Projektleitung DMC 2009
Vorgehensmodelle, Schritte, Varianten, Erkenntnisse, Ergebnisse

Christian Heinrich

Fachhochschule Brandenburg
heinricc@fh-brandenburg.de

Inhaltsverzeichnis

1 Einleitung

Diese Arbeit beschäftigt sich mit dem Vorgehen des Teams FH Brandenburg I beim Data-Mining-Cup 2009. Dazu werden in einer Einleitung die Aufgabe und das Team vorgestellt. Im zweiten Abschnitt der Arbeit wird erläutert, wie die Aufgabe erfüllt wird. In einem Fazit werden die Ergebnisse des Projekts zusammengetragen und die Vorgehensweise beurteilt.

1.1 Data-Mining-Cup

Der Datat-Mining-Cup (DMC) ist ein alljährlich stattfindender Wettbewerb, an dem nationale und internationale Bildungseinrichtungen mit maximal zwei Studententeams teilnehmen können. Bisher haben Bildungseinrichtungen aus über 40 Ländern an diesem Wettbewerb teilgenommen. Damit gilt der DMC als größter Data-Mining-Wettbewerb weltweit. Der DMC wird von dem Data-Mining-Unternehmen prudsys AG ausgerichtet.

1.2 Aufgabe und Termine

Die Aufgaben des DMC werden jedes Jahr von Wettbewerbspartnern bestimmt. Im Jahr 2009 ist die Libri GmbH der Wettbewerbspartner des DMC. Die Aufgabe ist die genauest mögliche Vorhersage für den Verkauf von 8 Buchtiteln an 2418 verschiedenen Standorten.

Für das Erstellen und Traininieren eines dafür geeigneten Klassifikators werden simulierte Einkaufsdaten von weiteren 2394 Standorten zur Verfügung gestellt. Die Attribute der Datenmenge sind 1856 Warengruppen von Büchern. Alle Daten beziehen sich auf einen festgelegten Zeitabschnitt.

Die Registrierung für den DMC 2009 ist ab dem 1. April 2009 möglich. Die Aufgabe wird mit Start des Wettbewerbes am 15. April 2009 veröffentlicht. Der Wettbewerb endet am 25. Mai 2009. Bis zu diesem Termin müssen die Vorhersagewerte für den Verkauf der 8 Buchtitel an den 2418 Standorten eingereicht werden. Die drei besten Teams stellen ihr Lösungsmodell am 23. Juni 2009 im Rahmen der prudsys Anwendertage vor und erhalten Preisgelder zwischen 1000 und 2500 EUR.

1.3 Vorbedingungen und Teammitglieder

Die Fachhochschule Brandenburg nimmt im Jahr 2009 zum ersten Mal am DMC teil. Dadurch kann bei der Erfüllung der Aufgabe nicht auf Erfahrungen aus dem Vorjahr bzw. aus den Vorjahren zurückgegriffen werden.

Das Team der FH Brandenburg für den DMC 2009 setzt sich aus 2 Bachelorstudenten, 3 Masterstudenten und einem Betreuer zusammen. Alle 5 Studenten verfügen zu Beginn des Projekts über Grundlagenwissen in dem Bereich „Künstliche Intelligenz“ haben aber keinerlei Erfahrungen mit Data-Mining.

Zur Erfüllung der Aufgabe wird das Programm RapidMiner[1] verwendet. RapidMiner ist ein Data-Mining-Programm, dass die Zusammenstellung von Data-Mining-Prozessen nach dem Baukastenprinzip ermöglicht. Dafür stellt es eine Vielzahl von Operatoren für die Umsetzung von Vorverarbeitungs- und Modellierungsschritten bereit. Keiner der Studenten ist zu Beginn des Projekts mit diesem Programm vertraut.

[1] http://www.rapidminer.com

2 Vorgehen Datamining Cup 2009

In diesem Abschnitt wird das Vorgehen des Teams FHB beim DMC 2009 erläutert. Der größte Teil der Arbeit erfolgt selbstständig. Die Teammitglieder treffen sich jedoch zwischen dem 19. März 2009 und dem 25. Mai 2009 einmal wöchentlich, um Arbeitsergebnisse auszuwerten, neue Ideen und das weitere Vorgehen zu besprechen. Diese Treffen, die Arbeitsergebnisse, die Vorgehensschritte sowie die Arbeitsverteilung werden in einem Wiki des Moodle-Systems der FH Brandenburg protokollarisch zusammen getragen und sind somit für alle Teammitglieder jederzeit verfügbar.

2.1 Vorbereitung

Vor der Veröffentlichung der Aufgabe des DMC 2009 am 15. April 2009 stehen dem Studententeam 4 Wochen zur Verfügung, um sich in das Thema Datamining einzuarbeiten, das Programm RapidMiner kennenzulernen und sich mit DMC-Aufgaben und -Lösungen der Vorjahre zu beschäftigen.

Die Einarbeitung in das Thema Datamining erfolgt zu einem Teil durch Seminare, die zu den wöchentlichen Terminen vom Teambetreuer Dipl.-Inform. Ingo Boersch geleitet werden. Diese Seminare werden durch ein Selbststudium der Studenten ergänzt, welches zu großen Teilen auf [WiFr01] basiert.

Zur Einarbeitung in das Programm RapidMiner werden die Tutorials des Programms verwendet.

Da die FH Brandenburg bisher noch nicht am DMC teilgenommen hat und somit nicht auf eigene Erfahrungen zurückgreifen kann, werden als Vorbereitung auf den DMC 2009 die Aufgaben und Siegerlösungen der Jahre 2005 und 2007 der RWTH Aachen betrachtet. Aus den Siegerlösungen der Vorjahre sollen Erfahrungswerte zur Organisation des Teams, Analyse und Vorverarbeitung der zu klassifizierenden Daten und zum Vorgehen bei Auswahl, Test und Evaluierung von Klassifikatoren und den daraus entwickelten Modellen gewonnen werden. Bei der Auswertung der Dokumente [BGH+08] und [WeBu05] fallen dazu insbesondere folgende Punkte auf:

- Organisation des Teams
  - Ein wöchentliches Treffen für Diskussionen, neue Ideen und Aufgabenverteilung
  - Zwischen den Treffen Aufgaben durchführen
  - Nutzung eines Wiki-Systems zur Dokumentation der durchgeführten Experimente
  - Motivation: Ergebnisse der anderen Teammitglieder übertreffen (die Güte der Ergebnisse wird anhand eines Performance-Wertes beurteilt, diesen gilt es zu übertreffen)

- Datenanalyse und Vorverarbeitung:
  - Daten analysieren und verstehen
  - Nach Zusammenhängen zwischen Attributen und Klasse suchen
  - Versuchen wichtige Attribute, zu identifizieren
  - Falls sinnvoll, neue Attribute erstellen (z. B. Verwendung der Zeilensumme der vorhandenen Attribute, wenn die Klasse zu den Attributen in der Summe eine Abhängigkeit zeigt aber nicht zu den einzelnen Attributen [vgl. BGH+08])
  - Umwandeln von Attributen (z. B. für eine Altersangabe eine Datumsangabe in einen zeitlichen Abstand umwandeln [vgl. WeBu05])
  - Feature-Selection – iterative Methode der Attributauswahl, bei der der Attributraum nach einer Teilmenge von Attributen durchsucht wird, die die Klasse mit der größten Wahrscheinlichkeit am besten vorhersagen wird. Dazu wird der Raum in einer von zwei Richtungen – vorwärts oder rückwärts - gefräßig durchsucht. Bei der Vorwärtsauswahl beginnt die Suche ohne Attribute und in jedem Schritt wird der Teilmenge ein Attribut hinzugefügt. Bei der Rückwärtsauswahl wird mit der vollen Attributmenge begonnen und in jedem Schritt ein Attribut gelöscht. [vgl. WiFr01]

- Klassifikation
  - Gewinnkriterium berücksichtigen – beachten ob für die Punktzahl, die verschiedene Fehler unterschiedlich bewertet werden (kostensensitive Klassifikation) oder nur die gesamte Anzahl der Fehlklassifikationen ausschlaggebend ist
  - Keine Entwicklung neuer Verfahren, sondern Verwendung bewährter Klassifikationsverfahren wie z. B. Support Vector Machines, künstliche neuronale Netze oder log-lineare Modelle
  - Verwendung von Klassifikatorkombinationen zum Ausgleich von Schwächen einzelner Methoden durch Stacking oder Voting

- Modellauswahl
  - Bestimmung der Prognosefähigkeit durch Kreuzvalidierung
  - Vermeidung von Überanpassung durch Aufteilen der Trainingsdaten in eine Trainingsmenge und eine Validierungsmenge – Diese Methode dient der Vermeidung von Überanpassung. Dabei wird ein zufällig gewählter Teil, oft zwischen 10% und 20%, der Trainingsdaten abgetrennt und für die Modellsuche nicht berücksichtigt. Mit diesen zurückgehaltenen Datensätzen wird später überprüft, wie das entwickelte Modell auf ungesehenen Daten generalisiert. [vgl. BGH+08]
  - Bewertung der verwendeten Klassifikatoren bzw. der erstellten Modelle durch Scores bzw. Performanz

- Kombination von guten Klassifikatoren liefert ein besseres Modell als ein einzelner guter Klassifikator

Für das Stundententeam der RWTH Aachen im DMC 2007 waren insbesondere Vorverarbeitung, Kreuzvalidierung und Klassifikatiorkombinationen die Schlüssel zu reproduzierbaren, guten Ergebnissen [BGH+08].

Die Organisation des Studententeams FH Brandenburg_I wird genau nach den Richtlinien, die unter dem o. g. Punkt „Organisation des Teams“ genannt werden, durchgeführt. In den folgenden Abschnitten wird erläutert, wie auch die restlichen Erfahrungswerte, zu finden unter den o. g. Punkten „Datenanalyse und Vorverarbeitung“, „Klassifikation“ und „Modellauswahl“, in die Aufgabenerfüllung mit einfliessen.

2.2 Analyse der Daten

Die Analyse und Vorverarbeitung der Daten ist ein wichtiger Schritt im Dataminingprozess und nimmt oft den größten Teil des gesamten Datamingprozesses in Anspruch [vgl. Boersch09].

Mit der Veröffentlichung der Aufgabe werden zwei Datenmengen zur Verfügung gestellt – eine Trainingsdatenmenge zur Erstellung eines Modells und eine Testdatenmenge, mit der eine Vorhersage für die 8 Buchtitel zu erstellen ist.

Die folgende Tabelle zeigt einen Ausschnitt der Datensätze aus der Trainingsmenge. Die Werte der Spalte ID stehen für Orte, Werte der Spalten WGXXXXX geben an, wieviele Bücher einer Warengruppe an einem bestimmten Ort verkauf wurden. Die erste Ziffer einer Warengruppe enthält Informationen über die Art der Produkte z. B. Hardcover oder Paperback. Die restlichen vier Ziffern enthalten hierarchische Informationen zum Inhalt der Bücher einer Warengruppe, so kann bspw. die zweite Ziffer für „Fiction“ und die dritte Ziffer für „Science Fiction“ stehen. Die Werte der Spalten T1 bis T8, deren Werte in den Testdaten unbekannt sind, geben die Verkaufszahlen eines Buchtitels an einem bestimmten Ort an. Die Vorhersage dieser Werte ist die Aufgabe des Wettbewerbs.

ID	WG 00000	WG 23000	WG 23110	WG 23120	...	WG 12600	WG 12700	WG 12900	T1	...	T8
3377	0	0	10	0	...	0	0	0	0	...	0
4355	0	0	0	0	...	0	0	0	0	...	0
1119	0	0	7	0	...	0	0	0	0	...	0
4429	0	0	0	0	...	0	0	0	0	...	0
2428	3	0	3233	0	...	0	0	0	0	...	0
1866	6	8	1451	4	...	0	0	0	1	...	7
2458	0	6	347	0	...	0	0	0	0	...	0
4531	7	12	1302	10	...	0	0	1	1	...	3
1884	0	0	339	0	...	0	0	0	0	...	0
5524	0	0	474	0	...	0	0	6	0	...	4
3647	0	3	372	0	...	0	0	19	1	...	4
...	...	...	...	...	...	...	...	...	...	...	...

Tab. 1: Ausschnitt aus der Trainingsmenge

Beim ersten Betrachten der Daten fällt insbesondere die sehr hohe Anzahl der Attribute mit 1856 Warengruppen auf. Weiterhin fällt eine große Häufigkeit von Nullwerten auf - sowohl in den regulären Attributen als auch in den Klassen.

Bei genauerer Betrachtung der Verteilung der Daten fällt neben der großen Häufigkeit von Nullwerten auf, dass die meisten Verkaufszahlen zwischen 0 und 3000 liegen. Es existieren aber auch vereinzelte Ausreißerwerte mit Verkaufszahlen von bis zu 170000. Die folgende Abbildung zeigt die Verteilung der Trainingsdaten mit der großen Häufigkeit von kleinen Werten und den großen Ausreißerwerten ohne die Werte der Klassenspalten.

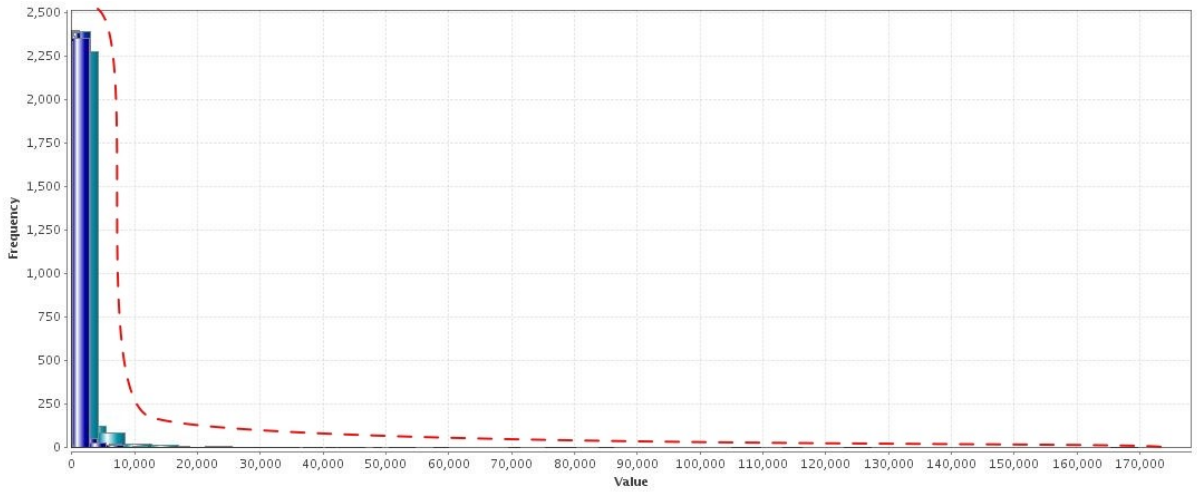

Abb. 1: Verteilung der Trainingsdaten

Diese Art der Verteilung wird als Long-Tail-Verteilung bezeichnet und ist durch sehr viele kleine Werte, einige mittlere Werte und vereinzelte sehr hohe Werte charakterisiert. Da die hohen Werte durch ihre geringe Häufigkeit nur schwer zu erkennen sind, wird diese Werteverteilung in Abbildung 1 durch die rote gestrichelte Linie verdeutlicht.

Eine weitere Möglichkeit zur Visualisierung der Verteilung der Trainingsdaten bietet das folgende Diagramm. Dabei sind niedrige Verkaufszahlen blau und hohe Verkaufszahlen rot dargestellt.

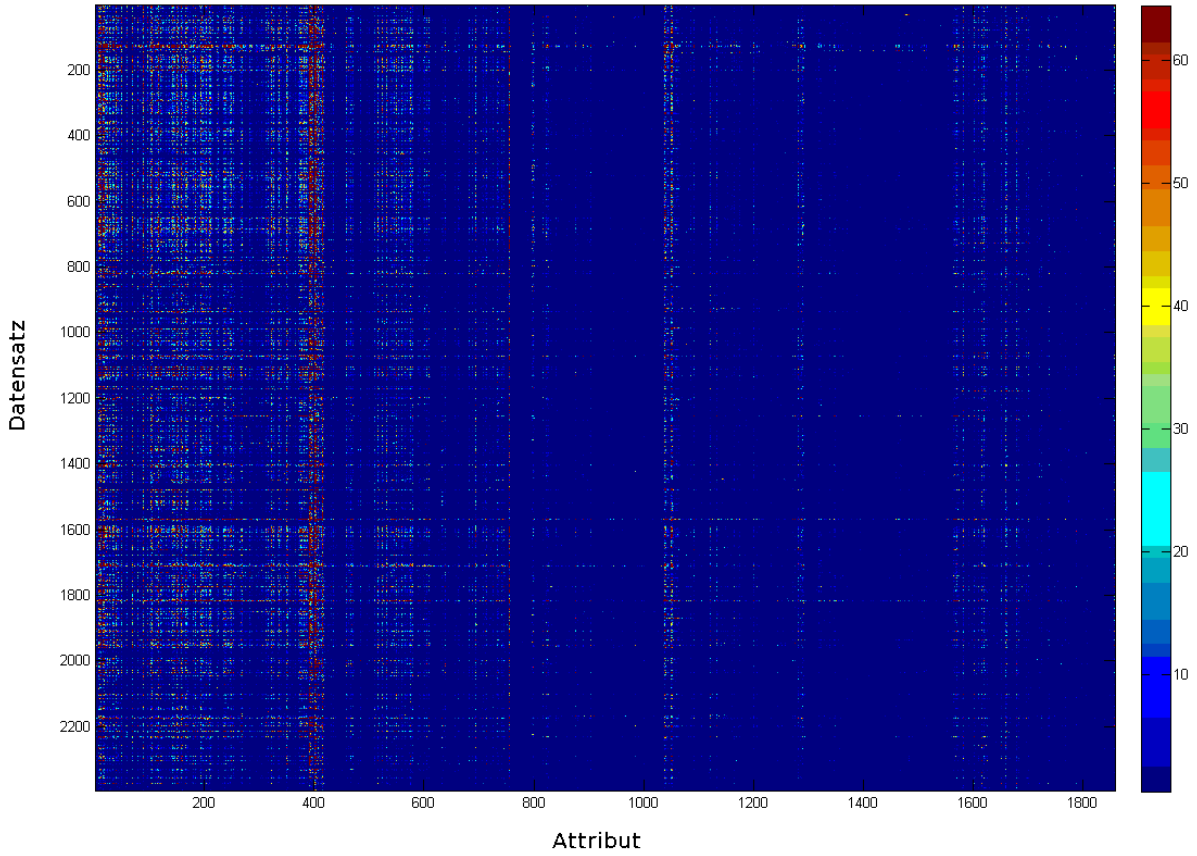

Abb. 2: Farbige Darstellung der Trainingsdaten

Aus Abbildung 2 werden ebenfalls die insgesamt geringen Verkaufszahlen deutlich. Durch vertikale und horizontale rote Linien ist aber auch zu erkennen, dass es sowohl Warengruppen als auch Orte mit besonders hohen Verkaufszahlen existieren.

Die Verteilung der Verkaufszahlen in den Klassenspalten ist ebenfalls auffällig. Die Spalte T6 weist mit 2275 verkauften Titeln an einem Ort die mit Abstand größte Verkaufszahl auf. Auch insgesamt wurden von T6 deutlich mehr Exemplare verkauft als von allen anderen Klassen. Die folgende Tabelle zeigt in der Spalte „Statistics“ die durchschnittlichen Verkaufszahlen und in der Spalte „Range“ die niedrigste und die höchste Verkaufszahl der Klassen T1 bis T8.

Name	Value Type	Statistics	Range
T1	integer	avg = 0.255 +/- 1.676	[0.000 ; 51.000]
T2	integer	avg = 0.534 +/- 2.970	[0.000 ; 96.000]
T3	integer	avg = 0.195 +/- 1.449	[0.000 ; 34.000]
T4	integer	avg = 0.234 +/- 1.864	[0.000 ; 63.000]
T5	integer	avg = 1.950 +/- 19.377	[0.000 ; 853.000]
T6	integer	avg = 5.617 +/- 49.094	[0.000 ; 2,275.000]
T7	integer	avg = 0.631 +/- 10.139	[0.000 ; 438.000]
T8	integer	avg = 0.801 +/- 22.372	[0.000 ; 1,090.000]

Tab. 2: Mittelwert, Standardabweichung und Wertebereich der Klassen in den Trainingsdaten

2.2.1 Schlussfolgerungen und erste Schritte

Da nicht nur eine Vorhersage getroffen werden muss sondern 8 und somit auch 8 verschiedene Modelle entwickelt werden müssen, ist es notwendig die Trainingsdaten in 8 Tabellen zu überführen, die jeweils alle Datensätze aber nur eine Klasse enthalten.

Da sich das Studententeam aufgrund der Erfahrungswerte von [BGH+08] entschieden hat zur Qualitätssicherung der Modelle, insbesondere zur Vermeidung von Überanpassung, ebenfalls eine Unterteilung der Trainingsdaten in Testmenge und Validierungsmenge vorzunehmen, müssen die 8 Trainingsdatensätze nochmals geteilt werden. Die Validierungsmenge entspricht üblicherweise 10% bis 20% der Trainingsdatenmenge [vgl. BGH+08]. Deshalb wird sich entschieden, jeden der 8 Trainigsdatensätze in einem Verhältnis von 90% Trainingsmenge zu 10% Validierungsmenge aufzuteilen. Dabei wird darauf geachtet, dass die Aufteilung der Datenmenge stratifiziert geschieht und somit bspw. verhindert wird, dass eine der beiden Teilmengen nur extreme Ausreißerwerte enthält.

Durch die große Anzahl von Warengruppen, deren Verkaufszahlen fast ausschließlich 0 entsprechen, stellt sich die Frage, ob alle Warengruppen für die Vorhersage der Klasse benötigt werden, oder ob bestimmte Warengruppen zu Gruppen zusammengefasst werden oder eliminiert werden können. In diesem Zusammenhang gilt es auch zu klären, welche Attribute in welchem Maße mit der Klasse korrelieren oder ob durch die Entwicklung neuer Attribute aus den vorhandenen eine stärkere Beziehung zu der Klasse entdecken ist. Dazu werden Spaltensummen, Zeilensummen, Summen mit gleicher Art der Produkte (Warengruppen, bei denen die erste Ziffer gleich ist) und Summen der Warengruppen mit gleichem Inhalt gebildet und auf ihre Korrelation zur Zielklasse geprüft.

Die Daten sind keine realen Daten sondern simulierte Daten. Das heißt, dass die Daten durch ein Programm erstellt wurden und somit eine Funktion existieren muss, die aus den Attributen die Klassen berechnen kann. Daraus entstehen zwei mögliche Ansätze für die Erstellung der Lösung. Der erste Ansatz liegt in der Entwicklung einer eigenen Funktion die sich mittels Regression der Funktion der Aufgabe annähert. Eine andere Möglichkeit die Funktion in Erfahrung zu bringen, bietet Social Engineering. Obwohl dieser Ansatz ergebnisorientiert ist, wird ihm nicht nachgegangen, da das Studententeam beschliesst, die Aufgabe mit Anwendungen der künstlichen Intelligenz zu lösen.

Die Verkaufszahlen in der Klasse T6 der Trainingsdaten sind wesentlich höher als in den restlichen Klassen. Deshalb kann davon ausgegangen werden, dass diese Klasse auch für die

Testdaten eine große Rolle spielt und diese Klasse somit die wichtigste der 8 Klassen ist. Aus diesem Grund wird das erste Modell für T6 entwickelt.

Um die Prognosefähigkeit der entwickelten Modelle beurteilen zu können und um eine Verbesserung zu ermöglichen, wird eine Kenngröße benötigt, an der die Lösungen gemessen werden können. In [BGH+08] wird diese Kenngröße als „Score“ bezeichnet und mit Hilfe einer fünffachen Kreuzvalidierung erstellt. Nach diesem Vorgehen richtet sich auch das Studententeam. Die Aufteilung der Trainingsdaten in Trainings- und Validierungsmenge und die Auftailung der Trainingsmenge durch die fünffache Kreuzvalidierung entsprechen also exakt dem Voregehen von [BGH+08]. Die folgende Grafik zeigt diese Aufteilung der Mengen. Dabei stellen blaue Blöcke die Kreuzvalidierungsmengen der Trainingsmenge dar, der grüne Block stellt die Validierungsmenge dar, und der rote Block stellt die Testdaten dar.

Abb. 3: Einteilung der Daten in Kreuzvalidierungsmenge,Validierungsmenge und Testdaten [BGH+08]

Für die Entwicklung der Modelle muss auch berücksichtigt werden, wie das Ergebnis bewertet wird. In dieser Aufgabe wird die Qualität des Ergebnisses nach folgender Formel berechnet.

$$d = \sum_{i=1}^{8} \sum_{j=1}^{2418} \left| p_{ij} - r_{ij} \right|$$

Dabei steht d für die Qualität des Ergebnisse, p_{ij} ist der vorhergesagte Wert für den i-ten Titel am j-ten Ort, und r_{ij} ist der wirkliche Wert für den i-ten Titel am j-ten Ort. Aus dieser Formel lässt sich ableiten, dass die Bewertung der Vorhersage nicht kostensensitiv sondern nach der Anzahl der Fehlklassifikationen bzw. der Summe des Betrags der Differenz zwischen vorhergesagtem und wirklichem Wert erfolgt. Das heißt, dass die Abweichung einer vorhergesagten Verkaufszahl von der wirklichen Verkaufszahl die Qualität immer linear beeinflusst. Für die zu treffende Vorhersage bedeutet das, dass eine gewisse Grundqualität damit gesichert werden kann, dass negative Vorhersagen auf 0 aufgerundet werden, da eine Verkaufszahl nicht kleiner als 0 sein kann, und die Verteilung der vorhergesagten Klassen der Trainingsdaten ungefähr der Verteilung der wirklichen Klassen der Trainingsdaten entspricht. Dabei muss aber eine Überanpassung vermieden werden.

2.3 Aufbereitung der Daten

Nach dem Aufteilen der Trainingsdatenmenge mit allen 8 Klassenspalten in Datensätze mit nur einer Klassenspalte, werden die erhaltenen 8 Trainingsmengen weiter unterteilt in Trainings- und Validierungsmengen. In Rapidminer wird dafür der Operator AbsoluteSplitChain verwendet. Mit Parametern lassen sich Anzahl der Datensätze in der Trainingsmenge, Anzahl der Datensätze in der Validierungsmenge und die Art der Aufteilung der Datensätze einstellen. Dazu werden die Daten mit einem ExampleSource-Operator eingelesen und für die weitere Verwendung mit zwei ExampleSetWriter-Operatoren gespeichert. Abbildung 4 zeigt diesen Prozess in der Sicht von Rapidminer.

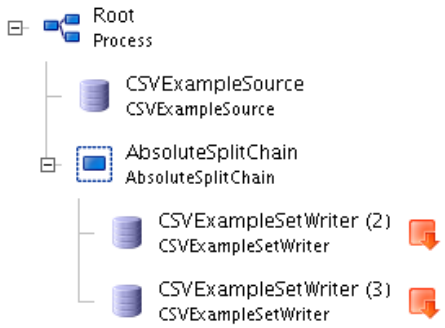

Abb. 4: Aufteilung in Trainings- und Validierungsmenge

In den nächsten Schritten der Vorverarbeitung wird nur noch mit der Trainingsmenge gearbeitet. Die Validierungsmenge wird erst zur Bewertung des mit der Trainingsmenge entwickelten Modells verwendet.

Der nächste Schritt ist die Entwicklung eines ersten einfachen Modells für T6. Wichtiger als das erste Modell ist der dabei entstehende Score des verwendeten Klassifikators. Dieser Score dient als Basiswert, den alle nachfolgend verwendeten Klassifikatoren schlagen müssen. Als erster Klassifikator wird in Rapidminer der Operator DefaultLearner verwendet, der durch Einstellung eines Parameters für jede Zeile der Klassenspalte „0“ vorhersagt. Abbildung 5 zeigt diesen Prozess in der Sicht von Rapidminer.

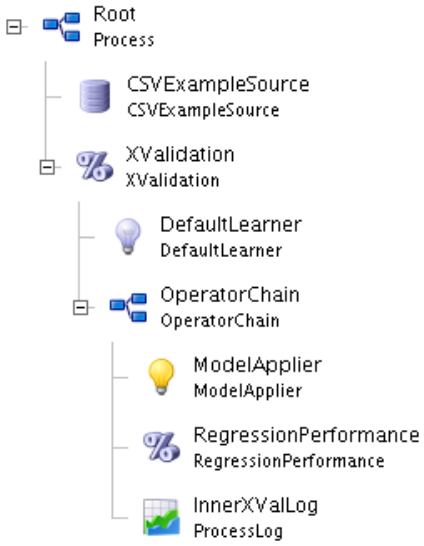

Abb. 5: Entwicklung des ersten Modells und Scores in Rapidminer

Dabei ist auch zu sehen, dass die Entwicklung des Modells bereits in eine Kreuzvalidierung, dargestellt durch den Operator XValidation, eingebettet ist. Der Operator RegressionPerformance liefert einen Performanzvektor, der je nach Parametereinstellungen aus einer Vielzahl von Werten bestehen kann. Für den Score ist jedoch nur ein Wert dieses Vektors von Bedeutung - der absolute Fehler. Dieser Score beträgt 5.671.

Die Aufgabe der nächsten Schritte ist es, diesen Score zu übertreffen bzw. zu unterbieten. Das soll z. B. durch das Entfernen unnötiger Attribute, das Testen verschiedener Klassifikatoren und das Modifizieren der Parameter von Operatoren geschehen.

Die Tabelle auf der folgenden Seite fasst den Verlauf dieser Entwicklung zusammen. Der Score wird dabei von 5.671 auf 2.814 verbessert. Ein besonders großer Sprung von 5.064 auf 4.182 ist durch die Einführung der Feature-Selection zu beobachten. Die letzte große Verbesserung wird durch eine zusätzliche Attributauswahl erreicht, die der Feature-Selection vorgeschaltet wird. Diese Attributauswahl beruht auf einem Chi-Quadrat-Test. Dabei werden Attributen Gewichte zugewiesen, die je nach Relevanz des Attributs für das Klassenattribut, 0 oder 1 betragen. Attribute mit einem Gewicht von 0 werden danach eliminiert.

Score	Zeitstempel	Eigenschaften / Änderungen
5.671	23.04.2009	• 5-fache Kreuzvalidierung mit DefaultLearner, der „0“ vorhersagt
5.158	24.04.2009	• 12-fache Kreuzvalidierung • DefaultLearner durch JMySVMLearner mit Standardeinstellungen ersetzt • Entfernen überflüssiger Attribute durch Operator RemoveUselessAttributes
5.064	25.04.2009	• 12-fache Kreuzvalidierung durch 5-fache Kreuzvalidierung ersetzt
4.182	30.04.2009	• Entfernen weiterer Attribute durch Feature Selection (Operator FeatureSelection) mit Vorwärtsselektion • 5-fache Kreuzvalidierung eingebettet in Feature Selection (wird deshalb in jedem Iterationsschritt der Feature Selection durchgeführt)
3.407	30.04.2009	• JMySVMLearner durch NearestNeighbors-Operator ersetzt
3.277	30.04.2009	• NearestNeighbors-Operator durch AttributeBasedVote-Operator ersetzt
3.047	30.04.2009	• AttributeBasedVote-Operator durch JMySVMLearner-Operator mit veränderten Parametern „convergence_epsilon“ und „L_pos“ ersetzt
2.814	01.05.2009	• RemoveUselessAttributes ersetzt durch AttributeWeightSelection-Operator, der überflüssiger Attribute mit Hilfe von Gewichten entfernt • Gewichte werden mit einem Chi-Quadrat-Test (Operator ChiSquaredWeighting) auf den Trainingsdaten erstellt und danach gespeichert • Diskretisieren der Daten für den ChiSquaredWeighting-Operator, weil dieser nur mit nominalen Attributen funktioniert (Operator AbsoluteDiscretization) • Entfernen weiterer Attribute durch Feature Selection (Operator FeatureSelection) mit Vorwärtsselektion • JMySVMLearner durch LinearRegression-Operator ersetzt

Tab. 3: Entwicklung der Scores für die Vorverarbeitung der Trainingsdaten für die Klasse T6

Die Vorverarbeitungsschritte für die Trainingsdaten für die restlichen 7 Klassen erfolgen nach dem gleichen Prinzip.

2.4 Erstellung der Vorhersagen

Nachdem der Prozess der Vorverarbeitung feststeht, kann mit der Erstellung der Vorhersagen begonnen werden. Bevor jedoch ein Klassifikator, der in dem Vorverarbeitungsschritt einen guten Score erzielt hat, benutzt wird, um mit ihm ein Modell auf den gesamten Trainingsdaten zu erstellen, muss er sich auf ungesehenen Daten bewähren. Dazu wird der Klassifikator auf der zurückgehaltenen Validierungsmenge angewendet und dabei nochmals ein Score erstellt.

Der Klassifikator, der den besten Score auf der Validierungsmenge erzielt, wird verwendet um das Modell aus den gesamten Trainingsdaten zu erstellen. Um eine Auswahl von Klassifikatoren zu erhalten, erstellt jeder der Studenten zu jeweils 4 Klassen einen Klassifikator. Jede dieser Lösungen basiert auf dem gleichen Prozess in Rapidminer. Es werden lediglich Klassifikatoren ausgetauscht oder Parameter modifiziert. Die folgende Tabelle zeigt die Lösungen mit dem höchsten Score auf der Validierungsmenge. Diese Klassifikatoren werden für die Erstellung der Vorhersage verwendet.

Klasse	Score auf der Validierungsmenge	Klassifikator / Operatorbezeichnung in Rapidminer
T1	0.340	JMySVM mit 10-facher Kreuzvalidierung
T2	0.365	W-M5P mit 5-facher Kreuzvalidierung
T3	0.372	W-M5P mit 10-facher Kreuzvalidierung
T4	0.216	JMySVM mit 10-facher Kreuzvalidierung
T5	1.885	W-M5P mit 5-facher Kreuzvalidierung
T6	3.532	W-M5P mit 5-facher Kreuzvalidierung
T7	0.495	W-M5P mit 5-facher Kreuzvalidierung
T8	0.417	W-M5P mit 5-facher Kreuzvalidierung

Tab. 4: Klassifikatoren für die Erstellung der Vorhersage

2.5 Gesamtprozess

In diesem Abschnitt wird der von Hannes Uhlmann entwickelte Prozess vorgestellt, der von von den Trainings- und Testdaten zu den Vorhersagen für die einzelnen Klassen führt.

Die Abbildung auf der folgenden Seite dient als Visualisierung des Prozesses.

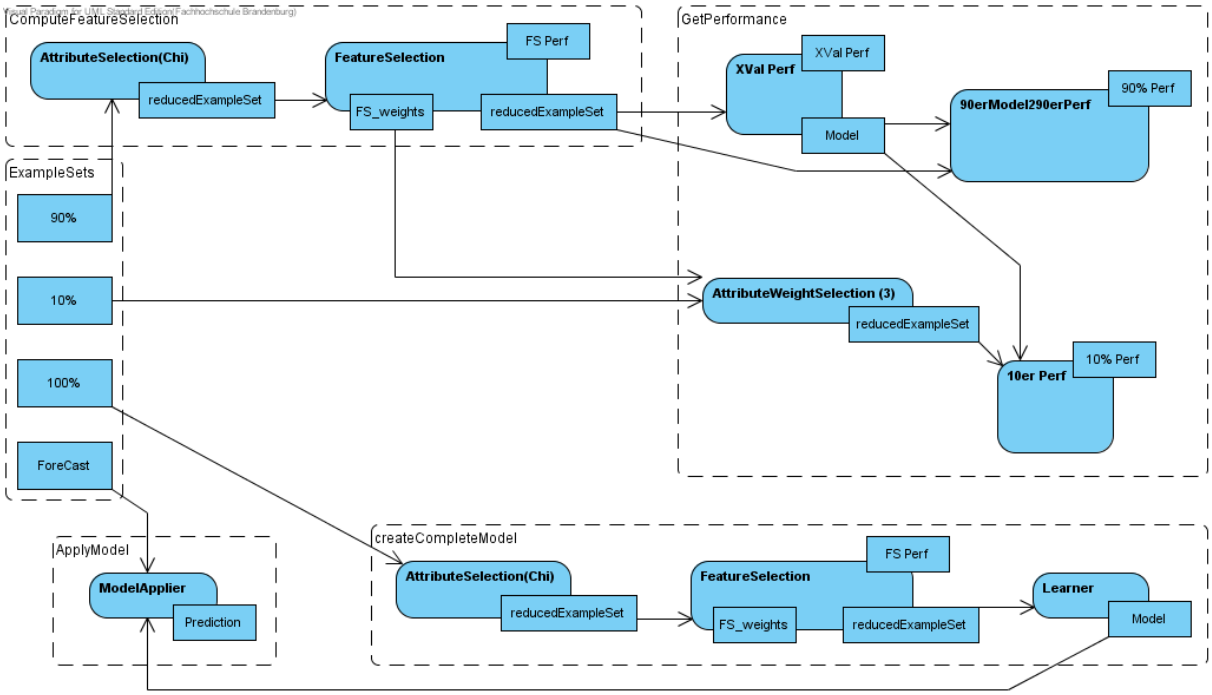

Abb. 6: Visualisierung des Prozesses [Uhlmann09]

Die Datenmengen sind in dem Kästchen „ExampleSets“ zu sehen. Hier befinden sich sowohl die Testdaten als auch die Trainingsdaten (komplett und aufgeteilt in Trainings- und Validierungsmenge).

Der Prozess läuft im Uhrzeigersinn von „ComputeFeatureSelection“ über „GetPerformance“ und „CreateCompleteModel“ bis zu „ApplyModel“ ab.

ComputeFeatureSelection

Im ersten Schritt wird auf der Trainingsmenge, gekennzeichnet als „90%“, der Chi-Quadrat-Test durchgeführt und Attribute mit geringer Relevanz für die Zielklasse eliminiert. Das Ergebnis ist eine Trainingsmenge mit weniger Attributen, gekennzeichnet als „reducedExampleSet“. Auf dieser Trainingsmenge wird die Feature-Selection durchgeführt. Dabei werden wiederum Attribute eliminiert, die Trainingsmenge verkleinert sich weiter und es entsteht ein Performance-Wert, gekennzeichnet als „FS Perf“.

Die für die Eliminierung verwendeten Gewichte werden gespeichert, um sie im nächsten Schritt auf die Validierungsmenge anwenden zu können.

GetPerformance

In diesem Schritt wird zunächst eine Kreuzvalidierung durchgeführt. Dabei wird auf die Trainingsmenge, letztes „reducedExampleSet“ aus „ComputeFeatureSelection“, ein Klassifikator angewendet. Hierbei entsteht das Modell und ein Performance-Wert.

Das entstandene Modell wird auf die Trainingmenge angewendet, wobei eine Vorhersage entsteht aus der ein weiterer Performance-Wert abgeleitet wird.

Als nächstes wird die Validierungsmenge geladen. Aus der Validierungsmenge werden mit Hilfe der gespeicherten Gewichte Attribute eliminiert, gekennzeichnet durch „Attribute-

WeightSelection (3)“. Auf die vorverarbeitete Validierungsmenge wird jetzt das Modell angewendet, wobei wieder eine Vorhersage entsteht aus der ein Performance-Wert abgeleitet wird.

Dieser Performance-Wert ist besonders wichtig, da er darüber Auskunft gibt, wie gut das Modell mit ungesehenen Daten umgehen kann.

CreateCompleteModel

In diesem Schritt werden zunächst Attribute aus der kompletten Trainingsmenge eliminiert, wie in dem Schritt „ComputeFeatureSelection“ beschrieben. Auf den entstandenen Datensatz wird der Klassifikator angewendet, der auch in den anderen Schritten zum Einsatz kommt. Hierbei entsteht das Modell, dass für die Vorhersage der Zielklassen benutzt wird.

ApplyModel

Im letzten Schritt wird das Modell aus „CreateCompleteModel“ auf die Testdaten angewendet. Dabei entsteht die Vorhersage für eine der Zielklassen.

3 Zusammenfassung

Der letzte Abschnitt fasst die Arbeit zusammen. Es wird geklärt, inwiefern, die Aufgabenstellung erfüllt werden konnte und welche Probleme dabei auftraten. Die Arbeit wird mit einem Fazit abgeschlossen.

3.1 Erfüllung der Aufgabenstellung

Die in dieser Arbeit beschriebene Vorgehensweise zur Erfüllung der Aufgabe des DMC 2009 hat zu einer Lösung geführt, die bis vor dem Wettbewerbsende eingereicht werden konnte. Die Aufgabe wurde somit erfüllt. Wie gut diese Lösung, verglichen mit den anderen eingereichten Lösungen ist, ist zu diesem Zeitpunkt noch nicht bekannt.

3.2 Herausforderungen

Die nicht vorhandenen Erfahrungen des Teams im Bereich Data-Mining in Verbindung mit einem engen Zeitplan stellten die größten Herausforderungen bei der Erfüllung der Aufgabe dar. Es konnten jedoch wichtige Erfahrungswerte zur Organisation des Teams, Analyse und Vorverarbeitung der zu klassifizierenden Daten und zum Vorgehen bei Auswahl, Test und Evaluierung von Klassifikatoren aus den Berichten der RWTH Aachen gewonnen werden. Die meisten der unter 2.1 genannten Punkte konnten erfolgreich umgesetzt werden. Allerdings konnten aus Zeitmangel bspw. keine Kombinationen von Klassifikatoren getestet werden - eine Methode, die in [BGH+08] als sehr erfolgreich vorgestellt wird.

Beim Einsatz von Rapidminer hat sich herausgestellt, dass mit dem Tool Dataminingprozesse insgesamt leicht umzusetzen sind. Dabei lässt sich jeder Stand des Prozesses leicht visualisieren. Operatoren können auch ohne ein tieferes Verständnis für die Algorithmen, die sich hinter ihnen verbergen, beliebig ausgetauscht und parameterisiert werden. Diese Möglichkeiten gepaart mit einem Mangel an Erfahrung haben zu einem sehr experimententellen Vorgehen bei der Auswahl von Klassifikatoren geführt. Es wurde alles versucht um den Score der anderen zu schlagen. Dabei haben sich mehrere Methoden als erfolgreich herausgestellt z. B. lineare Regression, Support Vector Machines und Modellbäume während andere Methoden wie z. B. genetische Algorithmen überraschend scheiterten.

Die Organisation des Teams mit wöchentlichen Treffen zum Auswerten des aktuellen Standes und zum Entwickeln neuer Ideen hat gut funktioniert. Auch die Benutzung eines Wikis als gemeinsame Wissensbasis hat sich als erfolgreich herausgestellt. Jedoch ist hierbei darauf zu achten, dass nach einigen Wochen die Struktur verloren gehen kann und dadurch wichtige neue Einträge übersehen werden können.

Als wichtigste Klasse wurde die Klasse T6 mit besonders hohen Verkaufszahlen identifiziert. Deshalb hat sich das Team besonders auf ein gutes Ergebnis für diese Klasse konzentriert. Zum Ende des Projekts hat sich jedoch herausgestellt, dass dadurch zu wenig Zeit bleibt, um auch die anderen Klassen noch ausführlich untersuchen zu können. Eine Aufgabenaufteilung des Teams zu einem früheren Zeitpunkt wäre im Nachhinein wünschenswert gewesen.

3.3 Fazit

Die größte Herausforderung für die Erfüllung der Aufgabe war der Mangel an Erfahrung. Diese Herausforderung konnte durch das Lernen von erfolgreichen Teams und durch den Einsatz von Rapidminer überwunden werden. Der Erfolg der eingereichten Lösung ist noch abzuwarten. Ungeachtet dessen kann die FH Brandenburg bei künftigen Data-Mining-Wettbewerben auf eigene Erfahrungen zurückgreifen. Insbesondere der unter 2.5 vorgestellte Gesamtprozess von den Trainings- und Testdaten zu den Vorhersagen kann dann eine große Hilfe sein.

Literatur

[Boersch09] I. Boersch: Data Mining – Suche nach verborgenen Mustern. In: Vorlesung Künstliche Intelligenz (2009)

[BGH+08] C. Buck, T. Gass, A. Hannig, J. Hosang, S. Jonas, J.-T. Peter, P. Steingrube, J. H. Ziegeldorf: Data-Mining-Cup 2007. In: Informatik-Spektrum, Springer (2008).

[Uhlmann09] H. Uhlmann: Dokument nicht veröffentlicht (2009).

[WeBu05] T. Weyand, C. Buck: Tausendmal probiert. In: http://www-i6.informatik.rwth-aachen.de/~deselaers/dmc-lab/reports/report-weyand-buck.pdf (2005).

[WiFr01] I. H. Witten, E. Frank: Data Mining – Praktische Werkzeuge und Techniken für das maschinelle Lernen, Carl Hanser Verlag (2001).