Tenure und Complain bei der Klassifikation entscheidend sind.

Die Regeln mit Complain als Zielvariable gibt es sehr viele Regeln die meistens False klassifizieren. Diese wird auch am geringen Kappa sichtbar. Alle Regeln können im Rapidminer unter *results tree* angeschaut werden.

5.2 Unsupervised Learning

5.2.1 Apriori-Algorithmus

Funktionsweise

Der Apriori-Algorithmus (AA) ist ein Algorithmus zur Assoziationsanalyse. Das Ziel ist das Aufdecken von Zusammenhängen in den gegeben Daten. Dazu werden diese Kombination von Attributen gesucht, die hohe Wahrscheinlichkeiten haben. Um die Funktionsweise des AA zu erläutern werden folgende Grundbegriffe eingeführt um die Daten im Kontext der Assoziationsanalyse zu beschreiben:

- Eine Menge möglicher **Items** $I = \{i_1, i_2, \dots, i_m\}$.
- Eine Menge möglicher **Itemmenge** $X = \{x_1, x_2, \dots, x_j\}$, welche aus Items bestehen: $x_i \subset I$.
- Eine Menge von **Transaktionen** $T = \{t_1, t_2, \dots, t_m\}$ mit $t_i \subseteq I$.
- **Assoziationsregeln** $X \rightarrow Y$, wobei $(X \subset I) \wedge (Y \subset I) \wedge (X \cap Y = \emptyset)$.

wobei die Assoziationsregeln als Aussagen der Form "*Wenn X eingetreten ist, dann besteht eine hohe Wahrscheinlichkeit, dass auch Y eintritt*" zu verstehen sind. Die Länge eines Itemsets ist definiert als die Anzahl der Elemente in dem Itemset (Kardinalität).

- Die Menge der **k-itemsets** X_k ist die Menger der Itsemsets die k Items enthalten: $X_k = \forall x_i \in X, |x_i| \geq k$.
- Die **Itemset Frequency** F ist die Anzahl der Transaktionen $t_i \in T$ die ein bestimmtes Itemset x_j beinhalten.
- **Frequent Itemsets** sind solche Itemsets $x_j \in X$, welche eine Häufigkeit h höher als ein bestimmter Threshold haben: $h \geq \phi$.
- Die **F_k Häufigkeit** ist identisch mit der Definition von frequent Itemsets, nur dass eine Länge von mindestens k für die Itemsets vorausgesetzt wird.

Um eine sinnvolle Analyse durchführen zu können müssen die Regeln bewertet werden. Dazu verwendet der AA zwei Messwerte, den Support und die Konfidenz:

- Der **Support** ist die Wahrscheinlichkeit dass eine Itemmenge in einer Transaktion vorkommt.
- Die **Konfidenz** ist die Güte einer Regel. Das heißt, falls in einer Regel $X \rightarrow Y$ X eintritt, wie hoch ist dann die Wahrscheinlichkeit dass auch Y eintritt.

Da das Durchsuchen aller Itemmengen sehr Rechenaufwändig ist, macht der AA sich eine Eigenschaft zu Nutze, die den Suchraum der Regeln drastisch reduziert:

$$\begin{aligned} & \text{Wenn die Itemmenge } Z \text{ nicht häufig ist, dann wird auch} \\ & \forall Z \cup i_j, \quad i_j \in I \text{ nicht häufig sein.} \end{aligned} \tag{5.9}$$

Das bedeutet, dass wenn wir eine Itemmenge haben, die nicht F_k häufig ist, dann wird auch keine weitere Erweiterung mit dieser Itemmenge häufig sein.

Der AA sucht nun zuerst F_1 häufige Itemmengen, danach F_2 häufige und so weiter. AA bricht ab falls nur noch eine F_k häufige Itemmenge gefunden wird oder falls ein größten- oder Zeitlimit erreicht wird.

Mit den gefundenen Itemmengen werden nun alle möglichen Assoziationsregeln ausprobiert und auf Güte bewertet. Alle Regeln die einer gewissen definierten Güte entsprechen werden als sinnvolle Regeln beachtet, die anderen werden verworfen.

Zur sinnvollen Evaluierung gibt es einige Metriken:

- **Leverage:** der Anstieg in der Sicherheit, also der Unterschied zwischen der Wahrscheinlichkeit, dass X und Y zusammen auftreten gegen ihre Einzelwahrscheinlichkeiten. $\text{Leverage}(X \rightarrow Y) = P(X \cap Y) - P(X) * P(Y)$.
- **Lift:** die Rate des Anstiegs in Sicherheit. $\text{Lift}(X \rightarrow Y) = \frac{P(Y|X)}{P(Y)}$.
- **Conviction:** Der umgekehrte Lift: $\text{Conviction}(X \rightarrow Y) = \frac{1-P(Y)}{1-P(Y|X)}$.
- **Confidence Ratio:** Positiver oder Negative Rate von Sicherheit und der theoretischer Wahrscheinlichkeit. $\text{Confidence Ratio}(X \rightarrow Y) = 1 - \min\left(\frac{P(Y|X)}{P(Y)}, \frac{P(Y)}{P(Y|X)}\right)$.

Training

- AA kann nur mit nominalen Attributen verwendet werden, deshalb ist es nötig numerische Attribute entweder zu diskreditieren.
- Da bei AA Kreuzvalidierung keinen Sinn macht, wird diese nicht durchgeführt.
- Als Metriken für die gefundenen Assoziationen wird die Confidence, der Leverage, Lift und die Conviction verwendet.
- Um sinnvolle Regeln zu finden, wird die Anzahl der Bins variiert.

Ergebnisse

Die Ergebnisse zeigen jeweils die relevanten Regeln, in denen die Confidence mindestens 0.7 beträgt.

5 Bins

Regel	Confidence	Leverage	Lift	Conviction
$\text{CashbackAmount} = (65-130) \rightarrow \text{PreferredOrderCat} = \text{Mobile}$	0.88	0.1	6.14	7.24
$\text{PreferredOrderCat} = \text{Fashion} \rightarrow \text{CashbackAmount} = (194.994 - 259.992)$	0.77	0.09	4.89	3.57
$\text{CashbackAmount} = (195-260) \rightarrow \text{PreferredOrderCat} = \text{Fashion}$	0.72	0.09	4.89	3.01
$\text{PreferredPaymentCat} = \text{E wallet} \rightarrow \text{CityTier} = (2.6-\infty)$	1.0	0.08	3.27	426.2
$\text{PreferredOrderCat} = \text{Mobile} \rightarrow \text{PreferredLoginDevice} = \text{Phone}$	0.71	0.07	3.25	2.69

20 Bins

Regel	Confidence	Leverage	Lift	Conviction
$\text{CashbackAmount} = (114-130) \rightarrow \text{PreferredOrderCat} = \text{Mobile}$	0.92	0.1	6.4	10.48
$\text{PreferredOrderCat} = \text{Mobile} \rightarrow \text{CashbackAmount} = (114-130)$	0.82	0.1	6.4	4.83
$\text{PreferredPaymentCat} = \text{E wallet} \rightarrow \text{CityTier} = (2.9-\infty)$	1.0	0.08	3.27	426.2
$\text{PreferredOrderCat} = \text{Mobile} \rightarrow \text{PreferredLoginDevice} = \text{Mobile}$	0.71	0.07	3.25	2.69
$\text{CouponUsed} = (-\infty-0.8) \rightarrow \text{OrderCount} = (-\infty-1.75)$	0.81	0.09	2.6	3.57
$\text{PreferredOrderCat} = \text{Mobile} \rightarrow \text{OrderCount} = (-\infty-1.75)$	0.72	0.06	2.32	2.46

50 Bins

Regel	Confidence	Leverage	Lift	Conviction
$\text{PreferredPaymentMethod} = \text{E wallet} \rightarrow \text{CityTier} = (2.960-\infty)$	1.0	0.08	3.27	426.2
$\text{PreferredOrderCat} = \text{Mobile} \rightarrow \text{PreferredLoginDevice} = \text{Phone}$	0.71	0.07	3.25	2.69
$\text{CouponUsed} = (-\infty-0.32) \rightarrow \text{OrderCount} = (-\infty-1.3)$	0.81	0.09	2.6	3.57
$\text{PreferredOrderCat} = \text{Mobile} \rightarrow \text{OrderCount} = (-\infty-1.3)$	0.72	0.06	2.32	2.46

Die Ergebnisse zeigen einen klaren Zusammenhang zwischen CashbackAmount und PreferredOrderCat . Bei 5 Bins sieht man, dass bei einem CashbackAmount im Bereich 65-130 zu 88% die PreferredOrderCat Mobile ist. In einem hohen Bereich für CashbackAmount (195-260), ist die PreferredOrderCat Fashion sehr wahrscheinlich. Bei 20 Bins erhält man eine etwas feingliedrigere Aufteilung. Bei einem CashbackAmount im Bereich (114-130) ist die Konfidenz, die PreferredOrderCat Mobile zu haben 0.92%. Das ist wahrscheinlich die Gleiche Regel wie bei 5 Bins, nur genauer. Auch der umgekehrte Fall trifft zu, falls man die PreferredOrderCat Mobile hat, ist der CashbackAmount um 82% im Bereich (144-130). Bei 50 Bins, der kleinteiligsten Aufgliederung, sieht man, dass wohl nur Menschen aus der CityTier größer als 2.960 die $\text{PreferredPaymentMethod}$ E wallet besitzen. Das lässt vermuten, dass die Methode E wallet nur in größeren Städten verwendet wird.

Die Analyse zeigt, dass es einen Zusammenhang zwischen CashbackAmount und der PreferredOrderCat gibt. Auch scheinen Personen aus größeren CityTiers , eher ungewöhnliche Zahlungsmethoden ($\text{PreferredPaymentMethod}$) wie E wallets zu verwenden.

5.2.2 Frequent Pattern Growth

Funktionsweise

Der Frequent Pattern Growth (FPG) Algorithmus kann als Weiterentwicklung des AA gesehen werden. Mit dem FPG kann man Korrelationen in den Daten finden indem analysiert wird, welche Attributkombinationen am häufigsten vorkommen. Der FPG findet frequente Patterns über eine Baumstruktur.

Training

- FPG kann nur mit binominalen Attributen verwendet werden, deshalb ist es nötig numerische Attribute erst zu nominale Attribute umzuwandeln und anschließend alle Werte in binominale Werte umzuwandeln.
- Da bei FPG Kreuzvalidierung keinen Sinn macht, wird diese nicht durchgeführt.

Ergebnisse

Da die numerischen Attribute für FPG in binominale Attribute umgewandelt werden müssen, geben diese nur einen sehr groben Bereich an.

Frequent Item Sets

Zuerst werden die gefundenen interessanten Frequent Itemsets untersucht, welche mindestens 2 Items enthalten (einelementrige Itemsets machen keinen Sinn).

Support	Item 1	Item 2	Item 3
0.361	PreferredLoginDevice = Mobile Phone	CashbackAmmount = (163.280- ∞)	
0.267	PreferredLoginDevice = Mobile Phone	Tenure = (9.5- ∞)	
0.220	PreferredLoginDevice = Mobile Phone	Tenure = (9.5- ∞)	CashbackAmmount = (163.280- ∞)

Aus diesen Frequent Item Sets lässt sich folgen, dass die Verwendung eines Mobiltelefons, die Tenure und der Cashback Amount zusammenhängen. Das lässt vermuten, dass die Verwendung über ein Mobiltelefon mindestens nicht schädlich für das Geschäft ist was heißt die Optimierung für Mobiltelefone ist in Ordnung, denn die Benutzer die über ein Mobiltelefon zugreifen haben eine lange Tenure und erhalten viel Cashback, es sind also eher Kunden die oft auf das Angebot zugreifen.

Association Rules

Es werden die gefundenen interessanten Frequent Itemsets untersucht, welche mindestens 2 Items enthalten.

Lift	Premises	Conclusion
1.647	CashbackAmmount = (163.280- ∞)	PreferredLoginDevice = Mobile Phone, Tenure = (9.5- ∞)
1.661	PreferredLoginDevice = Mobile	CashbackAmmount = (163.280- ∞)
1.463	Tenure = (9.5- ∞)	CashbackAmmount = (163.280- ∞)
1.732	DaySinceLastOrder = (4.5- ∞)	CuponUsed = (1.5- ∞)

Die Association Rules zeigen noch deutlicher, dass der CashbackAmmount und das PreferredLoginDevice zusammenhängen. Falls der CashbackAmmount 163.280 oder höher beträgt, ist die Chance dass ein Mobil Phone verwendet wird 64,7% höher als ohne diese Prämisse. Ebenso verhält es sich mit dem Tenure. In die andere Richtung ist es ebenso sehr viel wahrscheinlicher (66,1%), dass ein hoher CashbackAmmount vorliegt, falls ein Mobile Phone verwendet wird. Eine weitere Beobachtung ist, dass falls die Tage seit der letzten Bestellung (DaysSinceLastOrder) mehr als 4.5 sind, werden zu 73,2% wahrscheinlicher Cupons verwendet, als wenn die letzte Bestellung kürzer her ist. Das lässt vermuten, dass Cupons eine Wirkung auf getätigte Bestellungen haben, denn Kunden die länger nichts gekauft haben neigen dazu, eher Cupons zu verwenden, welche sie vermutlich zu einem erneuten Einkauf ermutigt haben.

5.2.3 K-Means

Das K-Means Verfahren ist ein zentroides Clusterverfahren welches ähnliche Datenobjekte einer vorher festgelegten Anzahl von Gruppen zuordnet.

Das K-Means Verfahren minimiert dabei die Summe der quadrierten Euklidischen Distanz, wodurch jeder Datenpunkt dem nächstgelegenen Clusterschwerpunkt zugeordnet wird. Die dazugehörige Zielfunktion gilt es hierbei zu minimieren:

$$J = \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \quad (5.10)$$

x_j = Datenpunkte

μ_i = Clustermittelpunkt

S_i = Cluster

Datenvorverarbeitung

Für die Durchführung der K-Means Methode wird der bereits von Fehlwerten bereinigte Datensatz verwendet, die Schritte hierzu werden ausgiebig im Kapitel Preprocessing besprochen. Zudem werden nur numerische Werte beachtet, Binomiale Werte werden Binärwerten konvertiert.

Da es sich bei der K-Means Methode um ein distanzbasiertes Verfahren handelt müssen die numerischen Werte zu Beginn normalisiert werden, da sonst numerisch größere Wertebereiche mehr Einfluss auf die Minimierungsfunktion hätten.

Nach einer statistischen Analyse der Daten zeigt sich, dass die meisten Werte keiner Normalverteilung folgen und keine Nennenswerten Ausreiser vorhanden sind. Aus diesem Grund bietet eine Min-Max Normalisierung hier mehr Vorteile.

Modell

Die optimale Clusteranzahl wird per Hyperparametersuche durch die Optimierung des Davis Bouldin Index [DB79] bestimmt.

iteration	Clustering (2...)	Davies Bouldin ↑
16	17	1.273
9	10	1.330
8	9	1.355
10	11	1.356
14	15	1.369
5	6	1.375
17	18	1.380
11	12	1.389

Fig. 14 Hyperparameteroptimierung für die Anzahl der Cluster k

Es stellt sich heraus dass eine Clusteranzahl von 17 laut Davis Bouldin Index hier das beste Ergebniss produziert. Auf Grund der besseren und sinnhaften Interpretierbarkeit der Cluster ist es hier jedoch sinnvoll, einen schlechteren Davis Bouldin Index zu wählen und das Modell mit 6 Clustern zu erstellen.

Um die Abstände der Datenpunkte zu ihren Nachbarn zu berechnen wird die Bregman Divergenz[BMD⁺05] mit der quadratischen euklidischen Distanz verwendet.

Ergebnisse

Wie die Grafik zeigt, sind die Cluster gut gleichverteilt. Cluster 3 hat etwas mehr Datenpunkte und repräsentiert eine größere Kundengruppe.

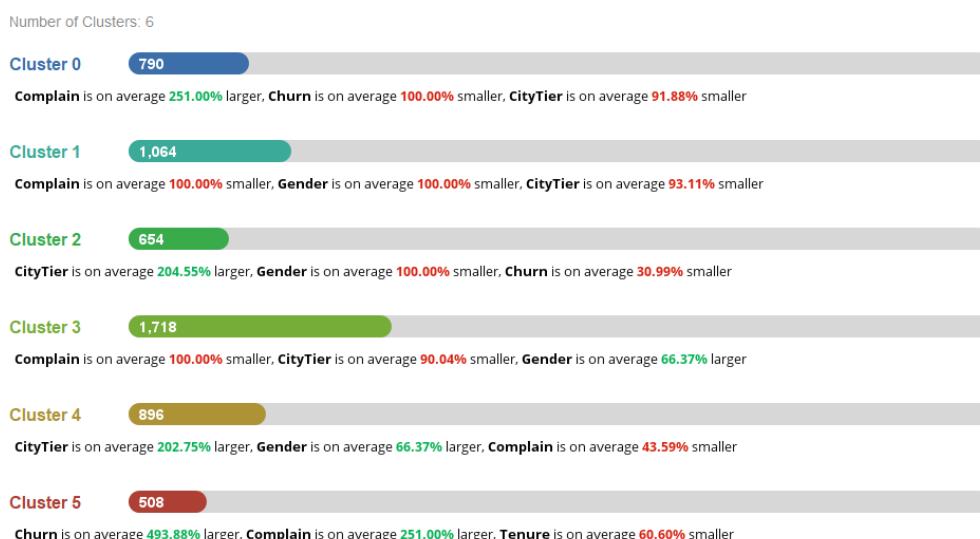


Fig. 15 Die Grafik zeigt einen Überblick der Cluster

Durch das Clustering können folgende Vermutungen über die verschiedenen Kundengruppen des Unternehmens angestellt werden:

Cluster 0: In diesem Cluster finden sich Kunden, die in einem geringen *City Tier* angehören, wordurch angenommen werden kann dass diese Kunden im Schnitt über weniger Wohlstand und Bildung verfügen. Im Hinblick auf die Sinus-Milieus [Kal07], kann die Annahme getroffen werden dass diese Kundengruppe aus einer Konsumorientierten Unterschicht stammt.

Cluster 1: Dieses Cluster repräsentiert ebenfalls Kundengruppen aus einer konsumorientierten Unterschicht. Diese Schicht besteht aber vorwiegend aus Frauen.

Cluster 2: In diesem CLuster befindet sich die Kundengruppe, welche aus einem eher wohlhabenderen Hintergrund stammt. Diese Kundengruppe wird ebenfalls vorwiegend durch Frauen repräsentiert.

Cluster 3: In diesem Cluster finden wir überwiegend Männer aus einer niedrigen Einkommensschicht. Den Werten sind zu entnehmen, dass hier eher zufriedene Kunden angesiedelt sind.

Cluster 4: Genau wie in CLuster 3 finden sich in CLuster 4 ebenfalls vorwiegend Männer, die aber anders als in CLuster 3 überwiegend zur Mittel- und Oberschicht gehören. Über die Werte ist anzunehmen, dass dies zufriedene männliche Kunden der oberen Schichten sind.

Cluster 5: Dieses Cluster ist besonders interessant. Auf Grund der Werte ist anzunehmen, dass sich in diesem Cluster die unzufriedenen Kunden befinden, welche das Unternehmen verlassen.

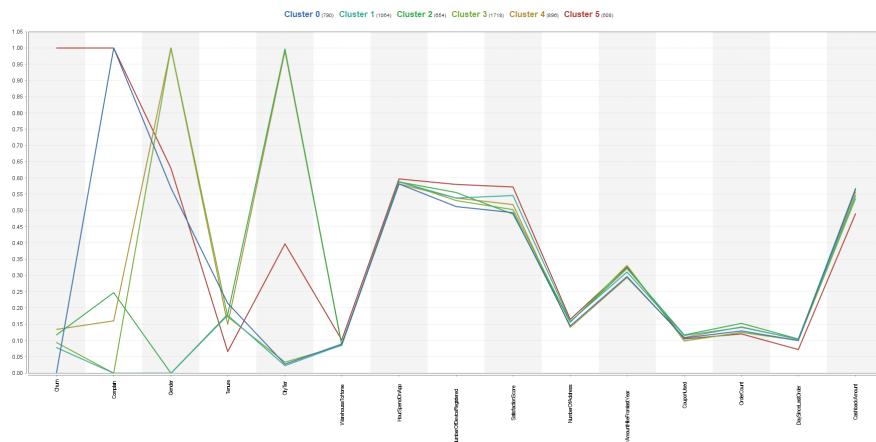


Fig. 16 Die Grafik zeigt das Centroid-Chart



Fig. 17 Die Grafik zeigt eine Heatmap der wichtigsten Attribute

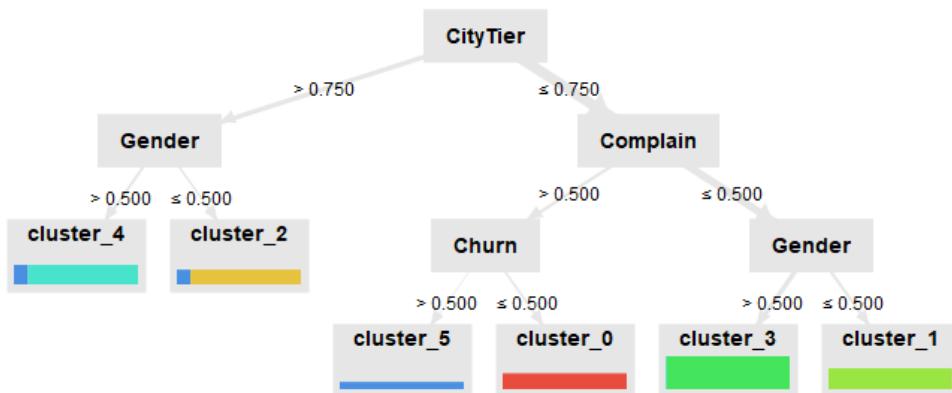


Fig. 18 Die Grafik zeigt die wichtigsten Attribute als Baum

Im Baum werden die wichtigsten Attribute für die Entscheidung einer Clusterzugehörigkeit noch einmal grafisch dargestellt. Es ist gut zu sehen dass die Attribute:

- CityTier
- Gender
- Complain

die drei wichtigsten Faktoren bei der Entscheidung einen Datenpunkt in ein bestimmtes Cluster zu ordnen, darstellten.

5.2.4 K-Medoid

Das K-Medoid Verfahren kann als Erweiterung des K-Means Verfahren verstanden werden. Während das K-Means Verfahren die Summe des quadratischen euklidischen Fehlers minimiert, minimiert das K-Medoid verfahren die Distanz zwischen den verschiedenen Punkten. Zudem werden beim K-Means Klustermittelpunkte gewählt, welche über den durchschnitt der umliegenden Punkte berechnet wird, was bedeutet dass diese Punkte nicht existieren. Im Gegensatz hierzu wählt das K-Medoid Verfahren einen Datenpunkt als Clustermittelpunkt.

Der Algorithmus kann wie folgt beschrieben werden:

1. Initialisierung erfolgt gleich wie beim K-Means Verfahren.
2. Clusterzuordnung erfolgt gleich wie beim K-Means Verfahren.
3. Für die Aktualisierung der Clustermittelpunkte werden die Centroide mit den ($m-1$) Punkten des Clusters getauscht und der Punkt mit dem geringesten Loss als neuer Centroid ausgesucht.

Datenvorverarbeitung

Für das K-Medoid Verfahren erfolgt die selbe Datenvorverarbeitung wie für das K-Means Verfahren

Es wird der bereits von Fehlwerten bereinigte Datensatz verwendet, die Schritte hierzu werden ausgiebig im Kapitel Preprocessing besprochen. Zudem werden nur numerische Werte beachtet, Binomiale Werte werden Binärwerten konvertiert.

Da es sich bei der K-Means Methode um ein distanzbasiertes Verfahren handelt müssen die numerischen Werte zu Beginn normalisiert werden, da sonst numerisch größere Wertebereiche mehr Einfluss auf die Minimierungsfunktion hätten.

Nach einer statistischen Analyse der Daten zeigt sich, dass die meisten Werte keiner Normalverteilung folgen und keine Nennenswerten Ausreiser vorhanden sind. Aus diesem Grund bietet eine Min-Max Normalisierung hier mehr Vorteile.

Modell

Die optimale Clusteranzahl wird per Hyperparametersuche durch die optimierung des Davis Bouldin Index [DB79] bestimmt.

iteration	Clusteri...	Davies Bouldin ↑
7	8	1.351
14	15	1.399
9	10	1.439
6	7	1.486
17	18	1.510
10	11	1.543
8	9	1.561

Fig. 19 Hyperparameteroptimierung für die Anzahl der Cluster k

Wie in der Auswertung der Hyperparameteroptimierung zu sehen ist, bringt ein Modell mit einer Clusteranzahl von 8 die besten Ergebnisse auf diesen Datensatz.

Ergebnisse

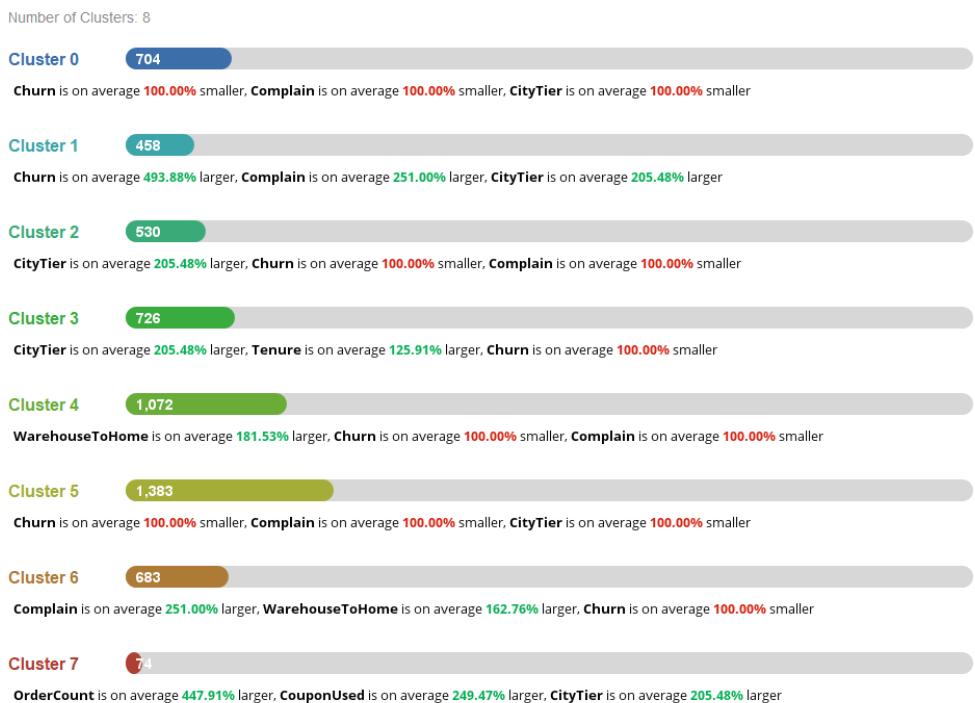


Fig. 20 Die Grafik zeigt einen Überblick der Cluster

Genneral generiert das K-Medoid ähnliche Aussagen wie das K-Means Modell. Im folgenden werden die Clusters kurz beschrieben und Annahmen über die Kundengruppen in diesem Cluster getroffen:

Cluster 0: In diesem Cluster finden sich Kunden wieder, die aus eher niedrigen Schichten stammen, aber dem Unternehmen eher treu bleiben und sich weniger beschweren. Hier sind also zufriedene Kunden der Unterschicht angesiedelt.

Cluster 1: In diesem Cluster sind Kunden der eher höheren Schichten angesiedelt. Den Werten ist zu entnehmen dass es sich hier um unzufriedene Kunden handelt die ihr Kundenverhältniss sehr häufig auflösen.

Cluster 2: In diesem Cluster befinden sich ebenfalls Kunden einer eher höheren Schicht. Im Gegensatz zu Cluster 2 sind hier jedoch Kunden angesiedelt, welche generell mit dem Service zufrieden sind.

Cluster 3: In Cluster 3 sticht hervor, dass es sowohl aus Kunden besteht, welche schon seit längerer Zeit Kunde des Unternehmens sind, wie auch der überdurchschnittliche *City Tier*. Anzunehmen sit, dass es sich in diesem Cluster um Langzeitkunden handelt, welche aus einer höheren Schicht stammen.

Cluster 4: In diesem Cluster befinden sich Kunden, die sich nicht unmittelbar in der Nähe eines Lagerhauses des Unternehmens befinden, und dennoch eher zufrieden mit dem Angeboteten Service sind.

Cluster 5: In diesem Cluster befinden sich Kunden die generell zufrieden sind mit dem angebotenen Service und aus einer eher niedrigeren Gesellschaftsschicht stammen.

Cluster 6: In diesem Cluster befinden sich Kundengruppen welche besonders weit weg von Lagerhäusern des Unternehmens angesiedelt sind. Diese Kundengruppe sit im schnitt eher UNzufrieden mit dem Service, da der *Complain* Wert sehr hoch ist. Hier könnte man weitere Daten erheben und untersuchen, ob dies im Zusammenhang mit der Entfernung zum nächsten Lagerhaus steht. Eventuelle Lieferverzögerungen oder Lieferengpässe könnten hier die Ursache sein.

Cluster 7: Dieses Cluster ist dadurch gekennzeichnet, dass Kunden Sehr viele Bestellungen aufgeben. Hier ist eine sehr konsumstarke Kundengruppe der höheren Gesellschaftsebenen anzutreffen.

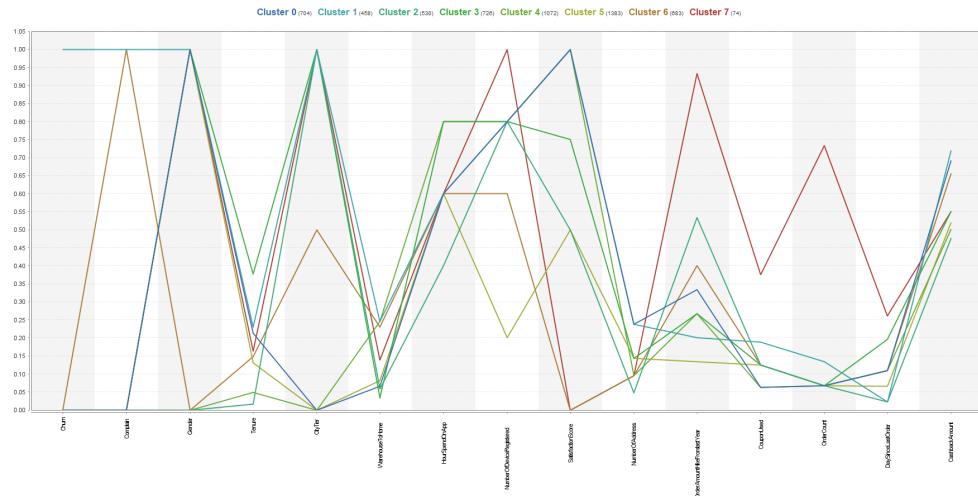


Fig. 21 Die Grafik zeigt das Centroid-Chart

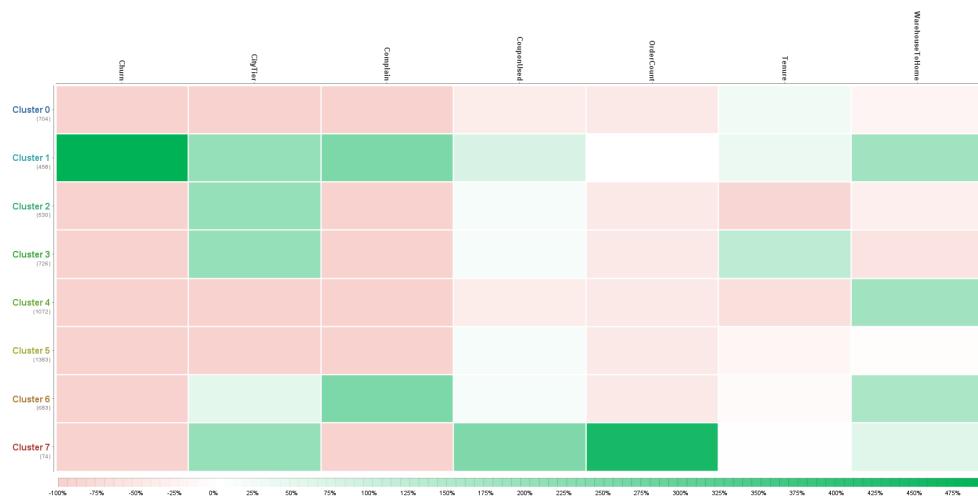


Fig. 22 Die Grafik zeigt eine Heatmap der wichtigsten Attribute

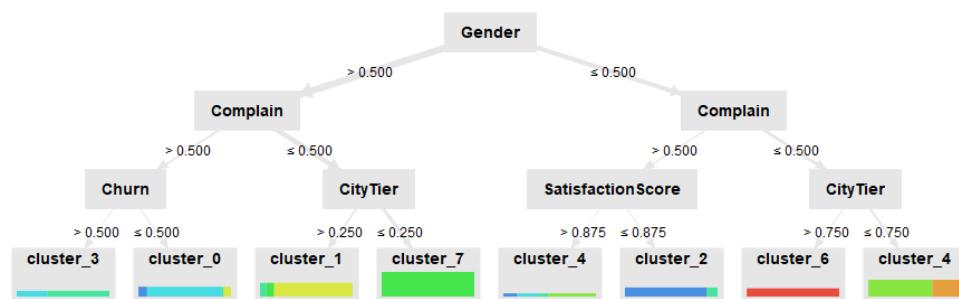


Fig. 23 Die Grafik zeigt die wichtigsten Attribute als Baum

5.2.5 Kernelized K-Means

Das Kernelized K-Means Verfahren ähnelt sehr stark dem herkömmlichen K-Means Verfahren. Der Unterschied besteht darin, dass die Features durch die Kernelization in höhere Räume abgebildet werden können, sodass im Gegensatz zum herkömmlichen K-Means auch nicht lineare Zusammenhänge betrachtet werden können. Es wird zusätzlich eine Featurefunktion beziehungsweise Kernelfunktion eingeführt:

$$J = \sum_{i=1}^k \sum_{x_j \in S_i} \|\phi(x_j) - \mu_i\|^2 \quad (5.11)$$

x_j = Datenpunkte

μ_i = Clustermittelpunkt

S_i = Cluster ϕ = Kernel – /Featurefunktion

Datenvorverarbeitung

Für das kernelized K-Means Verfahren erfolgt die selbe Datenvorverarbeitung wie für das K-Means Verfahren

Es wird der bereits von Fehlern bereinigte Datensatz verwendet, die Schritte hierzu werden ausgiebig im Kapitel Preprocessing besprochen. Zudem werden nur numerische Werte beachtet, Binomiale Werte werden Binärwerten konvertiert.

Da es sich bei der K-Means Methode um ein distanzbasiertes Verfahren handelt müssen die numerischen Werte zu Beginn normalisiert werden, da sonst numerisch größere Wertebereiche mehr Einfluss auf die Minimierungsfunktion hätten.

Nach einer statistischen Analyse der Daten zeigt sich, dass die meisten Werte keiner Normalverteilung folgen und keine Nennenswerten Ausreiser vorhanden sind. Aus diesem Grund bietet eine Min-Max Normalisierung hier mehr Vorteile.

Modell

Um das Beste Modell zu bestimmen werden beim Kernelized K-Means zusätzlich verschiedene Kerneltypen in die Liste der Hyperparameter aufgenommen.

Iteration	Clustering (2).k	Clustering (2).kernel_type	Clustering (2).kernel_degree	Davies Bouldin ↑
754	14	multiquadric	4.600	1.222
716	14	epanechnikov	4.600	1.225
1004	17	anova	6.400	1.232
258	12	epanechnikov	1.900	1.235
562	12	epanechnikov	3.700	1.241
278	13	gausian_combination	1.900	1.247
923	12	dot	6.400	1.248

Fig. 24 Hyperparameteroptimierung für die Anzahl der Cluster k und des Kernels

Nach Ausführung der Hyperparameteroptimierung zeigt sich eine Clusteranzahl von 14 in Kombination mit dem *multiquadratic* Kernel als vielversprechend. Der Parameter kernel degree kann in dem Fall ignoriert werden, da er keine Anwendung für diesen Kernel findet und nur als Konstante mitgeführt wurde.

$$k(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{\|\mathbf{x} - \mathbf{y}\|^2 + c}}$$

Fig. 25 Multiquadratic Kernel

Ergebnisse

Wie zu sehen ist, produziert das Modell stark unausgeglichene Cluster. Die Hyperparameteroptimierung des Kernelized K-Means durch RapidMiner funktioniert nur bedingt und braucht hier dringend Nachbesserung.

Auf Grund der sehr guten Ergebnisse der K-Means und K-Medoids Methode wird der Kernelized K-Means nicht weiter beschrieben.

Eine Interpretation der Ergebnisse könnte hier zu falschen Vermutungen führen.

Cluster Model

```

Cluster 0: 528 items
Cluster 1: 1076 items
Cluster 2: 0 items
Cluster 3: 0 items
Cluster 4: 0 items
Cluster 5: 0 items
Cluster 6: 0 items
Cluster 7: 0 items
Cluster 8: 0 items
Cluster 9: 0 items
Cluster 10: 0 items
Cluster 11: 0 items
Cluster 12: 2749 items
Cluster 13: 1277 items
Total number of items: 5630

```

Fig. 26 Die Grafik zeigt einen Überblick der gebildeten Cluster

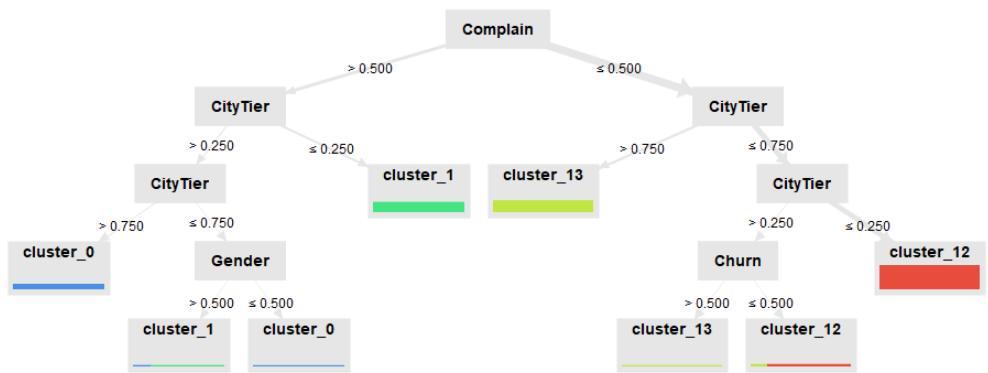


Fig. 27 Die Grafik zeigt den Kernel K-Means Algoritmus als Baum

5.2.6 Agglomerative Clustering

Datenvorverarbeitung

Für das Agglomerative Clustering Verfahren erfolgt die selbe Datenvorverarbeitung wie für das K-Means Verfahren

Es wird der bereits von Fehlwerten bereinigte Datensatz verwendet, die Schritte hierzu werden ausgiebig im Kapitel Preprocessing besprochen. Zudem werden nur numerische Werte beachtet, Binominale Werte werden Binärwerten konvertiert.

Da es sich bei der K-Means Methode um ein distanzbasiertes Verfahren handelt müssen die numerischen Werte zu Beginn normalisiert werden, da sonst numerisch größere Wertebereiche mehr Einfluss auf die Minimierungsfunktion hätten.

Nach einer statistischen Analyse der Daten zeigt sich, dass die meisten Werte keiner Normalverteilung folgen und keine Nennenswerten Ausreiser vorhanden sind. Aus diesem Grund bietet eine Min-Max Normalisierung hier mehr Vorteile.

Modell

Für das Agglomerative Clustering wird die Bregman Divergence mit dem quadratischen euklidischen Abstand verwendet. Zudem wird das Verfahren mit drei verschiedenen Konfigurationen getestet: **Single Link:** Es werden in jedem Iterationsschritt die beiden Cluster verbunden, dessen Datenpunkte den geringesten Abstand zueinander haben.

Complete Link: Es werden in jedem Iterationsschritt die Cluster verbunden, welche in Verbindung den kleinsten Durchmesser besitzen. **Average Link:** Dies ist ein Kompromiss zwischen Single Link und Complete Link.

Ergebnisse

Das Hierarchische Cluster Verfahren erzeugt ebenso wie das vorherige CLuster Verfahren Ergebnisse die nicht sehr vielversprechend sind. Es werden hier zu viele hierarchische Ebenen generiert, wodurch eine praktische Analyse der Ergebnisse unmöglich ist.

Interessant ist, zu bemerken, dass alle drei Methoden zum selben Ergebniss führen.

Hierarchical Cluster Model

```
Number of clusters :11259  
Number of items :5630
```

Fig. 28 Die Grafik ziegt die gebildeten Cluster

Das zugehörige Dendrogram ist zu unübersichtlich um es mit menschlichen Mitteln zu analysieren. Aus diesem Grund ist das Agglomerative Clustering für dieses Problem ungeeignet und wird auf Grund der sehr guten Ergebnisse des K-Means und K-Medoid Verfahrens nicht weiter berücksichtigt

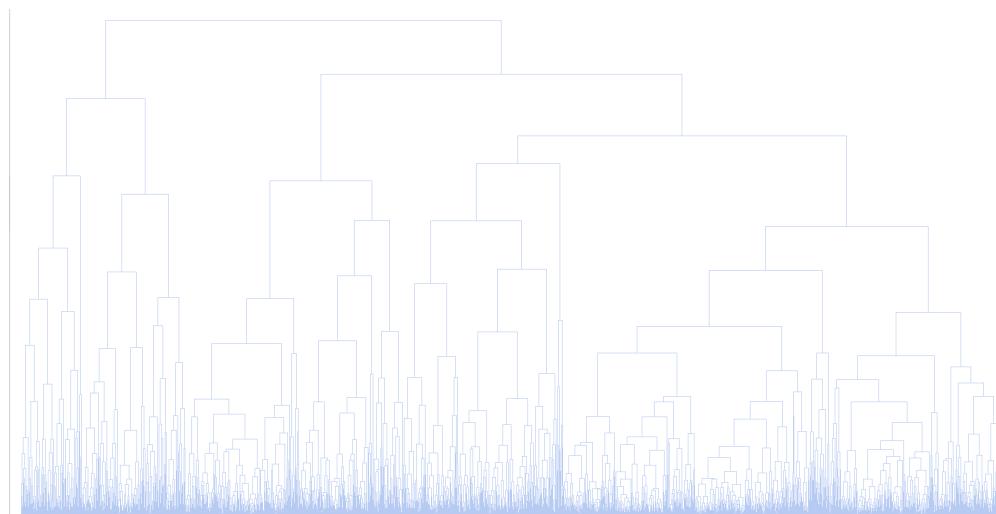


Fig. 29 Die Grafig zeigt das Dendogram

5.2.7 Density-Based Spatial Clustering of Applications with Noise

Der DBSCAN[SSE⁺17] Algorithmus ist ein Dichteverbundenes Clusteringverfahren, erkennt dadurch Gebiete mit hoher Dichte und fasst diese in einem Cluster zusammen. Die Daten werden beim DBSCAN dabei in drei verschiedene Kategorien eingeteilt:

- **Kernpunkt:** Um diesen Punkt wird ein Kreis mit Radius ϵ gespannt.
- **Randpunkt:** Liegt in der Umgebung eines Kernpunktes.
- **Rauschpunkt:** Ist weder Randpunkt noch Kernpunkt.

Der Algorithmus geht wie folgt vor:

1. Datenpunkte werden kategorisiert.
2. Rauschpunkte werden gelöscht.
3. Kernpunkte welche innerhalb des ϵ Radius liegen werden über Kanten miteinander verbunden.
4. Eine Menge Kernpunkte bildet dabei jeweils ein Cluster
5. Jeder Randpunkt wird dem Cluster eines benachbarten Kernpunkts zugewiesen.

Datenvorverarbeitung

Für das Agglomerative Clustering Verfahren erfolgt die selbe Datenvorverarbeitung wie für das K-Means Verfahren

Es wird der bereits von Fehlern bereinigte Datensatz verwendet, die Schritte hierzu werden ausführlich im Kapitel Preprocessing besprochen. Zudem werden nur numerische Werte beachtet, Binomiale Werte werden Binärwerten konvertiert.

Da es sich bei der K-Means Methode um ein distanzbasiertes Verfahren handelt müssen die numerischen Werte zu Beginn normalisiert werden, da sonst numerisch größere Wertebereiche mehr Einfluss auf die Minimierungsfunktion hätten.

Nach einer statistischen Analyse der Daten zeigt sich, dass die meisten Werte keiner Normalverteilung folgen und keine Nennenswerten Ausreiser vorhanden sind. Aus diesem Grund bietet eine Min-Max Normalisierung hier mehr Vorteile.

Ergebnisse

Es ist zu sehen, dass die Cluster des DBSCAN nicht gleichverteilt sind und sogar ein leeres Cluster beinhaltet.

Cluster Model

```
Cluster 0: 0 items
Cluster 1: 188 items
Cluster 2: 320 items
Cluster 3: 280 items
Cluster 4: 160 items
Cluster 5: 2190 items
Cluster 6: 594 items
Cluster 7: 1396 items
Cluster 8: 502 items
Total number of items: 5630
```

Fig. 30 Die Grafik zeigt einen Überblick der gebildeten Cluster

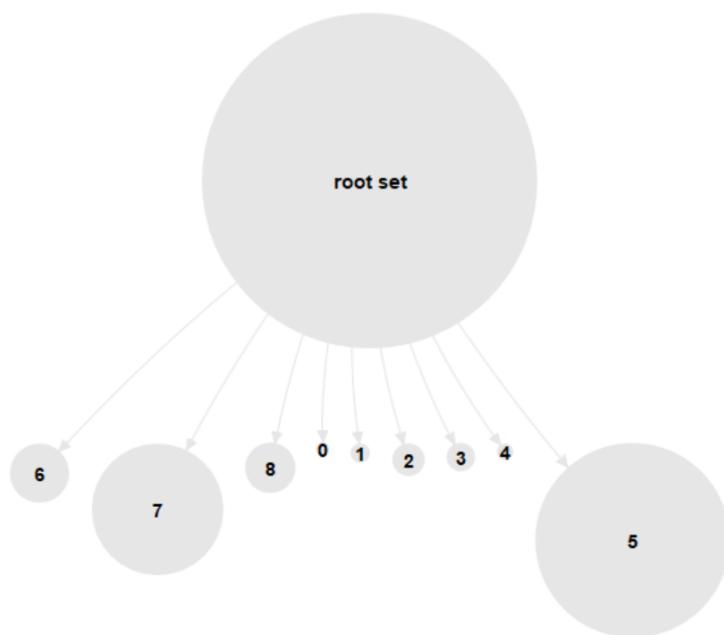


Fig. 31 Die Grafik zeigt die Verteilung der Cluster

Es zeigt sich, dass die wichtigsten Attribute welche einen Datenpunkt einem Cluster zuweisen *Gender*, *Complain* und *Churn* sind.

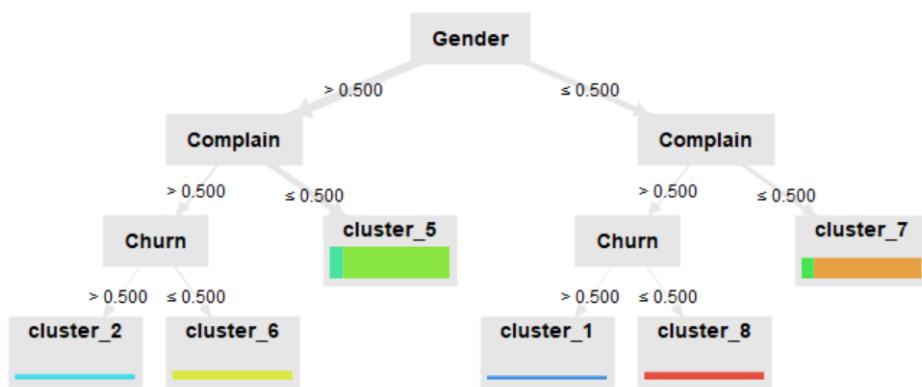


Fig. 32 Die Grafik zeigt die wichtigsten Attribute des Clusters als Baum

5.2.8 Self-organizing Map

Self-organizing Map ist eine computergestützte Datenanalysemethode, die nicht-lineare Datenabbildungen in niedrigere Dimensionen erzeugt. Alternativ kann das SOM als Clustering-Algorithmus betrachtet werden, der einen Satz von Clustern erzeugt, die in einem regelmäßigen Raster organisiert sind

Datenvorverarbeitung

Es wird der bereits von Fehlwerten bereinigte Datensatz verwendet, die Schritte hierzu werden ausgiebig im Kapitel Preprocessing besprochen. Zudem werden nur numerische Werte beachtet, Binominale Werte werden Binärwerten konvertiert.

Da es sich bei der K-Means Methode um ein distanzbasiertes Verfahren handelt müssen die numerischen Werte zu Beginn normalisiert werden, da sonst numerisch größere Wertebereiche mehr Einfluss auf die Minimierungsfunktion hätten.

Nach einer statistischen Analyse der Daten zeigt sich, dass die meisten Werte keiner Normalverteilung folgen und keine Nennenswerten Ausreiser vorhanden sind. Aus diesem Grund bietet eine Min-Max Normalisierung hier mehr Vorteile.

Modell

Für die Implementierung der Self-organizing Map werden die Standardeinstellungen von RapidMiner verwendet. Es werden 1000 Runden trainiert und die Netzgröße wird automatisch bestimmt.

Ergebnisse

Über die P-Matrix lässt sich sehr gut sehen wie die Cluster verteilt sind. Bei näherer Betrachtung der P-Matrix mit den Attributen *Churn*, *Complain* und *Gender* sieht man sehr gut, dass sich hier ein sehr großes Cluster in der rechten oberen Ecke bildet.

Generell bestätigt die Self-organising Map noch einmal die Clusterkonstellationen, welche bereits schon vom K-means und K-Medoids generiert werden.

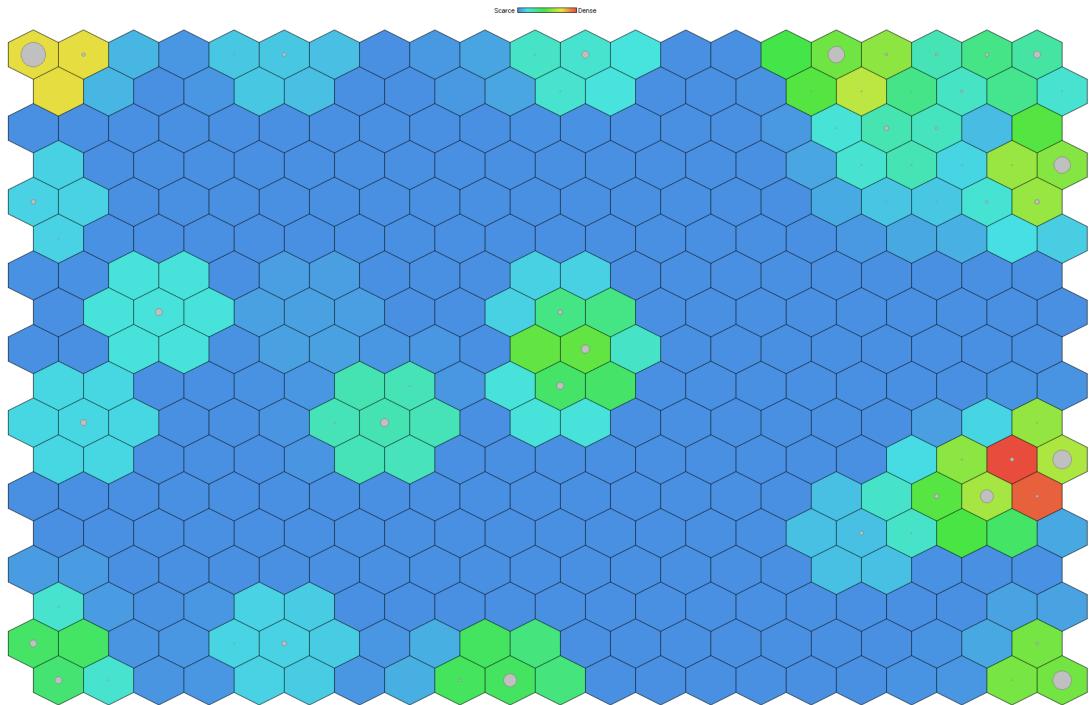


Fig. 33 Die Grafik zeigt die P-Matrix

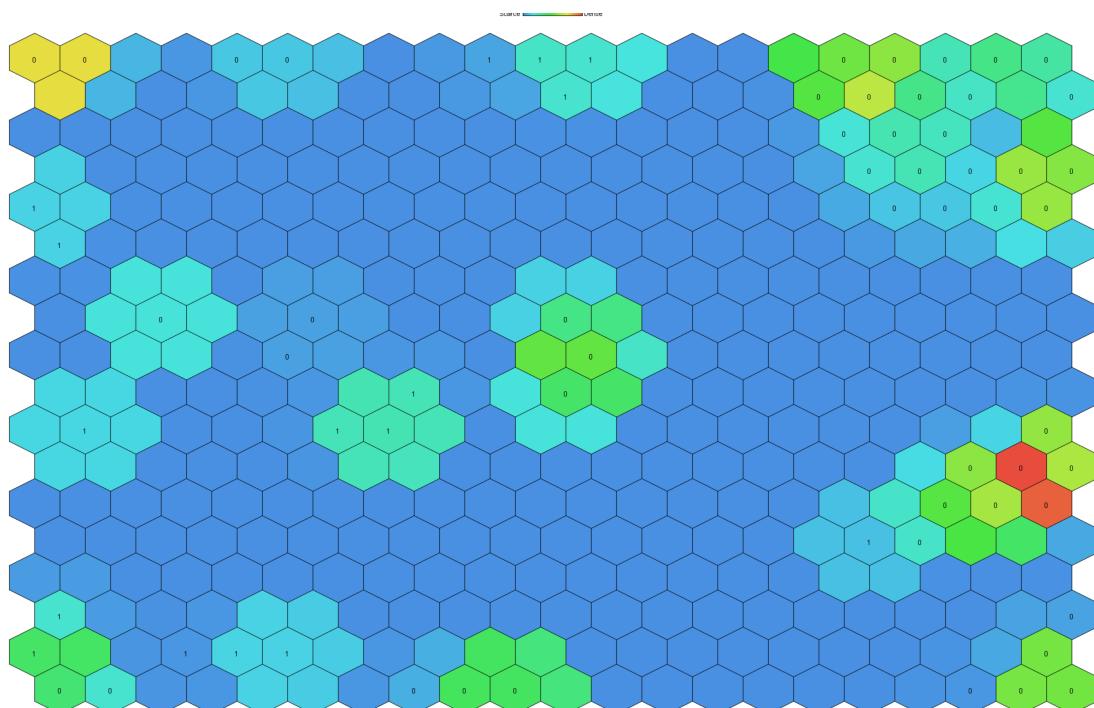


Fig. 34 Die Grafik zeigt die P-Matrix mit zugehörigem Churn

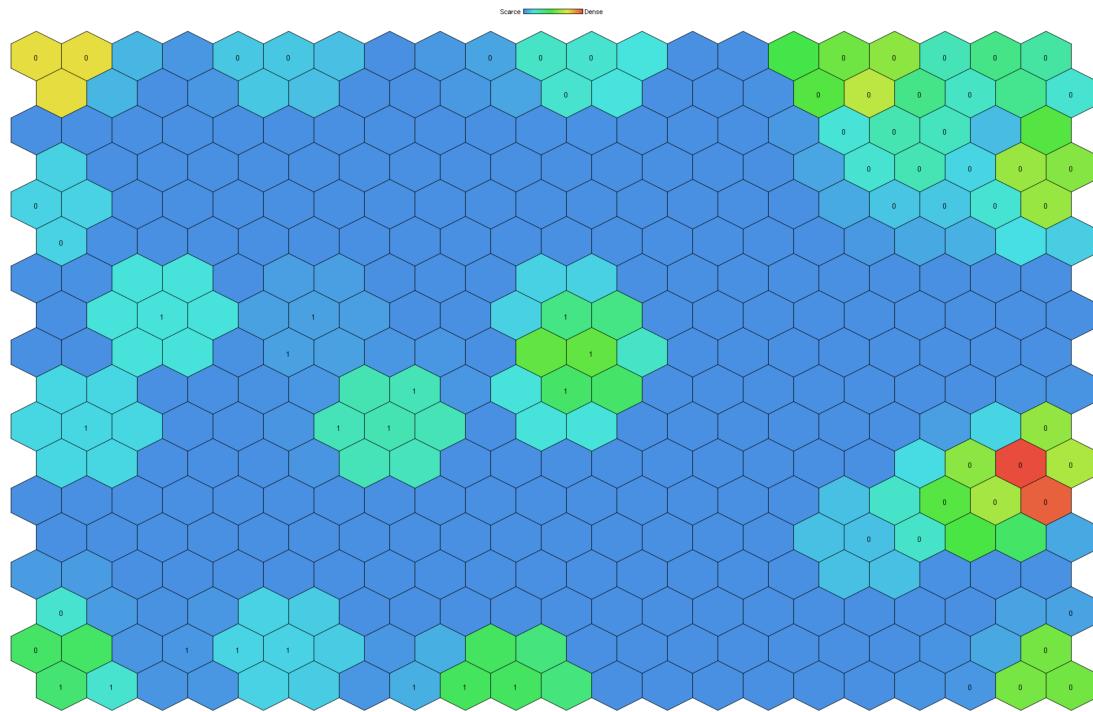


Fig. 35 Die Grafik zeigt die P-Matrix mit zugehörigem Complain

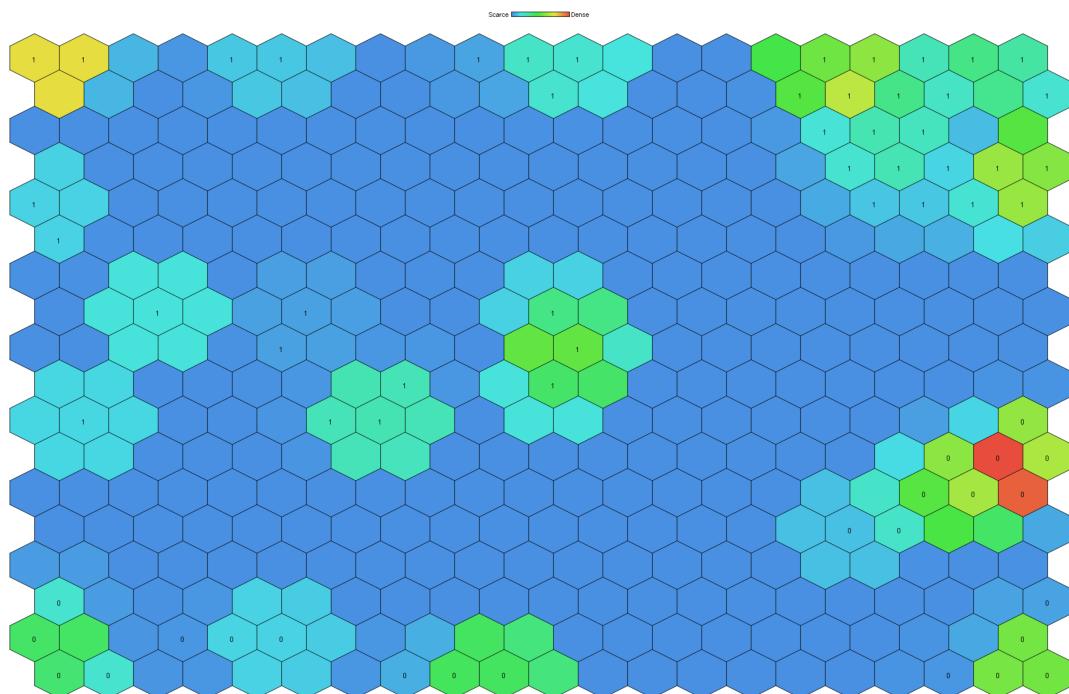


Fig. 36 Die Grafik zeigt die P-Matrix mit zugehörigem Gender

6 Zusammenfassung

Die Analyse mit **Naive Bayes** konnte zwar keine direkten Ergebnisse bezüglich der Klassifizierung von Churn und Complain liefern, die Feststellung, dass die Attribute isoliert betrachtet nicht genug Informationen liefern um eine Klassifizierung zuzulassen ist jedoch trotzdem nicht zu vernachlässigen. So kann man auf der Analyse mit Naive Bayes schlussfolgern, dass die Klassifizierung mit komplexeren Algorithmen, wie Support Vector Maschinen oder Neuronalen Netzen sinnvoll ist.

Die **Support Vector Maschinen** sowie die **Neuronalen Netze**, wurden hauptsächlich dafür verwendet um die Ergebnisse der Klassifizierung für die Zielattribute *Churn*, *Complain* und *CashbackAmount* zu maximieren. Hierbei fällt auf, das um so weniger Attribute verwendet werden, um so schlechter die Klassifizierung oder Schätzungen ausfallen. Da bei der Klassifizierung von *Complain*, unter Beachtung des Kausalmodells, das Attribut Churn nicht verwendet werden darf, fallen die Ergebnisse hier, dementsprechend etwas schlechter aus. Bei den Support Vector Maschinen ist der Grad des Kernels der wichtigste Parameter, bei den Neuronalen Netzen die Anzahl der trainierten Epochen.

Die **Lineare Regression** ergab, dass die präferierte Bestellkategorie einen signifikanten Einfluss auf den CashbackAmount hat. Dabei wurde festgestellt, dass die Bestellkategorien Fashion, Grocery und Others einen positiven Einfluss auf die Zielvariable haben und Laptop & Accessory, Mobile und Mobile Phone einen stark negativen Einfluss.

Die **Entscheidungsbäume** konnten die Attribute Churn und Complain gut Klassifizieren. Alle Erstellten Entscheidungsbäume teilen die Daten zu erst nach Tenure ein. Bei einem höheren Tenuer wandern ein Kunde eher nicht ab und beschweren sich auch weniger. Beschwert sich eine Kunde Wandert er auch häufiger ab. Bei dem Entscheidungsbaum wird auch sichtbar das ein höheres CashbackAmount eher dafür spricht, dass ein Kunde nicht abwandert bzw. sich nicht beschwert. Es gibt einige große Blätter, die zeigen bei welchen Attribut Kombination ein großes Teil der Kunden sich nicht beschweren bzw. nicht Abwandern. Neben den einfachen Entscheidungsbäumen wurde auch ein Random Forest erstellt, dieser erreichte die Besten Klassifikationsergebnis mit einem Kappa von 0.91 für Chrun bzw. einen Kapp von 0.873 für Complain. Mit dem Radom Forst wurde auch eine Regression für CashbackAmount durchgeführt dabei wurde ein RMSE von 8.815 erreicht.

Bei der **Clusteranalyse** heben sich vor allem die K-Means Methode und das K-Medoids Verfahren von den übrigen Algorithmen ab. Insgesamt bringt das K-Medoids Verfahren feinere Cluster hervor, bei denen deutliche Kundengruppen gebildet werden. Im Hinblick auf das K-Medoids Verfahren werden die Kunden-

gruppen kurz benannt:

- Zufriedene Kunden einer eher niedrigen Gesellschaftsschicht.
- Zufriedene Kunden einer höheren Gesellschaftsschicht.
- Konsumorientierte Kunden einer höheren Gesellschaftsschicht.
- Unzufriedene Kunden einer höheren Gesellschaftsschicht.
- Kunden die jeweils nah oder weit entfernt von Umschlagplätzen wohnen.

Bei der **Assoziationsanalyse** konnte mit dem Apriori-Algorithmus einige interessante Zusammenhänge erkannt werden. So ist die Wahrscheinlichkeit sehr hoch, dass wenn eine Person sich in einem bestimmten Intervall des Attributs CashbackAmmount befindet, sich in der PreferredOrderCat Mobile zu befinden. Auch konnte erkannt werden, dass ungewöhnliche Zahlungsmethoden wie E wallets eher in höheren CityTiers bevorzugt werden. Bei der Analyse der Frequent Item Sets konnte erkannt werden, dass die Verwendung eines Mobiltelefons, der Tenure und der CashbackAmmount zusammenhängen. Kunden die mit Mobiltelefonen den analysierten Dienst nutzen haben tendenziell eine hohe Tenure und erhalten viel Cashback. Der Dienst scheint gut für Mobiltelefone optimiert zu sein, da man aus einer hohe Tenure und viel Cashback zumindest folgern kann, dass diese Kunden den Dienst häufig und lange verwenden. Diese Annahmen wurden bei der Analyse der Association Rules noch verstärkt. Falls der CashbackAmmount einen bestimmten Wert überschreitet, ist die Chance, dass der Kunde ein Mobiltelefon verwendet mehr als die Hälfte höher als sonst. Eine weiterer interessanter Zusammenhang der mit Association Rules gefunden werden konnte ist, dass falls die letzte Bestellung eine bestimmte Zeit her ist, bei einer erneuten Bestellung eher Cupons verwendet werden. Das lässt die Vermutung zu, dass Cuponangebote Kunden dazu bewegen erneut zu bestellen. Die Wirkung von Cupons könnte von dem Dienst für künftige Werbeaktionen verwendet werden.

References

- [BMD⁺05] BANERJEE, Arindam ; MERUGU, Srujana ; DHILLON, Inderjit S. ; GHOSH, Joydeep ; LAFFERTY, John: Clustering with Bregman divergences. In: *Journal of machine learning research* 6 (2005), Nr. 10
- [DB79] DAVIES, David L. ; BOULDIN, Donald W.: A Cluster Separation Measure. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1 (1979), Nr. 2, S. 224–227. <http://dx.doi.org/10.1109/TPAMI.1979.4766909>. – DOI 10.1109/TPAMI.1979.4766909
- [DL10] DURGESH, K S. ; LEKHA, B: Data classification using support vector machine. In: *Journal of theoretical and applied information technology* 12 (2010), Nr. 1, S. 1–7
- [GBC16] GOODFELLOW, Ian ; BENGIO, Yoshua ; COURVILLE, Aaron: *Deep Learning*. MIT Press, 2016. – <http://www.deeplearningbook.org>
- [Ger17] GERON, Aurelien: *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA : O'Reilly Media, 2017. – ISBN 978-1491962299
- [Hof06] HOFMANN, Martin: Support vector machines-kernels and the kernel trick. In: *Notes* 26 (2006), Nr. 3, S. 1–16
- [Kal07] KALKA, Jochen: *Zielgruppen: wie sie leben, was sie kaufen, woran sie glauben; Sinus-Milieus von Sinus Sociovision, Semiotmetrie von TNS-Infratest, Zielgruppen-Galaxie von GIM*. MI Wirtschaftsbuch, 2007
- [PF13] PROVOST, Foster ; FAWCETT, Tom: Data science and its relationship to big data and data-driven decision making. In: *Big data* 1 (2013), Nr. 1, S. 51–59
- [SSE⁺17] SCHUBERT, Erich ; SANDER, Jörg ; ESTER, Martin ; KRIEGEL, Hans P. ; XU, Xiaowei: DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. In: *ACM Transactions on Database Systems (TODS)* 42 (2017), Nr. 3, S. 1–21
- [WH00] WIRTH, Rüdiger ; HIPP, Jochen: CRISP-DM: Towards a standard process model for data mining. In: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* Bd. 1 Springer-Verlag London, UK, 2000
- [Wik21] WIKIPEDIA: *Künstliches neuronales Netz — Wikipedia, die freie Enzyklopädie*. https://de.wikipedia.org/w/index.php?title=K%C3%BCnstliches_neuronales_Netz&oldid=17000000

[BCnstliches_neuronales_Netz&oldid=211863046](#). Version: 2021. – [Online; Stand 23. Juli 2021]