Datenanalyse + Wissenserwerb
Modulendprüfung 2009

Die nachfolgende Übungsprüfung dient als Lernhilfe für die Modulendprüfung DAN+WI. Urheber dieses Werks sind u.A. Dr. Thomas Olnhoff, welcher die Aufgabenstellungen entworfen hat, Christoph Moser, Nicolas Nussbaumer und Thomas Galliker, welche die Antworten verfasst haben.

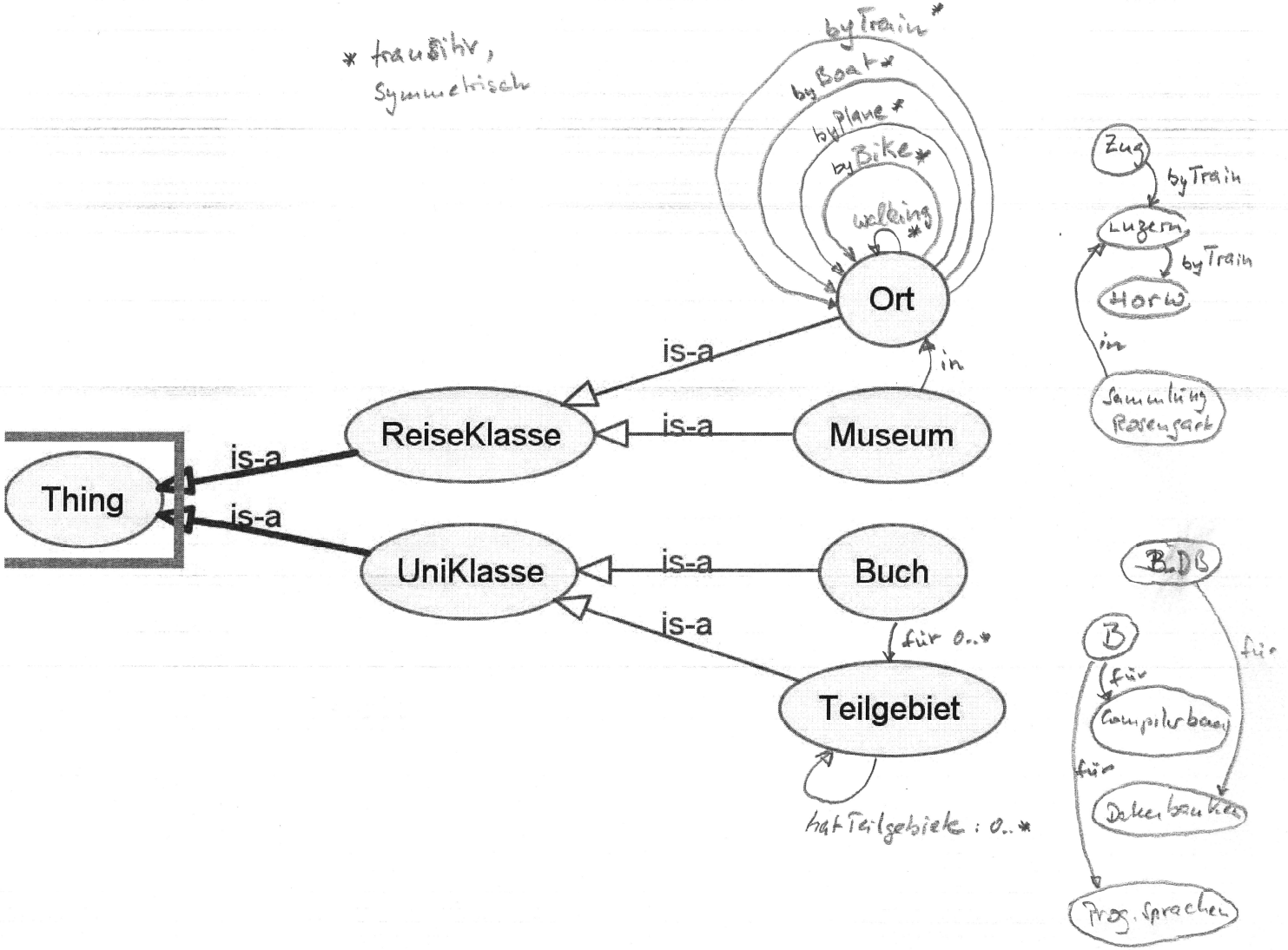
Teil 1: Semantic Web
(Regelzeit 30-35 Min je nach Semester)

Teil a) (25 Punkte)
Reisebeschreibungen sollen von einer OWL-Search-Engine gezielt gesucht werden können.
Jede einzelne Beschreibung hat in einer weltweit akzeptierten Ontologie Zusatzinformation (Metainformationen oder Annotationen) zu
- Besuchte Orte (es gibt einen globalen „Katalog“ aller Orte, die besuchten sind daraus)
- Besuchte Museen (Name) mit Ortsangabe
- Art des Fortkommens von einem Ort zum nächsten: walking (Wanderung), bicycle (Velo), train (Bahn), airplane (Flugzeug), boat (Schiff)

Geben Sie die informal (graphisch oder textuell) die OWL-Definitionen für diese Metainformationen.
Mit den Definitionen soll die Search-Engine auch folgende Resultate finden:
Wenn in einem Dokument d die Reisen per Bahn von Ort a nach Ort b und Ort b nach Ort c beschrieben sind, dann soll d auch gefunden werden bei einer Anfrage nach Bahnreisebeschreibungen
- von a nach c
- von c nach a.

Erweitern Sie die OWL-Definitionen auf der Instanzebene für eine Reisebeschreibung:
- Die Orte Zug, Luzern und Horw werden beschrieben.
- Eine Bahnreise wird beschrieben von Zug nach Luzern und weiter nach Horw
- Das Museum „Sammlung Rosengart“ in Luzern wird beschrieben.

Teil b) (5 Punkte)
Die Informatik hat Teilgebiete: Compilerbau, Datenbanken, Programmiersprachen. Informatik selber ist ein Teilgebiet. Es gibt Buch B für Compilerbau und Programmiersprachen. Es gibt Buch B_DB für Datenbanken.
Beschreiben Sie diese Informationen wieder (informal) mit OWL-Konzepten. Dass jedes Teilgebiet aus anderen Teilgebieten bestehen kann und dass Bücher für ein oder mehrere Teilgebiete relevant sein können: das muss auf Modellebene beschrieben sein.


* transitiv, symmetrisch
byTrain *
byBoat *
byPlane *
byBike *
walking *
Ort
is-a
in
ReiseKlasse
is-a
Museum
is-a
Thing
is-a
UniKlasse
is-a
Buch
is-a
für 0..*
Teilgebiet
hatTeilgebiete : 0..*
Zug
byTrain
Luzern
byTrain
Horw
in
Sammlung Rosengart
B.DB
B
für
Compilerbau
für
Datenbanken
für
Prog. Sprachen
Typ-Instanziierung "automatisch"

Teil 2: Data Warehouse

Regelzeit 15min, 15 Punkte

Ein internationales Handesunternehmen hat Datenbanken mit folgenden Tabellen:

- Produkt(Id, Bezeichnung, ProdukteGruppe, NormalPreis, ..)
- Lieferant(Id, Name, Ort, ..)

- Einkauf(Id, Lieferant, WunschLieferdatum, ..)
- EinkaufsPosition(Einkauf, Produkt, Anzahl, Lieferdatum, Preis, …)

- Einkauf verweist auf Lieferant.
- Einkaufsposition verweist auf Einkauf und Produkt.

Solche Datenbanken gibt es an vielen Orten der Welt (Verkaufsregionen des Unternehmens). Ein zentrales Data Warehouse soll globale Auswertungen ermöglichen über

- Anzahl und Preisentwicklung der eingekauften Produkte
- Liefertreue (Wunschlieferdatum – Lieferdatum >= 0 bedeuted gute Liefertreue)

Verdichtungen/ Auswahlen sollen auf Ebene Produkt, Produktgruppe, Lieferant, Wunschlieferdatum und Lieferdatum möglich sein. Diese Verdichtungen/ Auswahlen sollen beliebig kombinierbar sein.
Zu den Verdichtungen auf Datumsebene: Verdichtungen bezüglich Kalender-Halbjahren aber auch Finanzhalbjahren sollen „einfach“ möglich sein. Ein Finanzjahr kann z.B. von Anfang Oktober bis Ende September gehen (so in unserer Schule).
Definieren Sie das Schema eines geeigneten Data Warehouses (DW) (Tabellendefiniton wie oben genügen). (8 Punkte)

FactEinkauf(EinkaufId, PosNr, Lieferant, Wunschlieferdatum, Produkt, Anzahl, Lieferdatum, Preis) – Primärschlüssel sind zusammen EinkaufId und PosNr
DimProdukt wie oben Produkt (*, neuer Zusatz, siehe nächste Seite)
DimLieferant wie oben Lieferant (*, Erweiterungen entsprechend DimProdukt)
DimDatum(Datum, Jahr, .., KalenderHalbjahr, FinanzHalbjahr, ..) – Datum ist vom Typ Date, ein künstlicher Schlüssel (wie in DimTime von AdventureWorksDW) ist natürlich auch ok.

Von welcher Art ist Ihr DW-Schema (Stern- oder Schneeflockenschema)? (1 Punkt)
Sternschema

Geben Sie das select-statement, dass pro Produkt, Produktgruppe und Lieferant die durchschnittliche Liefertreue in allen Finanzhalbjahren aufzeigt. Die Verdichtungen über alle Kombinationen von Produkten, Produktgruppen, Lieferanten und Finanzhalbjahren sollen auch Teil des Ergebnisses sein. Insbesondere soll das „Grand Total“ Teilergebnis sein: die durchschnittliche Liefertreue über alle Produkte, Produktgruppen, Lieferanten und Finanzhalbjahre. (6 Punkte)

```
Select e.Produkt, p.ProdukteGruppe. e.Lieferant, d.FinanzHalbjahr,
avg(Wunschlieferdatum – Lieferdatum)
from (FactEinkauf e join DimProdukt p on e.Produkt = p.Id) join DimDatum d on
e.Lieferdatum = d.Datum group by cube(e.Produkt, p.ProdukteGruppe. e.Lieferant,
d.FinanzHalbjahr);
```

Kommentar von Hr. Olnhoff: *Ich nehme an, dass die Finanzhalbjahre seit Firmengründung hochgezählt sind (also nicht nur die Werte 1 oder 2 haben). Sonst würde man noch das Jahr ins Resultat nehmen*
(*) im letzten Jahr habe ich keine konkrete DW-DB vorgestellt. Dieses Jahr haben wir uns etwas genauer AdventureWorksDW angesehen. Zu den meisten Dim-Tabellen gibt es ein AlternateKey-Attribut und Attribute, die die zeitliche Gültigkeit eines Satzes festhalten. Mit dem AlternateKey kann der entsprechende aktuelle Satz in der operationalen DB gefunden werden. Aktualisierungen (updates) in der operationalen DB an Tabellen, die Dimensionstabelle entsprechen, ergeben neue Einträge im DW (mit neuen Gültigkeitsangaben). Für ein Produkt (bestimmt durch den AlternateKey) gibt es deshalb mehrere Einträge in DimProdukt, wenn der Preis sich ändert. Sie können so sehr effizient anhand der DimProdukt-Tabelle die Preisentwicklung verfolgen. Ich hatte diesbezüglich keine konkrete Frage gestellt. Anhand der FactEinkauf-Tabelle ist es auch möglich. Hier ein entsprechendes select-statement (in DimProdukt sind keine(!) zusätzlichen Einträge bei Preisänderungen):

```
Select e.Produkt, d.Jahr, d.Monat, avg(Preis/Anzahl)
from FactEinkauf e join DimDatum d on e.Lieferdatum = d.Datum
group by e.Produkt, d. Monat;
```

Vergleich der monatlichen Durchschnittspreise.
Wenn DimProdukt selber die Preisänderungen enthält, dann könnte man die Preisentwicklung immer noch via FactEinkauf ermitteln (allerdings wäre das nicht mehr ratsam!).

```
Select p.ProduktAlternateKey, d.Monat, avg(Preis/Anzahl)
from (FactEinkauf e join DimProdukt p on e.Produkt = p.Id) join DimDatum d on
e.Lieferdatum = d.Datum group by p.ProduktAlternateKey, d.Monat;
```

Beachten Sie auch, dass ich oben den cube-operator nicht benutzt habe (Zusatzinformation durch den cube-operator macht keinen Sinn).

Teil 3: Data Mining

Regelzeit 60 Min

A) Im vorgehenden Data Warehouse seien auch die Verkäufe mit erweiterten Kundendaten erfasst. Die Verkaufsdaten sind analog zu den Einkaufsdaten in den Tabellen Verkauf und VerkaufsPosition strukturiert. Statt der Referenz zu Lieferant, gibt es die Referenz zu Kunde. „Erweiterte Kundendaten" haben ausser Name und Adresse noch andere Attribute: Geschlecht, Alter, Einkommen, AnzahlKinder, …

A1) welche „klassischen" Analysen sind dann möglich? Mit welchen Verfahren (Algorithmen)? Zwei unterschiedliche Analysen, die ganz unterschiedliche Zusammenhänge herausfinden sollen, genügen für die volle Punktzahl. Und bei den möglichen Algorithmen für jede Analyse: beschränken Sie sich auf die, mit denen wir uns auch beschäftigt haben. (7 Punkte)

- Klassifikationsverfahren nach z.B. Produktegruppen (z.B. mit Entscheidungsbäumen, J48-Algorithmus)
- Assoziationsverfahren (Warenkorbanalyse), z.B . mit Apriori-Algorithmus

A2) Sind die Daten für die obigen Analysen/ Algorithmen passend aufbereitet? Wenn nicht: wie sieht eine „optimale" Aufbereitung aus? (3 Punkte)

Die Daten sollen in einer Satzstruktur vorliegen. Für die Klassifikation nach Produktegruppen braucht es deshalb pro FactVerkauf-Eintrag die Gruppe des Produktes und die Attribute des Kunden.
Für die Warenkorbanalyse gibt es eine neue Struktur: für jede Produktgruppe ein Attribut. Pro Verkauf(Id) gibt es einen solchen Satz. Für jedes Produkt in dem Verkauf (Warenkorb) wird die entsprechende Produktgruppe auf true gesetzt .In Weka (arff-Format) ist das der einzige, definierte Attributwert, sonst ist nur noch undefined zulässing (? in Weka).

B) Analysieren Sie mit Weka anhand des Datenbestandes customers.arff, welche Kriterien für einen Wechsel des Telekom-Anbieters relevant sind. Ein Kunde (customer) hat den Telekom-Anbieter gewechselt, wenn churn=1 ist. Zielattribut der Untersuchungen ist churn! Es ist nicht das letzte Attribut – Weka nimmt als default das letzte Attribut. Achten Sie, bitte, darauf.
Ihre Analyse basiert sicherlich auf Weka Output: kopieren Sie die wichtigen Output-Daten in eine Datei, erweitern Sie sie um die eigene Analyse. Geben Sie Ihre Lösung in einem entsprechenden PDF-File ab (Vorschlag).

B1) Wieviele Wechsler und Nicht-Wechsler gibt es in dem Datenbestand? (2 Punkte)
Weka Daten „customers.arff" laden. Datensatz einschränken auf Attribut „churn". Danach Clusteranaylse mit 2 Cluster auf Attribut „churn":

```
                         Cluster#
Attribute    Full Data          0          1
                (5000)     (3734)     (1266)
=========================================
churn                0          0          1


Clustered Instances

0       3734 ( 75%)
1       1266 ( 25%)
```

B2) Entscheidungsbaum J-48 (Classify –> trees)
Analysieren Sie mit einem Entscheidungsbaum-Algorithmus. (7 Punkte)
Schreiben Sie die Formel zur Berechnung der Gain-Ratio für das Attribut marital (verheiratet ja/nein). Die notwendigen Werte finden Sie sehr einfach in Weka! (4 Punkte dazu)

Customers Daten in Weka laden. Danach unter Preprocess die für diese Analyse unwichtigen Attribute entfernen:

- Pets..
- Townsize

Analyse mit Klassifizierungsalgorithmus J48 (J48 –C 0.25 –M 2) und dem Attribut „churn“ starten. (Das Resultat variiert je nach Auswahl der Attribute).

```
Correctly Classified Instances        3768               75.36    %
Incorrectly Classified Instances      1232               24.64    %
Kappa statistic                          0.2197
Mean absolute error                      0.3254
Root mean squared error                  0.4419
Relative absolute error                 86.0289 %
Root relative squared error            101.6215 %
Coverage of cases (0.95 level)          95.88   %
Mean rel. region size (0.95 level)      95.55   %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.92      0.736      0.787     0.92      0.848      0.64     0
                 0.264     0.08       0.527     0.264     0.352      0.64     1
Weighted Avg.    0.754     0.57       0.721     0.754     0.722      0.64

=== Confusion Matrix ===

    a    b   <-- classified as
 3434  300 |    a = 0
  932  334 |    b = 1
```

TP Rate = als a klassifiziert/ a Instanzen 3434/3734 = 0.92
FP Rate = alle anderen klassifiziert als a/ keine a Instanzen 932/(5000-3434+300) = 0.736

Fazit:
Nicht-Wechsler (0) können relativ gut ermittelt werden. Wechsler (1) jedoch nicht.

Die Kappa Statistik ist mit 21.97% eher schlecht. (In den Unterlagen werden zur Ermittlung der Kappa Statistik zwei Modelle verwendet, beim obigen Beispiel wird jedoch lediglich eines verwendet, wie wird daher der Wert berechnet.)
Zur Interpretation genügt es zu wissen, dass 1 (100%) der optimale Wert wäre, kleinere Werte als 0.5 (50%) als eher schlecht anzusehen sind. Die Zuordnung nach einem ramdom-Verfahren üben wir nicht - bleibt Weka-Geheimnis.

Die Error-Rate ist mit 101.6215% schlecht.

Relevante Attribute nach Entscheidungsbaum:

- Internet
- age
- employ

Eher nicht-relevante Attribute:

- pets-dogs, townsize

Immer auf Nicht-Wechsler "tippen" – würde das ähnlich gut sein?
Ja diese Analyse ist gleich gut, wie wenn man immer auf Nicht-Wechsler tippen würde.
Den 3734 Personen also 75% von 5000 Personen sind Nicht-Wechsler. Diese Analyse weisst die Person zu 75.36 % der korrekten Klasse zu.

Wie findet man die notwendigen Werte?

Die folgende Abfrage liefert die gewünschten Werte zurück:
Select marital, churn, count(*) from customer group by marital, churn

Oder

In Weka mit einem SimpleKMeans Cluster welcher nur die Attribute churn und martial als Input erhält. Vier Cluster erstellen lassen → numCluster:4

```
                         Cluster#
Attribute    Full Data          0          1          2          3
                (5000)     (1885)     (1849)      (552)      (714)
=============================================================================
marital              0          0          1          1          0
churn                0          0          0          1          1


Clustered Instances

0      1885 ( 38%)
1      1849 ( 37%)
2       552 ( 11%)
3       714 ( 14%)
```

Informationsmenge und Gain berechnen

Formel: Gain (Martial) = $Info(Kundenwechsel) - Info_{Martial}(Kundenwechsel)$ =
$0.566 - 0.5645 = \underline{0.0015}$

→ $Info(Kundenwechsel) = -p1 * log(p1) - p2 * log(p2)$ =
$-1266/5000 * log(1266/5000) - 3734/5000 * log(3734/5000) = \underline{0.566}$

Es gibt 2599 nicht-verheiratete (martial=0), von diesen sind x1 Wechsler (churn=1) und x2 keine Wechsler (churn=0)
(x2 = 2599 – x1)
→ x1 = 1885
→ x2 = 714

Es gibt 2401 verheiratete (martial=1), von diesen sind y1 Wechsler (churn=1) und y2 keine Wechsler (churn=0)
(y2 = 2401 – y1)
→ y1 = 552
→ y2 = 1849

$Info(Kundenwechsel_{Martial=0}) = (-x1/2599 * log(x1/2599) - x2/2599\ log(x2/2599)) = 0.588$
$Info(Kundenwechsel_{Martial=1}) = (-y1/2401 * log(y1/2401) - y2/2401 log(y2/2401)) = 0.539$

$Info_{Martial}(Kundenwechsel)$ =
$2599/5000 * Info(Kundenwechsel_{Martial=0}) +$
$2401/5000 * Info(Kundenwechsel_{Martial=1})$ =
$2599/5000 * (-1885/2599 * log(1885/2599) - 714/2599\ log(714/2599)) +$
$2401/5000 * (-552/2401 * log(552/2401) - 1849/2401 log(1849/2401)) = \underline{0.5645}$

B3) Naive Bayes (Klassifizierungsaufgabe)
Geben Sie die Wahrscheinlichkeiten für jeden Wert des Attributes internet (0..4). Sie finden sie wieder sehr einfach in Weka! Der arithmetische Ausdruck genügt mir – kein Rechnen!
(3 Punkte dazu)
Customers Daten in Weka laden. Danach unter Preprocess die für diese Analyse unwichtigen Attribute entfernen.
Analyse mit Klassifizierungsalgorithmus NaiveBayes (Cross-validation mit 10 Folds) und dem Attribut „churn“ starten. (Das Resultat variiert je nach Auswahl der Attribute).

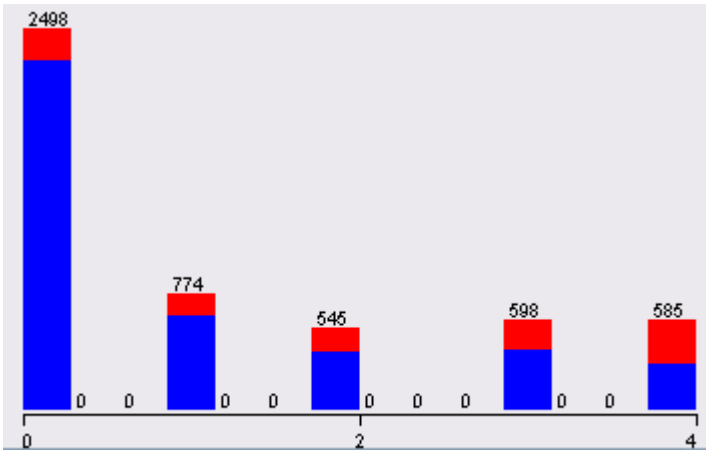
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3532               70.64    %
Incorrectly Classified Instances      1468               29.36    %
Kappa statistic                          0.3224
Mean absolute error                      0.3137
Root mean squared error                  0.4803
Relative absolute error                 82.9424 %
Root relative squared error            110.443  %
Coverage of cases (0.95 level)          89.8    %
Mean rel. region size (0.95 level)      74.05   %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.729     0.359      0.857     0.729      0.788      0.739    0
                 0.641     0.271      0.445     0.641      0.525      0.739    1
Weighted Avg.    0.706     0.337      0.752     0.706      0.721      0.739

=== Confusion Matrix ===

    a    b   <-- classified as
 2721 1013 |    a = 0
  455  811 |    b = 1
```

Es wurden weniger Nicht-Wechsler korrekt klassifiziert (2721). Die Wechsler (1) wurden hier besser klassifiziert (0.641). Der Root relavtive squared error ist höher als bei J48.

Die Kappa Statistik ist in dieser Analyse besser (0.322), als beim Entscheidungsbaum mit J48, allerdings immer noch ziemlich schlecht.

Fazit:
Der Naive Bayes Algorithmus ist für diese Analyse nicht zu gebrauchen.
Sie haben eine Methode als die beste (unter leider schlechten) begründet. Das ist mir wichtig.
Ist obige Antwort demnach korrekt?
>>Ja, aber auch nicht viel schlechter als andere!!

Wahrscheinlichkeitswerte für Attribut „internet“ berechnen
P(internet=0) = 2498/5000 = 0.4996
P(internet=1) = 774/5000 = 0.1548
P(internet=2) = 545/5000 = 0.1090
P(internet=3) = 598/5000 = 0.1196
P(internet=4) = 585/5000 = 0.1170



Sind hier wirklich nur die einfachen Wahrscheinlichkeitswerte gefragt? Oder müssten nicht besser die Wahrscheinlichkeiten in Abhängigkeit mit Wechsler/Nichtwechsler berechnet werden?
Also: P(internet=0 | churn=1) = … P(internet=1 | churn=0) = … usw..

B4) SimpleKMeans (Cluster)

Versuch 1:

churn	0	0	0
callcard	1	1	1
wireless	0	0	0
multline	0	1	0
voice	0	0	0
pager	0	0	0
internet	1.1996	1.3045	1.1171
hourstv	19.645	19.405	19.8336
ownpda	0	0	0
ownipod	0	0	0

```
Clustered Instances

0       2200 ( 44%)
1       2800 ( 56%)
```

Mit „Use Training Set“ wurden 2200 und 2800 Instanzen geclustert. In beiden Klassen ist das churn Attribut jedoch 0, deshalb können mit dieser Analyse die Nicht-Wechsler sowie Wechsler nicht ermittelt werden.

Versuch 2:

```
Clustered Instances

0       2200 ( 44%)
1       2800 ( 56%)


Class attribute: churn
Classes to Clusters:

    0    1  <-- assigned to cluster
 1621 2113 | 0
  579  687 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0

Incorrectly clustered instances :       2308.0    46.16   %
```

Mit dem Property „Classes to cluster evaluation” wurden ebenfalls 2200 und 2800 Instanzen geclustert. 46% wurden dabei falsch geclustert. Auch diese Methode bringt nicht die erwünschen Resultate.

Versuch 3:
Auch bei 4 Clustern bleibt das Attribut churn = 0.

Wie ernst muss man die Aussage nehmen: churn=0 in beiden Clusters? Sie haben doch nur 2 Clusters – richtig? -> Was meinen Sie damit?
>> es gibt auch churn=1. Entsprechende Sätze müssen auch eingeflossen sein. Aber die meisten (nicht alle) hatten churn=0. Egal welches cluster: churn=0 scheint die richtige Wahl – damit sind die beiden Cluster tatsächlich (so Ihre Aussage) nicht hilfreich.

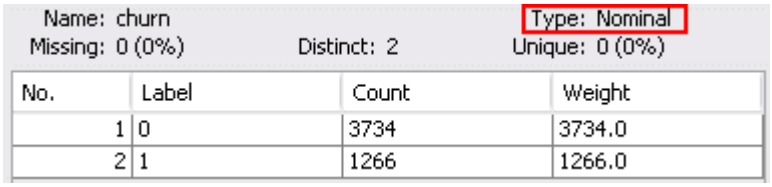
Es gibt wahrscheinlich zu wenig Datensätze mit churn 1, deswegen haben diese keinen grossen Einfluss der Clusterbildung mit wenigen Cluster. Nimmt man beispielsweise 20 oder 30 Cluster, gelingt es dem SimpleKMeans tatsächlich auch Cluster zu bilden, welche churn=1 enthalten.

B5) LinearRegression (Classify -> functions)
CustomersLinRegression Daten in Weka laden. Danach unter Preprocess die für diese Analyse unwichtigen Attribute entfernen.
Analyse mit Klassifizierungsfunktion LinearRegression und dem Attribut „churn“ starten. (Das Resultat variiert je nach Auswahl der Attribute). Folds auf 2 beschränken.

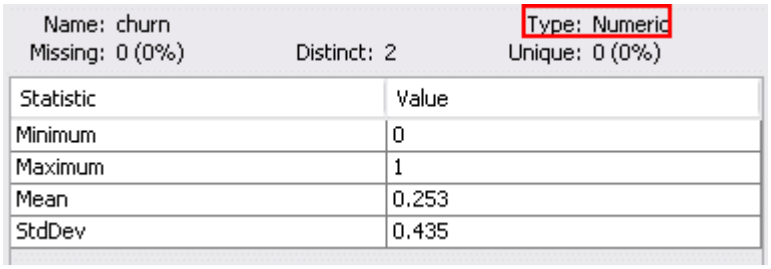
```
Time taken to build model: 27.13 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.3896
Mean absolute error                      0.3241
Root mean squared error                  0.4013
Relative absolute error                 85.7043 %
Root relative squared error             92.291  %
Total Number of Instances             5000
```

Der Korrelations-Koeffizient (0.3896) nahe bei 0 ist weisen die Merkmale einen geringen Zusammenhang auf. Eine Korrelation unter 0.5 (höher -0.5) ist schlecht.

Zusatzaufgabe
Welche Änderung bezgl. customers.arff muss gemacht werden und warum? (2 Punkte dazu)
→Die Linear Regression Analyse kann nur mit nummerischen Zielattributen durchgeführt werden. Vergleiche die folgenden „churn“ Typen:


Name: churn Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	0	3734	3734.0
2	1	1266	1266.0


Name: churn Type: Numeric
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

Statistic	Value
Minimum	0
Maximum	1
Mean	0.253
StdDev	0.435

B6) Multilayer Perceptron (Classify -> functions)

Customers Daten in Weka laden. Danach unter Preprocess die für diese Analyse unwichtigen Attribute entfernen.
Analyse mit Klassifizierungsfunktion „Multilayer Perceptron“ (Cross-validation mit 2 Folds) und dem Attribut „churn“ starten. (Das Resultat variiert je nach Auswahl der Attribute).

Achtung: Das Erstellen eines Modells dauert bei grossen Datenmengen (5000 Datensätze mit je 47 Attributen) extrem lange. Deshalb empfiehlt es sich bei Multilayer Perceptron Analysen jeweils die Daten in beiden erwähnten Dimensionen zu verkleinern.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         368               66.426  %
Incorrectly Classified Instances       186               33.574  %
Kappa statistic                          0.167
Mean absolute error                      0.3388
Root mean squared error                  0.5323
Relative absolute error                 83.282  %
Root relative squared error            118.1163 %
Coverage of cases (0.95 level)          82.6715 %
Mean rel. region size (0.95 level)      66.6968 %
Total Number of Instances              554

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.771     0.605      0.763     0.771     0.767      0.641    0
                 0.395     0.229      0.405     0.395     0.4        0.641    1
Weighted Avg.    0.664     0.499      0.662     0.664     0.663      0.641

=== Confusion Matrix ===

   a   b   <-- classified as
 306  91 |   a = 0
  95  62 |   b = 1
```

Es wurden lediglich 368 von 554 Instanzen korrekt klassifiziert. Die Kappa Statistik ist mit 16.7% schlecht. Error-Rate ist mit 118.1163% sehr hoch und damit ebenfalls schlecht. Verglichen mit dem Naive Bayes Algorithmus ist TP Rate von 77.1% ziemlich ähnlich.

Es wurden 66% korrekt Klassifiziert. Wenn man immer davon ausgehen würde, dass es sich um einen Nicht-Wechsler handelt, dann würde man mit 72 % korrekt klassifizieren.

Settings Multilayer Perceptron:

"trainingTime" parameter is the number of epochs (iterations) that the back propagation algorithm performs.

The **hidden layers** parameter can take a comma separated list of integers, or some special characters in order to define the number of nodes in the hidden layers. The special character are:

'a' = (attribs + classes) / 2, 'i' = attribs, 'o' = classes , 't' = attribs + classes.

So for example you could supply "a, 6", which would give you (attribs + classes) / 2 nodes in the first hidden layer and 6 in the second.

The thresholds are actually bias terms. I.e. like the intercept in a linear regression.

Zusatzaufgabe

Zeitaufwändige Analyse:

- Anzahl Attribute und die Beziehung mit den Hidden-Layers
- Einzelnen Gewichte der Kanten werden bei jedem Durchlauf angepasst bis Zielwert == Erwartungswert
- Die Lernrate spielt eine wichtige Rolle, mit ausreichendem experimentieren kann der optimale Wert herausgefunden werden. Ist sehr Fall-abhängig

Mit welchen Eigenschaften kann Analyse beschleunigt werden:

- Anzahl Attribute (Input-Werte) verkleinern. Je weniger Attribute umso schneller.
- Anzahl Datensätze verkleinern.
- Lernrate ? (kleinere Lernrate länger, da kleinere Sprünge) ??
- HiddenLayers