CS2364 Grundlagen der KI

WS23/24

Projekt 3:
Apps-Analyse
mit KNIME

Dokumentation

Gruppenmitglieder:
Thaddäus Friedel,
Dick Moritz

Technische Hochschule Mittelhessen
Prof. Dr. F. Kammer

1. Einleitung

Im Rahmen unserer Projektdokumentation zur Analyse von Verkaufsdaten eines Elektrohandels, liegt unser vorrangiges Ziel darin, durch die Auswertung dieser Geschäftsdaten wertvolle Erkenntnisse zu gewinnen, die eine gezielte Optimierung der Verkaufsstrategien ermöglichen. Um diese umfangreichen Daten effizient zu verarbeiten, haben wir das Tool KNIME verwenden.

Die Verbindung von Nodes in Workflows innerhalb von KNIME ermöglicht eine strukturierte Datenverarbeitung. Durch geschickte Verknüpfung der Inputs und Outputs der Nodes wird eine schrittweise Verarbeitung der Daten gewährleistet. Die visuelle Darstellung des Datenflusses in KNIME bietet nicht nur eine übersichtliche Benutzeroberfläche, sondern erleichtert auch das Verständnis des gesamten Prozesses.

Unsere Projektdokumentation wird die methodischen Schritte und Erkenntnisse im Detail beleuchten, um einen umfassenden Einblick in die Datenanalyse für die Verbesserung der Verkaufsstrategien zu gewähren. Dabei setzen wir den Fokus auf die Anwendung des KNIME-Systems zur Analyse der in der Datei pr3.zip bereitgestellten Verkaufsdaten. Mit dieser Dokumentation werden alle Schritte und verwendeten Werkzeuge im Detail erklärt. Für jede Aufgabe des Projekts wird das Konzept, die verwendeten Nodes, welche in der Node Übersicht genauer beschrieben sind, die Implementierung, mindestens eine exemplarische Ausführung, sofern möglich eine Evaluierung der Ergebnisse, sowie die Limits der erarbeiteten Lösung erläutert.

2. Begutachten der Datensätze

Alle für das Projekt relevanten Datensätze befinden sich im beigelegten Archiv “pr3.zip”. Darin enthalten sind, sowohl eine products.csv Datei, mit sämtlichen Informationen zu allen Produkten des Sortiments, sowie zwölf Sales_Monat_2019.csv Dateien, wobei jede dieser Dateien die Verkaufszahlen und Informationen zu einem Monat des Jahres 2019 repräsentiert.

Um die Daten anfänglich zu begutachten, haben wir zunächst jede der Dateien geöffnet, um eventuelle Auffälligkeiten bezüglich Inhaltes und Formatierung, sowie Gemeinsamkeiten, feststellen zu können.

Als erste Auffälligkeiten wären überflüssige Header-Zeilen in allen der Sales Dokumente, sowie uneindeutige Formatierung der Produkt-Datensätze im products.csv Dokument. Diese waren der erste Anhaltspunkt, um ein grobes Konzept für die Vorgehensweise bezüglich Datensanierung und Aufbereitung für die folgende Weiterverarbeitung zu erarbeiten.

3. Installation KNIME & Python

Um dieses Projekt und den dazugehörigen Workflow nutzen zu können, muss eine Installation der KNIME Analytics Platform erfolgen, welche unter diesem Link erhältlich ist (Knime Download). Es gilt zu beachten, dass für den Download keine Registrierung notwendig ist, jedoch muss das Kästchen, für das Akzeptieren der Nutzungsbedingungen, im Registrierungsformular angehakt werden. Für die Installation und auch das Erstellen eines eigenen Workflows, kann folgender Quickstart Guide verwendet werden (Quickstart KNIME).

Für die weiter Verwendung wird eine Installation der Python Extension für KNIME empfohlen. In diesem Projekt-Workflow wurde keine Python Node mit eigenem Code verwendet, es wäre jedoch eine Alternativlösung mittels dieser Node möglich gewesen. Die KNIME eigenen Tools und auch andere Extensions haben für die Verarbeitung dieser Daten jedoch problemlos funktioniert, weshalb hier keine Verwendung von Python Nodes für die Einbindung eigenen Codes von Nöten war. Für eine Weiterverarbeitung der Ergebnisse oder für einen anderen Lösungsansatz, bieten sich besagte Nodes jedoch gut an.

4. Datenjoin und –sanierung

Konzept

Um die gegebenen Datensätze verarbeiten und auswerten zu können, muss die Formatierung und die referenzielle Integrität der beiden Tabellen angepasst werden. Es muss eine Bereinigung aller Datensätze mit fehlenden oder fehlerhaften Werten, sowie das Entfernen von doppelt vorhandenen Einträgen vorgenommen werden, um eine optimale Weiterverarbeitung der Daten zu ermöglichen. Des Weiteren erfolgte die Zerlegung der Adress-Spalte in ihre atomaren Bestandteile um eine mögliche Kategorisierung nach US-Bundesstaat oder ähnlichem vornehmen zu können.

Verwendete Nodes

- CSV Reader
- Duplicate Row Filter
- RowID
- Row Filter
- String to Number
- Rule Engine
- Rule-based Row Filter
- Cell Splitter
- Column Renamer
- Column Aggregator

Implementierung

Zunächst werden zwei verschiedene CSV Reader Nodes erstellt, um jeweils alle Sales Daten und das Dokument mit Produktinformationen einzulesen und in KNIME als eine Tabellendarstellung zu erhalten. Um alle Sales Dateien in eine gemeinsame Tabelle zu überführen, haben wir zunächst einen Ordner erstellt und alle Sales Dateien abgelegt, um dann in den CSV Reader Optionen das Einlesen aller Dateien eines Ordners Nutzen zu können. Die Column Delimiter der CSV Reader müssen auf die jeweilige Formatierung des CSV-Files angepasst werden, somit ist für die Sales Dateien ein “,” und für die die Products Datei “/t” zu verwenden. Für das Einlesen des Product Files konnten wir darüber hinaus die “Has RowID” Funktion verwenden, welche den Primärschlüssel aus den Datensätzen entnimmt, anstatt eine eigenen anzulegen. Selbes Vorgehen war für die Sales Dateien nicht möglich, da aufgrund von Duplikaten kein eindeutiger Primärschlüssel aus den Datensätzen hervorging.

Der Nächste Schritt für die Säuberung der Sales Datensätze, ist demnach das Entfernen genannter Duplikate. Mittels der Duplicate Row Filter Node haben wir alle identischen Datensätze gefiltert und anschließend alle Duplikate entfernt.

Des Weiteren war eine Überarbeitung des Primärschlüssels dieser Tabelle nötig, um die OrderID als unseren Schlüssel festlegen zu können. Dafür wurde die RowID Node verwendet und damit den

bisherigen Primärschlüssel in Form der RowID zu ersetzen. Da es einige Bestellungen mit gleichem Schlüssel gab, welche jedoch unterschiedliche Bestellungen aufgaben, ist eine Fehlerhafte Vergabe der OrderID im Rohdatensatz anzunehmen. Die “Ensure Uniqueness” Option der RowID Node hat jedoch bei jedem dieser doppelten OrderIDs eine eindeutig unterscheidbare Kennung angefügt, z.B. 176560 und 176560(1).

Im nächsten Schritt wurde, mit der Row Filter Node, die übrig gebliebene Header Zeile entfernt, wobei alle anderen bereits bei der Säuberung von Duplikaten behandelt wurden. Wir haben die Funktion “use pattern matching” verwendet und als Pattern “Purchase Address” als String verwendet, da es keine tatsächliche Zeile mit einer gültigen Purchase Address geben kann, die diesen String enthält. Somit wird nur die Header Zeile gefiltert und durch die “Exclude rows by attribute value” Option schließlich entfernt.

Um sicherzustellen, dass die Werte in den Spalten Price Each, Product und Quantity Ordered den richtigen Datentyp besitzen, haben wir die String to Number Node verwendet. Diese Node wandelt in unserem Fall alle Werte in der Price Each Spalte in Zahlen vom Typ Double um und alle Werte in den Spalten Product und Quantity Ordered zu Zahlen vom Typ Integer um. Durch diese Vorgehensweise bereiten wir nicht nur die Zahlenwerte für zukünftige Berechnungen vor, sondern prüfen auch, ob es Einträge gibt, die fälschlicherweise Buchstaben enthalten.

Damit wir nun sichergehen können, dass keine fehlerhaften Datensätze vorhanden sind, verwenden wir Rule Engine und Rule-based Row Filter Nodes, in denen Regeln definiert werden, nach denen der Datensatz gefiltert wird. Zuerst wird überprüft, ob ProductID > 21 ist, da nur 21 verschiedene Produkte im Product.csv definiert sind, bei diesen Einträgen wird die ProductID auf “MISSING” gesetzt. Außerdem wird bei Einträgen, bei denen der Preis nicht zur ProductID passt, die ProductID ebenfalls auf “MISSING” gesetzt. Darauffolgend, werden die “MISSING” ProductIDs, anhand des im Datensatz angegebenen Price Each gesetzt, die fehlende bzw. fehlerhafte ProductID wird also anhand des gegebenen Price Each aufgefüllt. Außerdem wird darauf geachtet, dass nur eindeutig identifizierbare ProductIDs vergeben werden, ist der Price Each beispielsweise auf zwei verschiedene Produkte zutreffend, so kann der Datensatz nicht mit Sicherheit einem der beiden Produkte zugeordnet werden. Anschließend werden noch die Datensätze, bei denen Werte fehlen, oder die Kombination aus ProductID und Price Each nicht stimmt, aussortiert. In unserem Fall wurden hier 7 Datensätze aussortiert, welche nicht rekonstruierbar sind, da bspw. Werte für Adresse oder Quantity Ordered fehlen.

Weiterhin haben wir Cell Splitter und Column Renamer Nodes verwendet, um die Adresse in ihre atomaren Bestandteile zu zerlegen, so wird aus z.B.:

Purchase Address = 917 1st St, Dallas, TX 75001

=> Street = 917 1st St
=> City = Dallas
=> State = TX
=> ZIP => 75001

Die Sales Daten können nun verwendet werden, um weiterführende Analysen und Auswertungen durchzuführen, die einen tieferen Einblick in die Verkaufsperformance und -muster des Elektroartikelhandels ermöglichen.

Um die Produktinformationen der products.csv weiter verarbeiten zu können, war eine Zusammenfassung der Produkt-Kategorien nötig, welche durch uneindeutige Formatierung über vier

Spalten verteilt, aufgelistet waren. Mittels der Column Aggregator Node konnten alle dieser Spalten zu einer komprimiert werden.

Ausführung

Die folgenden Abbildungen geben einen Eindruck darüber, inwiefern der Prozess der Datensäuberung einen Einfluss auf die Datensätze genommen hat. Außerdem sind in Abbildung 5 noch die sieben entfernten Datensätze abgebildet, die nicht weiterverwendet werden konnten.

RowID	Order ID String	Product String	Quantity Ordered String	Price Each String	Order Date String	Purchase Address String
Row18...	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
Row18...	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
Row18...	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
Row18...	236744	24	1	2.99	08/09/19 20:21	11 Wilson St, Atlanta, GA 30301
Row13...	278836	22	1	11.99	11/26/19 20:48	989 12th St, New York City, NY 10001
Row3	176560	21	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
Row4	176561	21	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001

Abbildung 1: Auszug der Sales Datensätze vor Sanierung

RowID	Product Number (i...	Quantity Ordered Number (integer)	Price Each Number (dou...	Order Date String	Street String	City String	State String	ZIP Number (integer)
176558	19	2	11.95	04/19/19 08:46	917 1st St	Dallas	TX	75001
176559	9	1	99.99	04/07/19 22:30	682 Chestnut St	Boston	MA	2215
176560	11	1	600	04/12/19 14:38	669 Spruce St	Los Angeles	CA	90001
176560(1)	21	1	11.99	04/12/19 14:38	669 Spruce St	Los Angeles	CA	90001
176561	21	1	11.99	04/30/19 09:27	333 8th St	Los Angeles	CA	90001
176562	19	1	11.95	04/29/19 13:03	381 Wilson St	San Francisco	CA	94016
176563	9	1	99.99	04/02/19 07:46	668 Center St	Seattle	WA	98101

Abbildung 2: Auszug der Sales Datensätze nach Sanierung

RowID	produkt String	Column1 String	Column2 String	Column3 String	produktkategory String
1	17in_Monitor	?	?	monitor	?
2	20in_Monitor	?	?	monitor	?
3	27in_4K_Gaming_Monitor	?	monitor	?	?
4	27in_FHD_Monitor	?	monitor	?	?
5	34in_Ultrawide_Monitor	?	monitor	?	?

Abbildung 3: Auszug des Products Datensatzes vor Sanierung

RowID	produkt ↓ String	produktkategory String
1	17in_Monitor	monitor
2	20in_Monitor	monitor
3	27in_4K_Gaming_Monitor	monitor
4	27in_FHD_Monitor	monitor
5	34in_Ultrawide_Monitor	monitor

Abbildung 4: Auszug des Products Datensatzes nach Sanierung

RowID	Product Number (integer)	Quantity Ordered Number (integer)	Price Each Number (double)	Order Date String	Purchase Address String
1769...	9	1	99.99	04/06/19 19:33	?
1413...	15	1	14.95	01/08/19 19:27	?
2231...	1	150	?	436 Dogwood St, San Francisco, CA 94016	?
1627...	6	?	2.99	03/08/19 22:09	306 Lakeview St, New York City, NY 10001
2595...	15	1	14.95	10/20/19 21:05	?
2595...	6	1	2.99	10/16/19 19:00	?
2608...	19	1	11.95	10/13/19 17:06	?

Abbildung 5: Entfernte Datensätze

Limits der Lösung

Die Lösung funktioniert mit dem gegebenen Datensatz einwandfrei, allerdings ist das Problem mit der hier erstellten Lösung zur Sanierung und Säuberung der Datensätze mit KNIME, dass sie spezifisch auf den vorliegenden Datensatz zugeschnitten ist. Die Funktionalität des erstellten Workflows ist direkt abhängig von der Struktur, dem Format und den spezifischen Merkmalen, wie z.B. Price Each der einzelnen Produkte, des gegebenen Datensatzes. Daher können Anpassungen und Modifikationen erforderlich sein, um den Workflow erfolgreich auf andere Datensätze anzuwenden, auch wenn diese eine ähnliche Struktur besitzen. Jeder neue Datensatz muss vorher sorgfältig geprüft werden, um sicherzustellen, dass der Prozess genau arbeitet.

5. Kennzahlen

Konzept

Für die Interpretation der gesäuberten Daten, kann man Kennzahlen verwenden, welche sich aus selbst definierten Formeln errechnen lassen. Mit diesen Kennzahlen kann man schließlich Hypothesen zum Kaufverhalten aufstellen, sowie eine Evaluation zu Produkten, sowie deren betriebswirtschaftlichen Merkmalen geben. Wir haben uns im Rahmen des Projekts auf acht verschiedene Kennzahlen geeinigt und diese aus den Daten errechnet. Folgende Kennzahlen für das Jahr 2019 wurden verwendet:

1. Anzahl verkaufter Artikel
2. Umsatz nach Produkt
3. Gesamtumsatz
4. Umsatz pro verkauften Artikel
5. Umsatz pro Kunde
6. Anzahl verkaufter Artikel nach Produkt
7. Umsatz nach Monat
8. Umsatz nach Bundesstaat

Verwendete Nodes

- String to Date&Time
- Date&Time-based Row Filter
- Group By
- Column Renamer
- Row Aggregator
- Joiner
- Math Formula
- Date&Time to String

Implementierung

Zunächst fiel auf, dass sich im Datensatz für den Dezember noch 34 Einträge des 01.01.2020 unter den Daten von 2019 befanden. Um eine aussagekräftige Kennzahl zu ermitteln, müssen diese fälschlich einsortierten Daten entfernt werden, da sonst die Sicht auf das Geschäftsjahr 2019 oder den Monat Dezember im Jahr 2019 beeinträchtigt wird.

Diese Säuberung wurde mit der String to Date&Time Node und der Date&Time-based Row Filter Node durchgeführt. Erstere dient zur Umwandlung, der im String Format gegebenen Einträge der Zeitstempel je Bestellung, in ein tatsächliches Date Format, nachdem schließlich auch nach allen atomaren Bestandteilen eines Zeitstempels gefiltert werden kann. Innerhalb der Date&Time Node haben wir die Order Date Spalte gewählt und die Strings nach dem bisherigen Format eingelesen (MM/dd/yy HH:mm) und schließlich zu einem Date Format ohne die Tageszeit umgeformt (yy-MM-dd), da die, auf die Minute genaue, Zeit der Bestellung nicht relevant für unsere Errechnung der Kennzahlen war.

Folglich war es möglich, die Date&Time-based Row Filter Node zu verwenden, um alle Einträge aus 2020 zu entfernen. Innerhalb diese Node lässt sich ein Zeitraum festlegen, nach dem gefiltert werden soll, welcher sich, in unserem Anwendungsfall, auf den 01.01.2019 bis zum 31.12.2019 belief.

Um sechste Kennzahl, die Anzahl der Verkauften Artikel je Produkt , zu ermitteln, haben wir eine GroupBy Node verwendet, um alle Einträge nach deren respektiven Produkten zusammenzufassen. Zusätzlich wurde hier eine Aggregationsfunktion zur Summenbildung verwendet, um die Summe jeder Gruppe, also jedes Produkts, zu errechnen.

Aus dem gleichen Output besagter GroupBy Node kann man auch unsere erste Kennzahl entnehmen, die Anzahl der verkauften Artikel. Hierfür wurden alle Summen je Produkt noch ein weiteres mal aufsummiert, um eine Gesamtanzahl über alle Verkauften Artikel zu ermitteln. Dieses Ergebnis wurde unter Verwendung einer Row Aggregator Node erzielt, welche mittels Aggregationsfunktion alle Werte der Quantity Sold Spalte summiert und diese dann in einer neuen Spalte ausgibt.

Des Weiteren haben wir den Umsatz pro Produkt gefiltert, was unter Zuhilfenahme der Nodes für Math Formula und GroupBy möglich war. Als Math Formula wurde hier das Produkt aus der Stückzahl der Produkte einer Bestellung und des Preises je Produkt verwendet, um den gesamten Preis pro Bestellung zur erhalten. Mit diesem Wert kann man nun in GroupBy nach Produkt filtern, um herauszufinden, wie viel Umsatz pro Produkt gesamt erzielt wurde.

Darüber hinaus lassen sich die so errechneten Werte mittels einer Row Aggregator Node aufsummieren. Daraus können wir den gesamten Umsatz über alle Produkte für das Geschäftsjahr 2019 ermitteln.

Den Wert aus der vorherigen Berechnung kann man nun mit der Anzahl der gesamt verkauften Artikel mit Hilfe einer Joiner Node verbinden und beide Werte, oder Kennzahlen, in einer Tabelle zusammenführen. Dieser Schritt ist nötig, um eine weitere Kennzahl zu bestimmen, dem Umsatz pro verkauften Artikel. Diese Kennzahl ergibt sich aus der Division des gesamten Jahresumsatzes durch die Anzahl aller verkaufen Artikel, welche man, wie vorher schon verwendet, in einer Math Formula Node durchführen kann.

Um unsere fünfte Kennzahl, den Umsatz je Kunde berechnen zu können, brauchen wir den Gesamtumsatz, welchen bereits vorher ermittelt wurde, sowie die Anzahl der Kunden, die im Jahr 2019 mindestens eine Bestellung aufgegeben haben. Für die Bestimmung der Kundenanzahl war es notwendig eine GroupBy Node zu verwenden und alle Bestellungen, mit jeweiligem gesamten Bestellwert, einer eindeutigen Bestelladresse zuordnen zu können. Eine Bestelladresse wird in diesem Workflow als ein einzigartiger Kunde angesehen, da es keinerlei persönliche Daten zu jedem Kunden gibt, bis auf besagte Adresse. Diese setzt sich aus einer Gruppierung nach Street, City, State und dem jeweiligen ZIP-Code zusammen. Diese Auswertung wird nun mit dem gesamten Jahresumsatz mittels eines Joiners zu einer Tabelle zusammengefügt, welcher einer Math Formula Node als Input dient und uns den Umsatz pro Kunde durch die Division des Gesamtumsatzes durch die Anzahl der einzigartigen Kunden des Jahres 2019.

Für die Berechnung des Umsatzes pro Monat, war eine Date&Time to String Node essentiell, da wir so, mittels einer Formatierung des bisherigen Datums zu einer einzelnen Monatsangabe, extrahieren konnten, in welchem Monat die jeweilige Transaktion stattgefunden hat. Um nun an den Umsatz pro Monat zu kommen, wurde wieder der gesamte Preis für jede Bestellung aufsummiert und dann nach den eben extrahierten Monaten gruppiert. Dies gibt uns die Möglichkeit, den Einzelumsatz jedes Monats zu betrachten.

Des Weiteren bietet sich für diesen Datensatz eine Bestimmung des Umsatzes pro US-Bundestaat an, welcher, ausgehend vom Output der Math Formula Node unserer vorherigen Kennzahl, lediglich die Gruppierung nach Bundesstaat und die Aufsummierung der Bestellpreisen pro Bestellung erfordert.

Ausführung

Die folgenden Abbildungen zeigen die Ergebnisse, bzw. Auszüge der Ergebnisse, der Berechnungen der Kennzahlen.

RowID	Anzahl verkaufter Artikel *Number (integer)*
Row0	208762

Abbildung 6: Ergebnis Kennzahl 1.

RowID	Product *Number (integer)*	Umsatz *Number (double)*
Row0	2	453,488.77
Row1	3	2,432,757.62
Row2	4	1,130,624.62
Row3	5	2,352,898.08
Row4	6	92,621.23

Abbildung 7: Auszug des Ergebnisses von Kennzahl 2.

RowID	Umsatz *Number (double)*
Row0	34,456,563.85

Abbildung 8: Ergebnis Kennzahl 3.

R...	Anzahl verkaufter Artikel *Number (integer)*	Umsatz *Number (double)*	Umsatz pro verkauften Artikel *Number (double)*
Row0...	208762	34,456,563.85	165.052

Abbildung 9: Ergebnis Kennzahl 4.

RowID	Umsatz *Number (double)*	Anzahl Kunden *Number (double)*	Umsatz pro Kunde *Number (double)*
Row0...	34,456,563.85	140,766	244.779

Abbildung 10: Ergebnis Kennzahl 5.

RowID	Product *Number (integer)*	Anzahl verkaufter Artikel *Number (integer)*
Row4	6	30977
Row5	7	27615
Row6	8	15632
Row7	9	13426
Row8	10	4812

Abbildung 11: Auszug des Ergebnisses von Kennzahl 6.

RowID	Monat *String*	Umsatz *Number (double)*
Row0	01	1,812,727.92
Row1	02	2,200,078.08
Row2	03	2,804,964.38
Row3	04	3,389,117.99
Row4	05	3,150,616.23

Abbildung 12: Auszug des Ergebnisses von Kennzahl 7.

RowID	State String	Umsatz Number (double)
Row0	CA	13,699,301.02
Row1	GA	2,794,199.07
Row2	MA	3,657,300.76
Row3	ME	449,321.38
Row4	NY	4,660,502.6

Abbildung 13: Auszug des Ergebnisses von Kennzahl 8.

Evaluierung

Die berechneten Kennzahlen bieten eingehende Einblicke in die Verkaufsperformance des Elektroartikelhandels im Jahr 2019. Die Anzahl verkaufter Artikel, der Gesamtumsatz, der Umsatz nach Produkt und die Anzahl verkaufter Artikel nach Produkt ermöglichen umfassende Analysen der Produktperformance und Nachfrage. Der Gesamtumsatz dient als zentrale Metrik für die finanzielle Gesundheit des Unternehmens, während der Umsatz pro verkauften Artikel eine Feinanalyse der Profitabilität einzelner Produkte ermöglicht. Die Kennzahl Umsatz pro Kunde liefert Erkenntnisse über durchschnittliche Kundenausgaben und unterstützt Strategien zur Maximierung des Kundenwerts. Die Kennzahlen Umsatz nach Monat und Umsatz nach Bundesstaat eröffnen die Möglichkeit, regionale und saisonale Einflüsse auf die Verkaufsleistung zu untersuchen. Diese Analysen bieten wertvolle Erkenntnisse für die Anpassung von Verkaufsstrategien entsprechend den regionalen Gegebenheiten und saisonalen Trends. Insgesamt ermöglichen die Kennzahlen eine fundierte Grundlage für strategische Entscheidungen im Produktmanagement, Marketing und Vertrieb.

Aus den Ergebnissen der Kennzahlen lassen sich einige Erkenntnisse ziehen, beispielweise deutet der besonders hohe Umsatz in Kalifornien (CA) eine starke Nachfrage in dieser Region hin. Zudem zeigt sich, dass der Umsatz im Dezember am höchsten ist, was wahrscheinlich auf Weihnachtseinkäufe und die damit verbundene saisonale Nachfrage zurückzuführen ist. Produkte mit niedrigen Stückzahlen, jedoch hohem Verkaufspreis, können einen erheblichen Umsatzbeitrag leisten. Ein herausragendes Beispiel ist das Macbook Pro, das als umsatzstärkstes Produkt auf einen höheren Verkaufspreis im Vergleich zu anderen Produkten hinweisen könnte. Der hohe durchschnittliche Umsatz pro Kunde (244,78) deutet darauf hin, dass Kunden dazu neigen, mehrere Produkte zu erwerben oder sich für teurere Artikel zu entscheiden. Insbesondere Produkte wie das Macbook Pro und das iPhone, die zu den umsatzstärksten gehören, könnten maßgeblich zum durchschnittlichen Umsatz pro Kunde beitragen. Die Kennzahlen "Verkaufte Artikel" und "Gesamtumsatz" sind allein wenig aussagekräftig, ein Vergleich mit Daten aus Vorjahren wäre hier optimal, um Entwicklungen im Verkaufsverhalten und den finanziellen Erfolg besser zu interpretieren.

Limits der Lösung

Es ist wichtig zu beachten, dass die erstellte Lösung für die Berechnung von Kennzahlen speziell auf die Merkmale und Struktur des vorliegenden Datensatzes zugeschnitten ist. Die verwendeten Formeln und Methoden könnten möglicherweise nicht reibungslos auf andere Datensätze übertragen werden. Eine erfolgreiche Anwendung auf andere Datensätze erfordert daher eine gründliche Überprüfung und möglicherweise Anpassung der Formeln und Methoden.

6. Visualisierung auf Basis von Pivot Tabellen

Konzept

Um den Zusammenhang zwischen den verschiedenen Variablen oder Komponenten in den Datensätzen zu verstehen und angemessen einzuordnen, bietet sich eine grafische Darstellung an. Indem wir die Produktkategorien in Relation zu den Verkaufszahlen, dem Umsatz und der Anzahl der Transaktionen setzen und dabei die Zeit berücksichtigen, können wir wertvolle Erkenntnisse über saisonale Trends oder geografische Besonderheiten gewinnen.

Durch diese Analyse können wir Muster und Zusammenhänge erkennen, die uns dabei helfen, die Dynamik des Marktes besser zu verstehen. Zum Beispiel könnten wir feststellen, dass bestimmte Produktkategorien in bestimmten Jahreszeiten beliebter sind als andere, oder dass bestimmte Regionen einen signifikanten Einfluss auf den Umsatz haben.

Verwendete Nodes

- Joiner
- GroupBy
- Pivot
- Line Plot
- Color Manager
- 3D Scatter Plot (Plotly)

Implementierung

Um die Visualisierung der besprochenen Relationen durchzuführen, bedarf es zunächst einer gründlichen Aufbereitung der relevanten Daten. Zunächst werden alle Bestellungen mit den entsprechenden Gesamtpreisen mittels eines Joiner Nodes mit den aufbereiteten Daten aus der "products.csv"-Datei verknüpft, um die Produktnamen einzubeziehen. Diese Verknüpfung basiert auf der übereinstimmenden Produkt-ID in beiden Tabellen.

Für die weitere Verarbeitung werden GroupBy-Nodes für jede Relation verwendet. Zunächst werden die Daten nach dem Bestelldatum, der Produktkategorie und der Zeit gruppiert, wobei die Anzahl der bestellten Artikel und der Gesamtumsatz summiert werden. Anschließend wird eine Pivot-Node verwendet, um die Daten in eine Form zu bringen, die für ein Liniendiagramm besser geeignet ist.

Das Vorgehen für die anderen Relationen verläuft analog, wobei einmal die Anzahl der Transaktionen benötigt wird.

Nachdem alle Pivot-Tabellen erstellt wurden, kann unter Verwendung der Line Plot Node ein Liniendiagramm für alle Relationen erstellt und anschließend ausgewertet werden. Dies ermöglicht eine klarere Darstellung der Zusammenhänge und Trends in den Daten.

Zusätzlich haben wir den gleichen Vorgang mit Verkaufsdaten je Tag, anstatt pro Monat, wiederholt, sowie mit alternativer Darstellung, indem wir die Einzelprodukte anstelle der Produktkategorien in Verhältnis zu Gesamtumsatz, Anzahl der Transaktionen und auch dem Umsatz nach Bundesstaat gesetzt haben.

In einer weiteren Aufgabe wurde eine Pivot-Tabelle erstellt, die eine Gruppierung nach Bundesstaat, Kategorie und Zeit erfordert, wobei der Umsatz als Linien in unserem Diagramm betrachtet wird. Diese Darstellung funktioniert analog zu den bisherigen Liniendiagrammen und ist deshalb nicht als eigene Sektion gelistet. Die Tabelle wurde nach Nutzung der Pivot-Node in Zeilen nach Monat und in Spalten

nach jeder möglichen Kombination von Staat und Produktkategorie unterteilt. So kann der Umsatz pro Staat und Kategorie über den zeitlichen Verlauf eines Jahres optimal dargestellt werden. Um eine Darstellung über alle drei Dimensionen zugleich zu bekommen, haben wir zusätzlich einen 3D Plot generieren lassen, der jedoch den Datensatz vor Anwendung der Pivot Node verwendet.

Ausführung

Die Abbildungen 14, 15 und 16 zeigen die aus den Line Plots entstandenen Diagramme, welche die Produktkategorien in Relation zu den Verkaufszahlen, dem Umsatz und der Anzahl der Transaktionen nach Monat darstellen. Abbildungen 17 zeigt einen Auszug aus der Pivot-Tabelle, welche den Umsatz nach State und Produktkategorie anzeigt, in Abbildungen 18 und 19 werden diese Daten graphisch dargestellt.

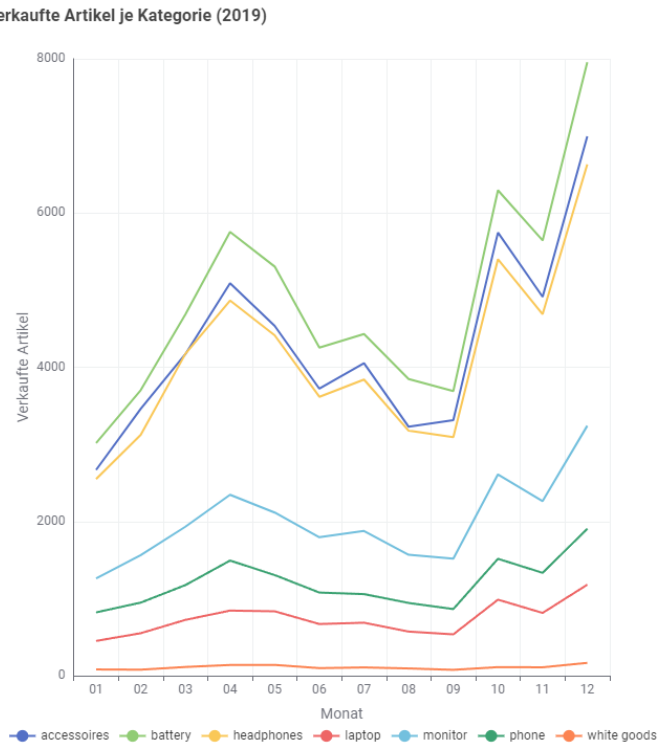

Abbildung 14: Verkaufte Artikel je Kategorie

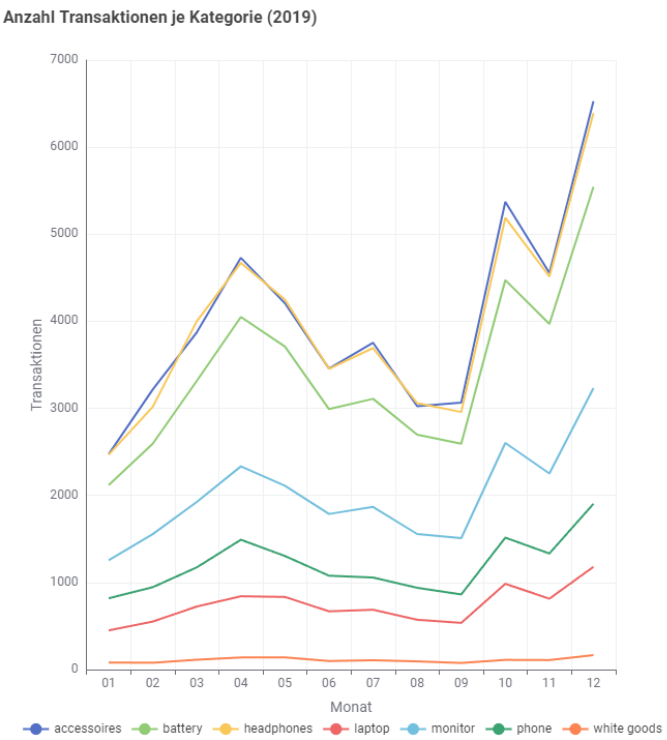

Abbildung 15: Anzahl Transaktionen je Kategorie

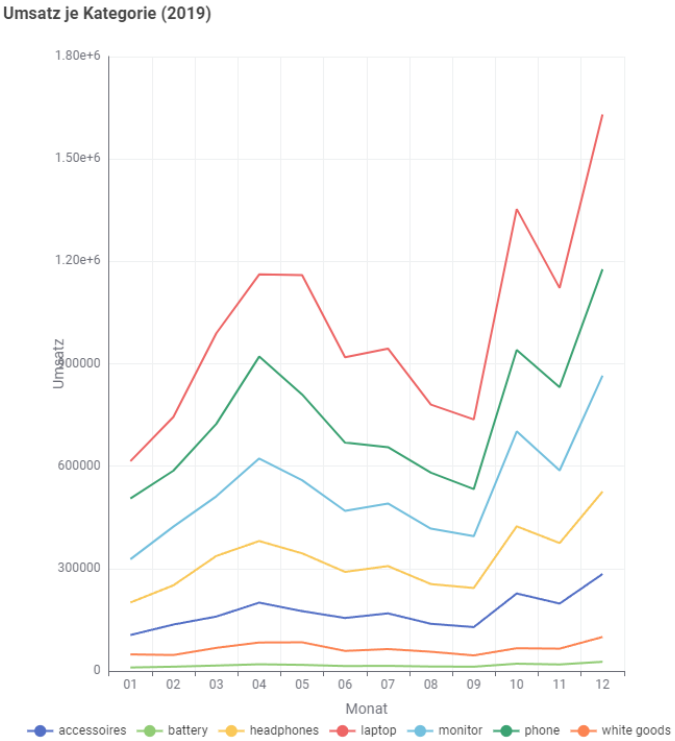

Abbildung 16: Umsatz je Kategorie

RowID	O... St...	CA_a... Number ...	GA_a... Number ...	MA_a... Number (...	ME... Numb...	NY_a... Numbe...	OR_... Numbe...	TX_a... Number ...	WA_ac... Number (...	CA_... Numbe...	GA_... Numb...
Row0	01	38,401.55	9,334.95	12,248.9	2,164.65	13,346.55	6,270.45	15,314.2	8,247.45	4,151.85	787.21
Row1	02	49,925.55	13,334.15	15,990.15	1,650.15	20,240	5,796.85	18,665.7	10,407.05	5,116.32	969.04
Row2	03	63,218.9	14,745.45	20,088.85	994.5	22,042.9	6,476.4	20,319.55	11,538	6,354.78	1,242.21
Row3	04	80,209.05	17,861.65	19,319.05	1,472.6	27,017.85	12,719.3	27,168.65	14,385.25	8,092.94	1,521.01
Row4	05	70,562.45	14,451.05	22,337.4	3,864.85	21,099.85	8,414.8	20,970.05	13,670.65	6,850.85	1,411.26
Row5	06	62,732.4	13,931.35	16,023.85	2,208.4	18,736.3	8,379.05	21,787.2	11,442.7	5,695.81	1,106.02
Row6	07	66,967.05	11,470.3	17,378.9	2,187.5	23,411.1	9,504.5	24,084.3	13,779.65	6,186.93	1,149.14
Row7	08	53,489.85	10,117.2	15,522.65	2,340.95	18,994.6	9,141.65	18,147.45	10,484.8	5,027	1,035.64
Row8	09	54,608.6	10,548.6	13,423.3	1,408.1	13,578.6	6,145.75	17,924.8	10,923.3	4,948.82	1,074.45
Row9	10	89,208	21,865.6	26,281.3	1,442.85	31,040.85	12,038.25	28,311.25	16,827.7	8,345.77	1,596.58
Row10	11	77,046.9	14,975.05	24,038.45	3,408.35	25,598.5	10,368.2	26,013.45	16,109.35	7,670.77	1,401.47
Row11	12	112,140.6	20,029.35	30,558.45	3,214.75	37,620.15	13,411.75	43,946.2	23,165.25	10,692.27	2,177.51

Abbildung 17: Auszug aus der Apriori Tabelle zum Umsatz nach State und Produktkategorie

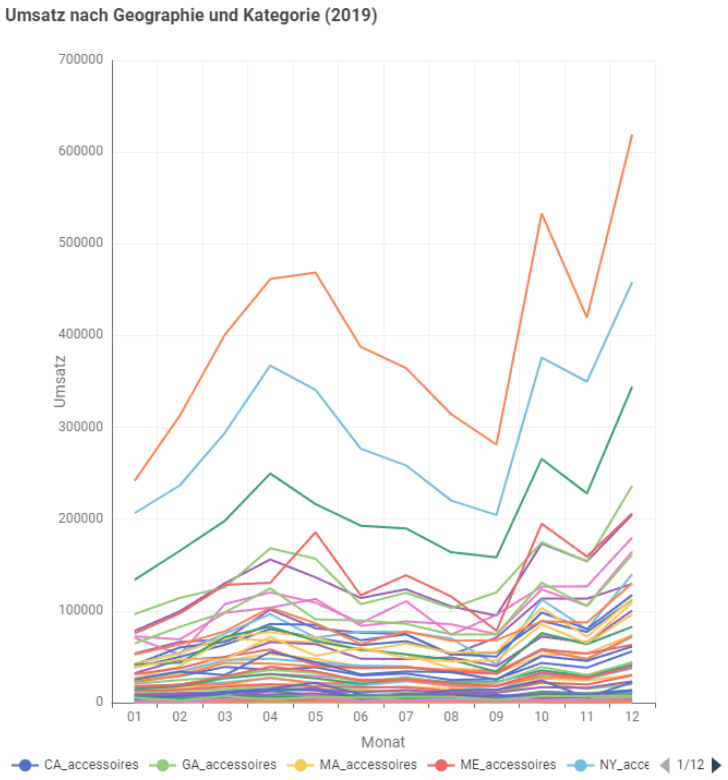

Abbildung 18: 2D-Darstellung zum Umsatz nach State und Produktkategorie

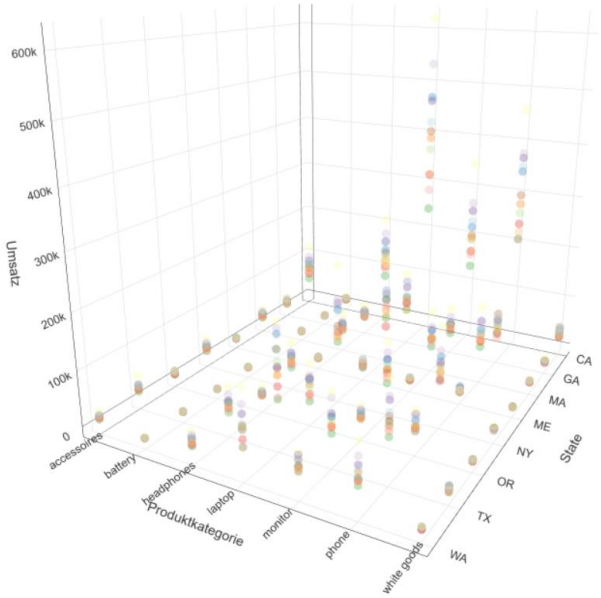

Abbildung 19: 3D-Darstellung zum Umsatz nach State und Produktkategorie

Evaluierung

Die Visualisierungen der verkauften Artikel, Transaktionsanzahl und Umsatz je Kategorie bieten wertvolle Einblicke, die in strategische Entscheidungen im Produktmanagement und Marketing einfließen können. Die Analyse der Anzahl der verkauften Artikel und Transaktionen ermöglicht die Identifikation beliebter Produktkategorien und die Erkennung saisonaler oder trendbedingter Muster, während der Umsatz je Kategorie Einblicke in die Profitabilität bietet.

Dabei hebt sich deutlich heraus, dass Batterien, Accessoires und Kopfhörer zu den meistverkauften Produktkategorien gehören, während 'white goods' (Waschmaschinen & Trockner) mit Abstand am wenigsten verkauft wurden. Die Produktkategorien mit den meisten Transaktionen sind Accessoires, Kopfhörer und Batterien. Im Gegensatz dazu verzeichnen 'white goods' die mit Abstand geringste Anzahl von Transaktionen zu anderen Kategorien. In Bezug auf den Umsatz zeigen die Visualisierungen, dass Laptops, Smartphones und Monitore die umsatzstärksten Produktkategorien sind. Auffällig ist, dass Batterien und 'white goods' den geringsten Umsatz generieren. Diese Erkenntnisse ermöglichen gezielte strategische Maßnahmen, um den Fokus auf profitablere Kategorien zu legen und das Marketing entsprechend anzupassen.

Die Visualisierungen, welche die Produkte, statt der Produktkategorien in Relation zu den Verkaufszahlen, dem Umsatz und der Anzahl der Transaktionen nach Monat darstellen, könnten verwendet werden, um eine noch detailliertere Unterscheidung, bzw. Analyse vorzunehmen, indem man einzelne Produkte, statt nur deren Kategorie, miteinander vergleicht. Die Visualisierung pro Tag statt pro Monat könnte auch verwendet werden, um Trends noch genauer zu erfassen und diese zu nutzen, indem man das Marketing entsprechend anpasst.

Aus den Daten der Pivot-Tabelle, welche zusätzlich die geographische Lage miteinbezieht, lässt sich schließen, dass preishöhere Produkte besonders hohen Umsatz in CA erzielen, während in ME auffällig wenig Umsatz generiert wird, diese Informationen können verwendet werden, um regionale Verkaufsstrategien zu optimieren und gezielt auf die unterschiedlichen Bedürfnisse und Präferenzen

der Kunden in verschiedenen Bundesstaaten einzugehen. Die Pivot-Tabelle, die die geographische Lage berücksichtigt, ermöglicht es, regionale Unterschiede im Umsatz genauer zu analysieren.

Limit der Lösung

Die Übersichtlichkeit der Darstellung in Liniendiagrammen hängt stark von der Anzahl der darzustellenden Elemente ab. Bei einem Sortiment mit mehr Kategorien, könnte es schwierig sein, Trends und Ausnahmen zu erkennen, es sei denn, diese heben sich deutlich von den übrigen Linien ab. Dies liegt daran, dass der begrenzte Platz im Diagramm die Unterscheidung und Analyse vieler Linien erschwert.

Des Weiteren ermöglicht ein Liniendiagramm maximal die Darstellung von zwei Dimensionen, was bedeutet, dass eventuelle Wechselwirkungen mit einer möglichen dritten Dimension nicht berücksichtigt werden können. Dadurch könnten wichtige Zusammenhänge und Einflüsse übersehen werden, insbesondere wenn mehrere Faktoren gleichzeitig betrachtet werden müssen.

Um diese Einschränkungen zu überwinden, könnten alternative Visualisierungsmethoden wie Cluster oder 3D Diagramme sämtlicher Arten verwendet werden. Diese erlauben es, mehrere Dimensionen gleichzeitig zu betrachten und ermöglichen eine bessere Erfassung komplexer Zusammenhänge in den Daten.

7. Apriori Algorithmus

Konzept

Um die Verkaufsstrategie eines Unternehmens zu verbessern, ist es oft hilfreich, zu wissen, welche Produkte häufig gemeinsam gekauft werden oder welche Produkte mit einer höheren Wahrscheinlichkeit gekauft werden, wenn sich bereits ein anderes Produkt im Warenkorb befindet. Für eine solche Analyse bietet sich die Anwendung des Apriori-Algorithmus an.

Der Apriori-Algorithmus analysiert Bestellungen, die zuvor zu einer Liste formatiert wurden und alle Bestellungen zum gleichen Zeitpunkt von derselben Person zu einem Warenkorb zusammenfassen. Auf Basis dieser Daten liefert der Algorithmus Assoziationsregeln.

Der Support-Wert gibt an, wie oft ein bestimmtes Produkt oder eine Produktkombination in den Transaktionen vorkommt, während die Confidence die Wahrscheinlichkeit misst, dass ein Produkt B gekauft wird, wenn Produkt A bereits im Warenkorb ist. Diese Werte bieten Unternehmen wichtige Erkenntnisse darüber, welche Produkte gemeinsam verkauft werden und wie stark die Beziehung zwischen verschiedenen Produkten ist.

Durch die Analyse der Assoziationsregeln können Unternehmen gezieltere Marketing- und Cross-Selling-Strategien entwickeln. Zum Beispiel können sie Produkte, die oft gemeinsam gekauft werden, gemeinsam bewerben oder spezielle Angebote für bestimmte Produktkombinationen anbieten.

Verwendete Nodes

- Number to String

- Joiner
- GroupBy
- Association Rule Learner (Borgelt)
- Row Filter
- Column Filter
- Scatter Plot

Implementierung

Um die Daten in der Apriori Node verarbeiten zu können, müssen zuerst alle Werte in der Product-Spalte, die bisher als Integer repräsentiert waren, in einen String umgewandelt werden. Dies wird mit der Number-to-String-Node durchgeführt, da die Apriori Node nur mit String-Input arbeiten kann.

Der nächste Schritt besteht darin, die Sales-Daten mit den Produktinformationen in einer Joiner-Node zusammenzuführen, um für alle Bestellungen auch den Namen des jeweiligen Produkts in die Tabelle einzubringen. Dabei werden die Namen zusätzlich zu den IDs benötigt, um eine bessere Übersicht zu erhalten und die später erstellten Warenkörbe nicht nur anhand der ID entschlüsseln zu müssen.

Daraufhin erfolgt mit einer GroupBy-Node eine Gruppierung nach Bestellungen, wobei alle Einträge mit derselben Adresse und dem exakt gleichen Zeitpunkt zu einer Bestellung zusammengefasst werden. Durch eine Aggregationsfunktion entsteht eine Liste, die den Warenkorb der jeweiligen Bestellung darstellt.

Mithilfe dieser Listen, die alle Artikel der Bestellung enthalten, kann die Apriori Node oder, wie in unserem Workflow genannt, der Association Rule Learner (Bogelt), gestartet werden. Als Optionen müssen die Liste der Produkte als Input ausgewählt werden sowie ein Wert für den minimalen Support festgelegt werden. Da wir eine hohe Zahl an Einzelbestellungen haben, wird der Support-Wert auf 0,0 gesetzt, um die relativ geringe Anzahl an Bestellungen mit mehr als einem Produkt zu berücksichtigen. Die "Minimum set size" wird auf zwei festgelegt, um Einzelbestellungen zu filtern, und die "Maximale Set Size" muss mindestens so hoch sein wie das größte Set. Ein Wert von zehn ist dabei mehr als ausreichend. Der letzte relevante Wert ist die "Minimum rule confidence", für die ebenfalls 0,0 gewählt wird, um alle Regeln anzeigen zu können und sie anschließend je nach Output der Apriori Node gegebenenfalls noch filtern zu können.

Um die interessanten Regeln zu filtern, wird anschließend ein Row Filter eingesetzt, der alle Regeln mit einem Lift über dem Wert zehn herausfiltert, um unrealistische Regeln zu entfernen. Wenn der Lift-Wert auffällig hoch ist, wurde meist eine bestimmte Kombination aus Produkten nur ein einziges Mal erworben, was jedoch für unsere Evaluierungen nicht relevant ist.

Schließlich wird eine Column Filter Node verwendet, um die Tabellendarstellung auf die wichtigen Metriken zu beschränken.

Weiterführend sollte der Zusammenhang zwischen Preis und bestellter Menge der einzelnen Produkte analysiert werden. Dafür wird die Tabelle des Joiners für das Apriori-Modell verwendet und nach ProductID und Preis je Produkt gruppiert. Die Ergebnisse werden dann in einem Scatter Plot dargestellt, um die bestellte Menge in Relation zum Preis zu setzen.

Ausführung

Die folgenden Abbildungen 20, 21 und 22 stellen Auszüge der erstellten Assoziationsregeln, mit verschiedenen Eingabe- und Filterwerten, dar. Abbildung 23 präsentiert einen Scatter Plot, der die Beziehung zwischen dem Preis eines Produkts und der Anzahl der getätigten Bestellungen darstellt.

RowID	Consequent *String*	Antecedent *List*	RelativeItemSetSupport% *Number (double)*	RuleConfidence% *Number (double)*	RuleLift *Number (double)*
Row...	Lightning_Charging_Cable	[iPhone]	0.566	14.8	1.22
Row...	iPhone	[Lightning_Charging_Cable]	0.566	4.68	1.22
Row...	USB-C_Charging_Cable	[Google_Phone]	0.559	18.1	1.474
Row...	Google_Phone	[USB-C_Charging_Cable]	0.559	4.56	1.474
Row...	Wired_Headphones	[iPhone]	0.259	6.75	0.639

Abbildung 20: Auszug der Assoziationsregeln, nach Support absteigend sortiert

RowID	Consequent *String*	Antecedent *List*	Relativelte... *Number (double)*	RuleConfidence% *Number (double)*	RuleLift *Number (double)*
Row...	Google_Phone	[USB-C_Charging_Cable,Wired_Headphones]	0.049	42.9	13.848
Row...	iPhone	[Lightning_Charging_Cable,Wired_Headphones]	0.035	48.8	12.74
Row...	iPhone	[Apple_Airpods_Headphones,Lightning_Charging_Cable]	0.026	40.5	10.569
Row...	Google_Phone	[Bose_SoundSport_Headphones,USB-C_Charging_Cable]	0.02	34.3	11.088
Row...	Vareebadd_Phone	[USB-C_Charging_Cable,Wired_Headphones]	0.018	16.3	14.047

Abbildung 21: Auszug der Assoziationsregeln, gefiltert mit Lift > 10, nach Support absteigend sortiert

RowID	Consequent *String*	Antecedent *List*	RelativeItemSetSupport% *Number (double)*	RuleConfidence% *Number (double)*	RuleLift *Number (double)*
Row...	USB-C_Charging_Cable	[Google_Phone]	0.559	18.1	1.474
Row...	USB-C_Charging_Cable	[Vareebadd_Phone]	0.206	17.8	1.455
Row...	Lightning_Charging_Cable	[iPhone]	0.566	14.8	1.22
Row...	Wired_Headphones	[Google_Phone]	0.237	7.64	0.724
Row...	Wired_Headphones	[Vareebadd_Phone]	0.084	7.22	0.683

Abbildung 22: Auszug der Assoziationsregeln, gefiltert mit Support > 0.05, nach Confidence absteigend sortiert

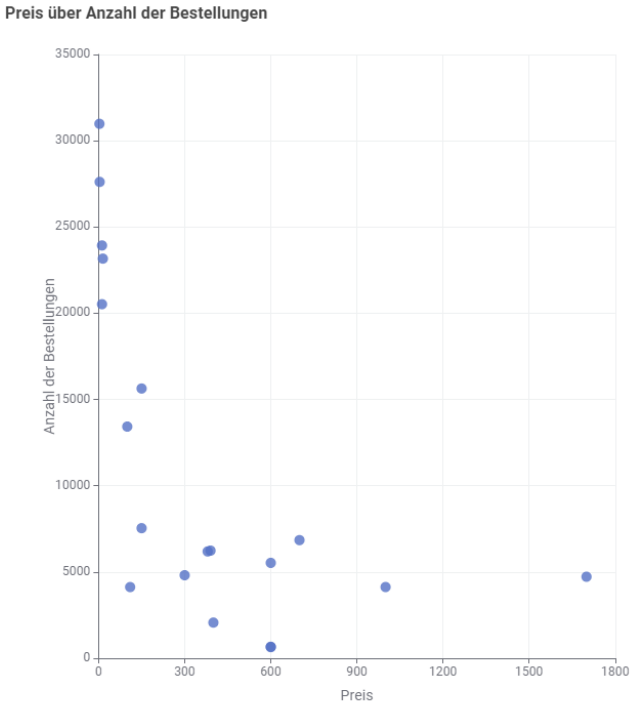

Abbildung 23: Diagramm mit dem Preis der Produkte über die Anzahl der Bestellungen

Evaluierung

Die Evaluation der aus dem Apriori-Algorithmus abgeleiteten Assoziationsregeln offenbart einige Erkenntnisse über das Kaufverhalten der Kunden. Besonders im Fokus stehen hierbei die Verknüpfungen zwischen Smartphones, Ladekabeln und Kopfhörern. Die erlangten Erkenntnisse ermöglichen nicht nur ein besseres Verständnis des Kundenverhaltens, sondern eröffnen auch die Möglichkeit für gezielte und effektive Marketingstrategien.

Smartphones gehen oft Hand in Hand mit Ladekabeln: Wenn ein Kunde ein USB-C-Ladekabel im Warenkorb hat, steigt die Wahrscheinlichkeit, dass er auch ein Google Phone kauft, um etwa 1,5-mal. Bei einem Lightning-Ladekabel ist Wahrscheinlichkeit für den Kauf eines iPhones fast 5%. Zusätzlich zeigt sich, dass Kunden, die bereits Ladekabel und Kopfhörer erworben haben, mit hoher Wahrscheinlichkeit auch ein Smartphone kaufen – der Anstieg liegt bei über 10-fach.

Die gewonnenen Erkenntnisse durch die Assoziationsregeln bieten tiefgehende Einblicke in das Kundenverhalten und Ermöglichen die Entwicklung präziserer Marketing- und Cross-Selling-Strategien. Durch die gezielte Bewerbung von Produktbündeln, wie beispielsweise Smartphones, Ladekabeln und Kopfhörern, oder das Anbieten von Sonderangeboten für bestimmte Produktkombinationen, lässt sich die Wahrscheinlichkeit für zusätzliche Käufe erhöhen. Diese strategischen Maßnahmen tragen dazu bei, die Umsätze zu steigern, die Kundenzufriedenheit zu verbessern und schlussendlich die Rentabilität des Unternehmens zu erhöhen.

Das Diagramm, welches den Preis der Produkte mit der Anzahl der Bestellungen in Beziehung setzt, zeigt, dass kostengünstige Produkt deutlich öfter als teure Produkte bestellt werden.

Limit der Lösung

Die Assoziationsanalyse, insbesondere unter Verwendung des Apriori-Algorithmus, weist gewisse Limitationen auf, die bei der Interpretation der Ergebnisse berücksichtigt werden sollten. Ein wesentlicher Punkt ist, dass in den vorliegenden Kaufdaten rund 80 Prozent der Bestellungen lediglich ein Produkt beinhalten. Dieser Umstand kann die Effektivität der Apriori-Analyse beeinträchtigen, da

der Algorithmus darauf ausgelegt ist, Zusammenhänge zwischen mehreren Produkten zu identifizieren. Bei einem Großteil der Bestellungen, die nur ein Produkt umfassen, können die Regeln weniger aussagekräftig sein und die Fähigkeit des Algorithmus zur Generierung signifikanter Assoziationen einschränken. Es ist daher wichtig zu berücksichtigen, dass die Anwendbarkeit der Assoziationsanalyse von der Datenstruktur und der Vielfalt der in den Transaktionen enthaltenen Produkte abhängt.

8. Clustern der Kunden

Konzept

Clustering ermöglicht es, Ähnlichkeiten in Daten zu identifizieren und sie in Gruppen oder Clustern darzustellen, basierend auf den gemeinsamen Merkmalen oder Eigenschaften. Dies bietet besondere Vorteile, bei großen Datensätzen, bei welchen die Zusammenhänge nicht direkt erkennbar sind.

Häufig finden solche Methoden Anwendung bei der Analyse von Kaufverhalten von Kunden und zur Identifikation von Kunden-Gruppen oder –Segmenten. Kunden werden somit in verschieden Gruppen geteilt und es können effizientere Marketingstrategien angewendet werden, da man leichter Bedürfnisse und Vorlieben der ganzen Gruppe erreichen kann.

Für dreidimensionale Daten bieten Clustering-Algorithmen eine effektive Möglichkeit, komplexe Strukturen zu analysieren und zu verstehen. Durch die Visualisierung der Daten in einem dreidimensionalen Raum können Cluster identifiziert werden, die in bestimmten Bereichen des Raums konzentriert sind, was auf gemeinsame Merkmale oder Beziehungen zwischen den Datenpunkten hinweisen kann.

Darüber hinaus können Clustering-Algorithmen auch in zweidimensionalen Darstellungen von Daten verwendet werden, um Muster zu erkennen und Zusammenhänge zu identifizieren. Obwohl die Dimension der Daten reduziert wird, können Clustering-Techniken dennoch wertvolle Einblicke bieten und bei der Entdeckung von Strukturen oder Gruppierungen in den Daten helfen.

Verwendete Nodes

- Math Formula
- GroupBy
- Column Aggregator
- Category to Number
- RowID
- Column Filter
- k-Means
- Color Manager
- Scatter Plot

Implementierung

Bevor ein Cluster erstellt werden kann, müssen zunächst die relevanten Werte beschafft oder berechnet werden. Diese Berechnung findet wieder in einer Math Formula Node statt. In dieser wurde, wie bereits zuvor verwendet, der Gesamtpreis pro Bestellung ausgerechnet und dann an eine GroupBy Node weitergegeben.

Da es um ein Clustering zum Kaufverhalten der Kunden geht, muss nach Kunden gruppiert werden, die über die einzigartigen Bestelladressen ausfindig gemacht werden. Es werden auch alle Preise der Einzelbestellungen aufsummiert, um zu sehen, wie viel ein Kunde insgesamt ausgegeben hat.

Zusätzlich wird dasselbe Prinzip bei der Anzahl der bestellten Artikel angewendet, und darüber hinaus wird gezählt, wie viele Bestellungen es pro Kunde gab.

Um den k-Means-Algorithmus verwenden zu können, bedarf es noch einiger Umformungen, um eine passende Formatierung zu schaffen. Zuerst haben wir die Adressen der Kunden mit einer Column Aggregator Node zusammengefügt und dann, mit der Category to Number Node, die verbliebenen Spalten mit String-Werten in repräsentative Zahlen umgeformt, da die k-Means-Node ausschließlich mit Zahlen arbeiten kann.

Schließlich haben wir die Row ID mit Hilfe der RowID Node angepasst und danach die Adresse gefiltert, da diese nur noch in Form einer eindeutigen ID wichtig war.

Die k-Means-Node wurde mit 3 Clustern konfiguriert, die unserer Ansicht nach eine optimale Darstellung für das Kaufverhalten der Kunden wiedergaben.

Schließlich wurde noch ein Color Manager verwendet, um den Clustern eine geeignete farbliche Erscheinung zu geben, und ein Scatter Plot für die endgültige Visualisierung erzeugt. Wir haben uns für die Anzahl der Bestellungen pro Kunde als horizontale Achse entschieden, sowie die korrespondierenden Gesamtausgaben als vertikale Achse.

Ausführung

Die folgenden Abbildungen zeigen 2D- und 3D-Scatter Plots, welche die geclusterten Daten nach unterschiedlichen

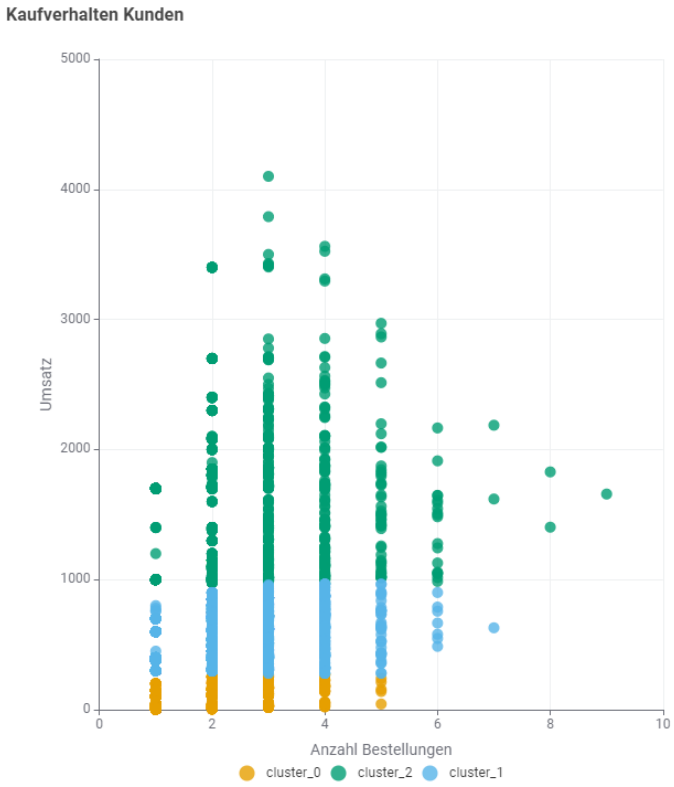

Abbildung 24: 2D-Diagramm, welches den geclusterten Datensatz nach Anzahl der Bestellungen und Umsatz darstellt

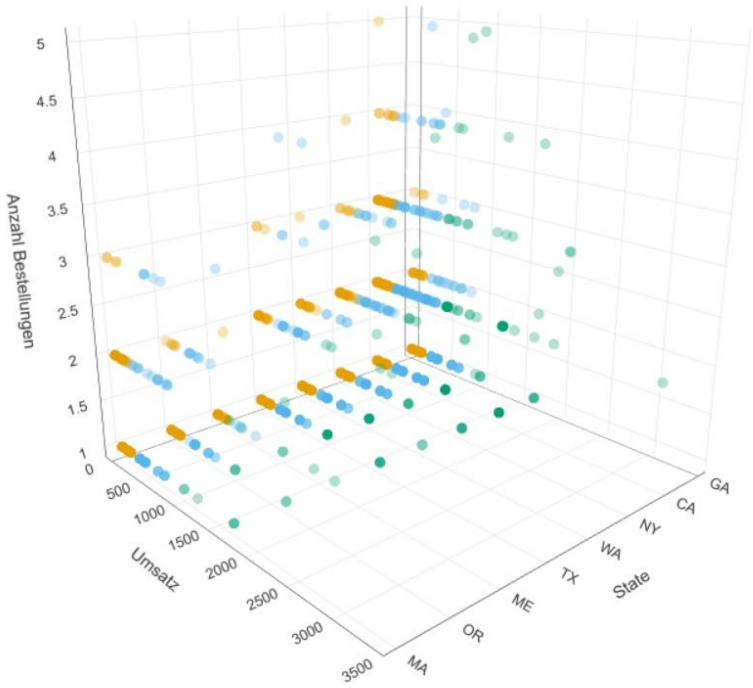

Abbildung 25: 3D-Diagramm, welches den geclusterten Datensatz nach Anzahl der Bestellungen, Umsatz und Bundestaat darstellt

Evaluierung

Die Anwendung des KMeans-Algorithmus für das Kunden-Clustering hat zu einer Segmentierung in drei Gruppen geführt: Orange (0), Blau (1) und Grün (2). Jede dieser Gruppen repräsentiert unterschiedliche Verhaltensmuster und Kaufgewohnheiten.

Im Orangenen Cluster finden sich Kunden mit wenigen Bestellungen, in der Regel zwischen 1 und 3, und geringen Gesamtausgaben, die häufig unter 300 liegen. Diese Gruppe könnte potenziell auf Angebote für Erstkäufe oder gezielte Werbemaßnahmen für kleine Transaktionen reagieren.

Der Blaue Cluster umfasst Kunden mit mittleren Bestellmengen, meistens zwischen 2 und 4, sowie mittleren Gesamtausgaben im Bereich von 300 bis 1000. Dies legt nahe, dass diese Kunden eher auf eine breitere Palette von Produkten zugreifen und möglicherweise auf Cross-Selling-Angebote ansprechen könnten.

Im Grünen Cluster befinden sich Kunden mit mehreren und teureren Bestellungen, die über 1000 liegen. Diese Gruppe könnte für gezielte Marketingstrategien für hochpreisige Artikel interessant sein oder von exklusiven Angeboten profitieren.

Die Unterscheidung zwischen den Clustern ermöglicht es dem Unternehmen, gezielte Marketingstrategien und personalisierte Angebote für jede Kundengruppe zu entwickeln. Dies trägt nicht nur zur Verbesserung der Kundenzufriedenheit bei, sondern auch zur Steigerung der Umsätze und der Gesamteffizienz des Marketingansatzes.

Die Ergänzung der geographischen Dimension in Abbildung 25 liefert zusätzliche Einsichten in die Kundencluster. Auffällig ist, dass die überwiegende Mehrheit der Kunden des Grünen Clusters aus Kalifornien stammt. Im Gegensatz dazu zeigt sich, dass in Staaten wie Maine nur wenige Kunden dem

Grünen Cluster angehören. Dies unterstreicht die Bedeutung regionaler Unterschiede im Kundenverhalten und ermöglicht es dem Unternehmen, ihre Marketingstrategien noch präziser an die spezifischen Anforderungen unterschiedlicher geografischer Märkte anzupassen.

Limit der Lösung

Das Hauptproblem bestand darin, dass die Variablen im Datensatz zu stark miteinander korreliert waren. Dies führte dazu, dass die Daten bereits vor dem Clustering in gewisser Weise vorsortiert waren oder weniger Unterschiede zwischen den Datenpunkten aufwiesen. In solchen Fällen kann es schwieriger sein, klare Cluster zu identifizieren, da die Unterschiede zwischen den Gruppen weniger deutlich sind. Dies könnte zu weniger aussagekräftigen Clustern führen oder dazu, dass der Algorithmus Schwierigkeiten hat, die Daten in sinnvolle Gruppen zu unterteilen. Zusätzlich waren die meisten Metriken diskret skaliert, wodurch sich eine Clusterbildung nochmals erschwert, sofern man eine geringe Anzahl an Merkmalsausprägungen hat. Somit war es zwar möglich ein Clustering anzuwenden, jedoch wäre diese Methodik für einen anderen Datensatz wahrscheinlich besser geeignet.

9. Fazit

Das Projekt war insgesamt erfolgreich und hat wichtige neue Erkenntnisse über das Kaufverhalten der Kunden geliefert. Die Verwendung von KNIME erwies sich als äußerst effektiv, da die Plattform eine benutzerfreundliche Umgebung bietet, um komplexe Datenanalysen durchzuführen und verschiedene Algorithmen anzuwenden.

Die Anwendung von Clustering-Algorithmen ermöglichte es, Muster und Zusammenhänge in den Daten zu erkennen, was entscheidend für die Entwicklung gezielter Marketingstrategien war. Insbesondere erwies sich die Verwendung des k-Means-Algorithmus als erfolgreich, um Kunden in verschiedene Gruppen zu segmentieren und ihr Kaufverhalten besser zu verstehen.

Obwohl das Projekt bereits bedeutende Erkenntnisse geliefert hat, zeigt es auch das Potenzial für zukünftige Analysen mit umfangreicheren Datensätzen. Mit mehr Daten könnten noch detailliertere Einblicke gewonnen werden, um die Effektivität der Marketingstrategien weiter zu verbessern.

Insgesamt hat das Projekt nicht nur gezeigt, wie wertvoll die Anwendung von Datenanalysewerkzeugen wie KNIME sein kann, sondern auch, wie wichtig es ist, Datenanalysen für fundierte geschäftliche Entscheidungen einzusetzen.

10. Auflistung aller verwendeten Nodes

- CSV Reader: Liest .csv Files ein und gibt sie als Tabelle zurück
- Duplicate Row Filter: Entfernt doppelt vorhandene Datensätze
- RowID: Erlaubt es eine eigene RowID festzulegen oder die bisherige RowID zu ersetzen
- RowFilter: Filtert Zeilen nach eigens festgelegten Kriterien
- String to Number: Konvertiert einen String Wert zu einem Datentyp für Zahlen
- Rule Engine: Erlaubt das Definieren von Regeln, nach denen Einträge überprüft werden

- Rule-based Row Filter: Filtert Zeilen nach selbst Definierten Regeln
- Cell Splitter: Spaltet die Werte einer Zelle in verschiedene Spalten auf
- Column Renamer: Benennt eine Spalte um
- Column Aggregator: Fasst mehrere Spalten zu einer zusammen
- String to Date&Time: Konvertiert eine Zeitangabe in String Format zu einem Date Format
- Date&Time-based Row Filter: Filtert Zeilen, die sich nicht innerhalb oder außerhalb eines definierbaren Zeitraums befinden
- GroupBy: Gruppiert Daten nach Attributen und kann Aggregationsfunktionen ausführen
- Joiner: Führt eine Join-Operation auf zwei Tabellen aus und führt diese zusammen
- Math Formula: Erlaubt es mathematische Formeln zu definieren und diese auf die Daten anzuwenden
- Date&Time to String: Wandelt ein Date Format zu einem String Format um
- Pivot: Erstellt eine Pivot Tabelle nach einer festgelegten Ausrichtung
- Line Plot: Erzeugt ein Liniendiagramm zum Datensatz
- Number to String: Wandelt eine Zahl in ein String Format um
- Association Rule Learner (Borgelt): Wendet den Apriori-Algorithmus auf den Datensatz an und gibt Assoziationsregeln zurück
- Category to Number: Ersetzt eine Spalte zu einer ID als Number
- k-Means: Wendet den k-Means Algorithmus an, um Cluster zu erzeugen
- Color Manager: Erlaubt eine selbst definierte Vergabe von Farben für Plots

11. Quellen

- https://www.knime.com/downloads (Download der KNIME Analytics Software)
- https://moodle.thm.de/pluginfile.php/723896/mod_resource/content/1/KNIME_Quickstart.pdf (Quickstart für die Erstellung eines KNIME Workflows)
- https://hub.knime.com/knime/extensions/org.knime.features.js.plotly/latest/org.knime.dynamic.js.v30.DynamicJSNodeFactory:891785b9 (KNIME Plotly Extension)
- https://hub.knime.com/knime/extensions/org.knime.features.ext.itemset/latest/org.knime.ext.itemset.nodes.learner.AssociationRuleLearnerNodeFactory (Association Rule Learner Extension)
- https://hub.knime.com/knime/extensions/org.knime.features.base/latest/org.knime.base.node.mine.cluster.kmeans.ClusterNodeFactory2 (k-Means Extension)
- Abbildung 1-25: Aus dem KNIME-Workflow des Projekts zu entnehmen