Diplomarbeit

zum Thema

Diskriminanzanalyse

zur Erlangung des akademischen Grades

Diplom-Informatikerin (FH)

vorgelegt am
Fachbereich Mathematik, Naturwissenschaften und Informatik
der Fachhochschule Gießen-Friedberg

von
Lena-Luisa Heß

im Dezember 2003

Referent: Prof. Dr. O. Hoffmann
Korreferent: Prof. Dr. A. Laun

Inhaltsverzeichnis

Kapitel 1

Einleitung

Als ich anfing, mich mit dem Thema *Diskriminanzanalyse* zu beschäftigen, nahm ich mir erst einige Lexika zur Hand, um mir einen Überblick zu verschaffen, worum es überhaupt geht.
So findet man z.B. in Meyers großem Taschenlexikon in 24 Bänden ([Mey90]) zum Thema Diskriminanzanalyse:

> Diskriminanzanalyse [lat./griech.]
> (Unterscheidungsanalyse), Verfahren der analytischen Statistik
> Sind von zwei oder mehr statistischen Grundgesamtheiten Stichproben bekannt, so liefert die Diskriminanzanalyse optimale Trennkriterien in Form von bestimmten linearen Funktionen der Stichprobenwerte (sog. Diskriminanzfunktionen), die es erlauben, dass weitere Stichproben einer dieser Grundgesamtheiten mit bestimmten Wahrscheinlichkeiten zugeordnet werden können.

Bei der Lektüre war mir auch aufgefallen, dass sie ein *"multivariates statistisches Verfahren"* ist.

Damit ließ sich das Thema wenigstens eingrenzen: Es geht um ein *"multivariates statistisches Verfahren"*, bei dem versucht wird, aus einer bereits in Gruppen eingeteilten Datenmenge eine Funktion zu berechnen, die die Daten möglichst gut wieder diesen Gruppen zuordnet – und die später auch neue, noch ungruppierte Daten in diese Gruppen einordnen kann.

Doch was sind "multivariate statistische Verfahren"?? Gibt es noch mehr als nur die Diskriminanzanalyse? Wenn ja: Wie unterscheiden sie sich von der Diskriminanzanalyse und was tun sie?

Dieser Frage widme ich mich im 2. Kapitel. Dort gebe ich einen Überblick über die multivariaten Verfahren. Nachdem ich hierbei über zwei mathematische Probleme (die *Matrix-Invertierung* und das *Eigenwert-Problem*) "gestolpert" bin, beschreibe ich diese in einem eigenen Unterkapitel und gehe danach auf die *Diskriminanzanalyse* ein. Dabei gehe ich auf die theoretischen Grundlagen getrennt von der mathematischen Seite ein.

Abschließend vergleiche ich die Diskriminanzanalyse noch mit *neuronalen Netzen*, weil beide Verfahren zu den (selbst-) lernenden Verfahren gehören.

Nachdem die theoretischen Fragen bis dahin geklärt sein sollten, gilt das 3. Kapitel der praktischen Umsetzung. Ich beschreibe kurz meine Wahl der Entwicklungsumgebung, bevor ich auf den Entwurf bzw. den Aufbau der Software eingehe. Anschließen zeige ich die beiden Umsetzungen der mathematischen Probleme, bevor ich einige Worte zur Online-Hilfe verliere. Außerdem beschreibe ich noch die Testphase und meinen Versuch, das Programm nach Linux zu portieren. Das Ende dieses Kapitels bildet die Beschreibung von zwei Anwendungsbeispielen, die die Probleme der Diskriminanzanalyse zeigen.

Zum Ende der Arbeit gebe ich eine Zusammenfassung über das, was meiner Meinung nach in meinem Programm gelungen ist – und was nicht. Welche Teile meines "Wunsch-Programms" realisiert wurden – und warum die anderen Teile nicht umgesetzt wurden.
Einige Fachbegriffe, die nicht unbedingt geläufig sind, werden im Glossar (ab Seite 73) erklärt.

Kapitel 2

Mathematischer Hintergrund und Problemstellung

2.1 Multivariate Verfahren

Die multivariaten Verfahren sind durch eine gemeinsame, gleichzeitige Analyse mehrerer Merkmale bzw. deren Ausprägungen gekennzeichnet. Im Gegensatz zu den univariaten Verfahren können hier auch die Abhängigkeiten zwischen den Merkmalen berücksichtigt werden. Es gibt zwei Arten von multivariaten Verfahren:

- Gruppen-bildende Verfahren (z.B. Regressionsanalyse)
  Die noch nicht in Gruppen eingeteilten Daten werden aufgeteilt.
- Objekt-klassifizierende Verfahren (z.B. Diskriminanzanalyse)
  Die bereits in Gruppen eingeteilten Daten werden untersucht, um mit Hilfe der Ergebnisse später neue Datensätze in diese Gruppen einordnen zu können.

2.1.1 Kurze Beschreibung einzelner Verfahren

Hier folgt eine Liste, in der ich eine Auswahl multivariater Verfahren beschreiben möchte. Diese Beschreibungen sollen nur einen Überblick geben, worauf es bei den verschiedenen Methoden ankommt. Die Liste erhebt keinen Anspruch auf Vollständigkeit!

2.1.1.1 AID-Analyse (Kontrastgruppen- oder Baumanalyse)

AID = **A**utomatic **I**nteraction **D**etector
Das Ziel der AID-Analyse ist die Beschreibung einer abhängigen Variablen. Hierzu werden die Ausgangsdaten nach und nach in Gruppen aufgeteilt, die bestimmte Merkmalskombinationen aufweisen; dabei bringt jedes neue Merkmal neue Informationen zur Beschreibung der abhängigen Variablen.

2.1.1.2 Clusteranalyse

Die Clusteranalyse beschäftigt sich mit

- der Erkennung von Strukturen in einer Menge von Objekten – und
- der Klassifikation von Objekten.

Die Daten werden so in Gruppen eingeteilt, dass Objekte, die zur selben Gruppe gehören, einander möglichst ähnlich sind. Außerdem sollen Objekte, die unterschiedlichen Gruppe zugeordnet wurden, möglichst unterschiedlich sein.

2.1.1.3 Conjoint-Measurement

Das Ziel des Conjoint-Measurements ist es, den Anteil vorgegebener Nutzen-Attribute am Gesamtnutzen eines Objektes zu bestimmen. Es dient der Messung psychologischer Werturteile und ermittelt den Beitrag einzelner Faktoren zu einem gegebenen Gesamturteil.

2.1.1.4 Diskriminanzanalyse

Ein Ziel der Diskriminanzanalyse ist die Trennung einer Anzahl von Personen / Fällen / Beobachtungen in verschiedene Untergruppen aufgrund des Einflusses mehrerer unabhängiger Variablen. Anders als bei der Regressionsanalyse (s. S. 9) ist die abhängige Variable hier nicht stetig, sondern stellt Gruppen dar.
Durch die resultierende Diskriminanzfunktion kann folgende Frage beantwortet werden: ''In welche Gruppe ist ein neues Element, dessen Gruppenzugehörigkeit nicht bekannt ist, aufgrund seiner Merkmalsausprägung einzuordnen?". Darin zeigt sich das andere Ziel der Diskriminanzanalyse: Die Zuordnung neuer Elemente hit Hilfe der Diskriminanzfunktion.

2.1.1.5 Faktorenanalyse

Die Faktorenanalyse versucht, eine Menge mit vielen Merkmalen zu einer Menge mit weniger Merkmalen zusammenzufassen. Dabei wird versucht, die Merkmale auf einige, wenige „künstliche“ Merkmale / Faktoren zurückzuführen, die selber nicht messbar sind, aber die Gesamtdatenmenge besser beschreiben können als die ursprüngliche Vielzahl von Merkmalen; z.B. Führungsqualität, charakterliche Eignung, ...

2.1.1.6 Kontingenzanalyse

Die Kontingenzanalyse untersucht die Zusammenhänge zwischen nominalskalierten Variablen.
Die 'Instrumente" der Kontingenzanalyse sind der χ^2-Test und der Φ-Koeffizient.

2.1.1.7 Kovarianzanalyse (ANCOVA)

ANCOVA = **An**alysis of **Cova**riances
Die Kovarianzanalyse entspricht einer Varianzanalyse (s. S. 9) mit vorgeschalteter Regressionsanalyse(s. S. 9). Man will damit den Einfluss der zusätzlich eingeführten Kovariante herausfinden. Die Kovariante ist eine metrisch skalierte unabhängige Variable.

2.1.1.8 Korrelationsanalyse

Mit der Korrelationsanalyse wird eine Angabe über das Vorhandensein und die Stärke von Abhängigkeiten gemacht. Dabei entspricht die Korrelation der Assoziation von Merkmalen also dem Grad des linearen Zusammenhangs zwischen den Merkmalen.
Es gibt drei Typen der Korrelationsanalyse:

1. einfache Korrelation
   Untersucht wird der Zusammenhang zwischen zwei Merkmalen

2. multiple Korrelation
   Untersucht wird der Zusammenhang zwischen einem Merkmal und einer Gruppe von Merkmalen

3. qualitative Korrelation
   Untersucht wird der Zusammenhang zwischen zwei Gruppen von Merkmalen

Bei allen Berechnungen ist jedoch darauf zu achten, das ein sachlogischer Zusammenhang existiert. Man sollte also nicht versuchen, etwas über den Zusammenhang zwischen der Anzahl von Störchen und der menschlichen Geburtenzahl zu finden oder zwischen der Schuhgröße des Kellners und der zu erwartenden Höhe an Trinkgeld.
Auf der Korrelationsanalyse beruhen Faktorenanalyse, partielle Korrelation und Regressionsanalyse.

2.1.1.9 LISREL-Analsye

LISREL = **Li**near **S**tructural **Rel**ationship
LISREL ist ein Computer-Programm zur Überprüfung von komplexen Kausalstrukturen.

2.1.1.10 Mehrdimensionale Skalierung (MDS)

Mit der mehrdimensionalen Skalierung versucht man die Positionierung von Objekten im Wahrnehmungsraum von Personen. Verwendet werden nur wahrgenommene globale Ähnlichkeiten zwischen Untersuchungsobjekten. Die Merkmalsausprägungen finden hier keine Verwendung! Sie wird häufig verwendet, wenn keine oder nur vage Informationen darüber vorhanden sind, welche Eigenschaften für die subjektive Beurteilung von Objekten relevant sind.

2.1.1.11 (Multiple) Varianzanalyse ((M)ANOVA)

(M)ANOVA = (**M**ultiple) **An**alysis **o**f **Va**riance
Die Varianzanalyse ist eine der allgemeinsten statistischen Analysemethoden. Sie ist ein Mittelwerttest für mehrere Stichproben. Die Varianz der zusammengefasst betrachteten Gruppen wird mit der Varianz innerhalb der einzelnen Gruppen in Beziehung gesetzt. Die abhängige Variable muss intervallskaliert sein; die unabhängige Variable i.d.R. nominalskaliert. Das Ziel ist die Klärung der Frage, ob sich die Mittelwerte einer oder mehrerer abhängiger Variablen für Gruppen von Fällen, verursacht durch eine unabhängige Variablen (ANOVA) oder mehrere unabhängige Variablen (MANOVA), signifikant unterscheiden. Zu unterscheiden sind:

- ANOVA:
  - *einfaktorielle* Varianzanalyse: Einfluss einer abhängigen Variablen auf eine unabhängige Variable; es werden Rückschlüsse auf die Grundgesamtheit gemacht
  - *mehrfaktorielle* Varianzanalyse: Einfluss mehrerer abhängiger Variablen auf eine unabhängige Variable
- MANOVA:
  - *mehrdimensionale* Varianzanalyse: Einfluss mehrerer abhängiger Variablen auf mehrere unabhängige Variable; die mehrdimensionale Varianzanalyse ist *keine* Hintereinander-Ausführung von mehrfaktoriellen Varianzanalysen, da die abhängigen Variablen auch untereinander voneinander abhängen können.

2.1.1.12 Regressionsanalyse

Die Regressionsanalyse gehört zwar nicht zwangsläufig zu den multivariaten Verfahren, spielt dabei aber eine so große Rolle, dass sie nicht unerwähnt bleiben sollte. Durch ihre Flexibilität ist sie sowohl für die Erklärung von Zusammenhängen wie auch für die Durchführung von Prognosen verwendbar.
Sie untersucht die Abhängigkeit metrischer Variablen, wodurch auch die Unterschiede zu erklären sind:

- *einfache lineare* Regression: Abhängigkeit einer abhängigen Variaben von einer anderen unabhängigen Variablen
- *multiple lineare* Regression: Abhängigkeit einer abhängigen Variaben von mehreren anderen unabhängigen Variablen
- *multivariate lineare* Regression: Abhängigkeit mehrerer abhängiger Variaben von mehreren anderen unabhängigen Variablen. Auch hier kann man - wie bei der multivariaten Varianzanalyse (s.o.) - nicht einfach mehrere multiple lineare Regressionen berechnen, weil die abhängigen Variablen untereinander zuammenhängen können.

- *einfache nicht-lineare* Regression: nichtlineare Abhängigkeit einer abhängigen Variablen von einer anderen unabhängigen Variablen

2.1.2 Einteilung Multivariater Verfahren

Bei der Einteilung der multivariaten Verfahren gibt es verschiedene Ansätze. Ein möglicher Ansatz ist die Unterscheidung danach, ob die Verfahren Strukturen geben (z.B. Regressionsanalyse) oder diese nur untersuchen (z.B. Diskriminanzanalyse). Dabei überprüfen die strukturprüfenden Verfahren die Zusammenhänge zwischen Variablen, wenn über den Zusammenhang vor Anwendung der Analyse schon Hypothesen existieren. Bei den strukturegbenden Verfahren liegen noch keine Hypothesen vor.

Strukturprüfende Verfahren	Strukturgebende Verfahren
Regressionsanalyse	Faktoranalyse
Varianzanalyse	Clusteranalyse
Diskriminanzanalyse	Mehrdimensionale Skalierung
LISREL	AID-Analyse
Conjoint-Measurement	

Tabelle 2.1: Einteilung multivariater Verfahren (1)

Eine weitere Möglichkeit der Einteilung besteht darin, die beobachteten Merkmale zu betrachten: sind alle Merkmale, die in die Berechnung mit einfließen, gleichberechtigt (Dependenzanalyse) oder gibt es Merkmale, die wichtiger sind als andere (Interdependenzanalyse)?

Dependenzanalyse	Interdependenzanalyse
Varianzanalyse	Clusteranalyse
Kovarianzanalyse	Faktoranalyse
Regressionsanalyse	Mehrdimensionale Skalierung
Diskriminanzanalyse	Korrelationsanalyse
AID-Analyse	Kontingenzanalyse
Conjoint-Measurement	

Tabelle 2.2: Einteilung multivariater Verfahren (2)

Außerdem kann man die Verfahren danach unterscheiden, ob sie Unterschiede oder Zusammenhänge analysieren.

Unterschiede zwischen Stichproben	Zusammenhänge zwischen Variablen
Varianzanalyse	Korrelationsanalyse
Kovarianzanalyse	Regresionsanalyse
Diskriminanzanalyse	Faktorenanalyse
Clusteranalyse	

Tabelle 2.3: Einteilung multivariater Verfahren (3)

Für die strukturprüfenden Verfahren kann man noch eine differenziertere Aufteilung treffen, je nach Art der Variablen:

		unabhängige Variable(n)	
		metrisch	nominal
abhängige Variable(n)	metrisch	Regressionsanalyse	Varianzanalyse
	nominal	Diskriminanzanalyse	Kontingenzanalyse

Tabelle 2.4: Einteilung strukturprüfender Verfahren

2.2 Ausgewählte mathematische Probleme

2.2.1 Matrix-Invertierung

Bei der Matrix-Invertierung ist darauf zu achten, dass die zu invertierende Matrix A quadratisch ist und ihre Determinante ungleich 0 ist.
Danach lässt sich die zu A inverse Matrix A^{-1} durch das Gauß-Jordan-Verfahren berechnen:

1. Man bildet eine neue Matrix, indem man an die zu invertierende Matrix A die Einheitsmatrix (Diagonal-Elemente = 1, Rest = 0) E anhängt.

$$(A|E) = \left(\begin{array}{cccc|cccc} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{array}\right)$$

2. Durch elementare Zeilenumformungen (Vertauschen von Zeilen, Multiplikation einer Zeile mit einem Skalar $\neq 0$, Addition des Vielfachen einer Zeile zu einer anderen Zeile) wird B nun so umgeformt, dass E den ursprünglichen Platz von A einnimmt. Die gesuchte Matrix A^{-1} befindet sich auf dem ursprünglichen Platz der Einheitsmatrix.

$$(E|A^{-1}) = \left(\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & 1 & \dots & 0 & a'_{21} & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & a'_{n1} & a'_{n2} & \dots & a'_{nn} \end{array}\right)$$

Sollte die Matrix nicht invertierbar sein, weil entweder die ursprüngliche Matrix nicht quadratisch ist oder sie die Determinante 0 hat, ist es nicht möglich, die Einheitsmatrix auf der linken Seite zu rekonstruieren.
Außerdem sollte man das Vertauschen von Spalten zulassen und anwenden, um zuverlässiger zu Ergebnissen zu kommen.

Ein Beispiel zur Verdeutlichung:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 9 \end{pmatrix}$$

$$\begin{aligned}
(A|E) &= \left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 4 & 0 & 1 & 0 \\ 1 & 4 & 9 & 0 & 0 & 1 \end{array}\right) \begin{array}{l} \\ -\text{Zeile}_1 \\ -\text{Zeile}_1 \end{array} \\
&= \left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & -1 & 1 & 0 \\ 0 & 3 & 8 & -1 & 0 & 1 \end{array}\right) \begin{array}{l} \\ \\ -3 \cdot \text{Zeile}_2 \end{array} \\
&= \left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & -1 & 1 & 0 \\ 0 & 0 & 2 & 2 & -3 & 1 \end{array}\right) \begin{array}{l} -\text{Zeile}_2 \\ \\ \\ \end{array} \\
&= \left(\begin{array}{ccc|ccc} 1 & 0 & -1 & 2 & -1 & 0 \\ 0 & 1 & 2 & -1 & 1 & 0 \\ 0 & 0 & 2 & 2 & -3 & 1 \end{array}\right) \begin{array}{l} +\frac{1}{2} \cdot \text{Zeile}_3 \\ -\text{Zeile}_3 \\ \\ \end{array} \\
&= \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 3 & -\frac{5}{2} & \frac{1}{2} \\ 0 & 1 & 0 & -3 & 4 & -1 \\ 0 & 0 & 2 & 2 & -3 & 1 \end{array}\right) \begin{array}{l} \\ \\ : 2 \end{array} \\
&= \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 3 & -\frac{5}{2} & \frac{1}{2} \\ 0 & 1 & 0 & -3 & 4 & -1 \\ 0 & 0 & 1 & 1 & -\frac{3}{2} & \frac{1}{2} \end{array}\right) \\
&= (E|A^{-1})
\end{aligned}$$

$$A^{-1} = \begin{pmatrix} 3 & -\frac{5}{2} & \frac{1}{2} \\ -3 & 4 & -1 \\ 1 & -\frac{3}{2} & \frac{1}{2} \end{pmatrix}$$

2.2.2 Lösung des Eigenwert-Problems

Ist R eine quadratische Matrix und gilt für eine beliebige Zahl l und einen Vektor v, der nicht der Nullvektor ist, die Gleichung $R * v = l * v$, dann heißt l ''Eigenwert von R" und v heißt ''zum Eigenwert l gehörender Eigenvektor".

Prinzipiell existieren zwei Möglichkeiten, das Eigenwertproblem zu lösen. Die eine Idee stützt sich auf Extremaleigenschaften der Eigenvektoren und ist nur bei symmetrischen Matrizes zu verwenden.
Die andere geht von einer Basis von Eigenvektoren aus, aus denen man eine unendliche Folge von Vektoren bildet, so dass eine bestimmte Folgengliederkomponente bei der Entwicklung überwiegt. Die so konstruierte Folge strebt der Richtung nach gegen einen Eigenvektor.

Um das Eigenwertproblem zu lösen sind folgende Schritte notwendig:

1. Umwandeln der Gleichung

   $$R * v = l * v$$
   $$R * v - \lambda * v = 0$$

   $$(R - \lambda * I) * v = 0 \tag{2.1}$$

   Wobei I die $(m \times m)$Einheitsmatrix ist und $(R - \lambda * I)$ vom Typ $(m \times m)$ ist.

2. Untersuchen der Gleichung 2.1
   Es ist ein homogenes Gleichungssystem. Daher gibt es für die in v stehenden Koeffizienten zwei Lösungen:

   - Triviallösung: Alle Elemente von v sind gleich Null
     Das ist die einzige Lösung, falls $(R - \lambda * I)$ eine reguläre Matrix ist.
   - nicht-triviale Lösung: Es existieren unendlich viele Vektoren v, die mindestens ein Element $\neq 0$ besitzen. Ist ein Vektor v Lösung des Gleichungssystems, so ist jeder Vielfache dieses Vektors auch Lösung des Gleichungssystems.
     Diese Variante tritt ein, wenn $(R - \lambda * I)$ eine singuläre Matrix ist.

   Da aber in der Überlegung zu Formel 2.1 (s.o.) ausgeschlossen wurde, dass v der Nullvektor ist, kommt hier nur der nicht-triviale Fall in Betracht.

   $$\det(R - \lambda * I) = 0 \tag{2.2}$$

3. Berechnung des charakteristischen Polynoms
   Um 2.2 zu lösen, müssen wir die Berechnung der Determinanten betrachten. Diese führt bei einer $(m \times m)$-Matrix zu einem Polynom m-ten Grades, dem *charakteristischen Polynom* oder der *charakteristischen Gleichung.*
   Unter Normalbedingungen (d.h. wenn R vom Rang m ist) hat das charakteristische Polynom m Lösungen.

Alle diese Überlegungen gelten für symmetrische Matrizes. Bei asymmetrischen Matrizes gilt es noch einige Feinheiten zu beachten.
So gilt nach [Mey03], dass sich jede quadratische Matrix X auf eine und genau eine Art durch reelle Eigenvektoren und Eigenwerte darstellen läßt.

$$X = P * \Delta * Q^T$$

Δ ist die Diagonalmatrix der Eigenwerte und P und Q sind orthogonale Matrizes der zugehörigen rechts- und linksseitigen Eigenvektoren.
Sei V die Matrix der Eigenvektoren und Δ die Diagonalmatrix der Eigenwerte, nach 2.1 gilt dann bei symmetrischen Matrizes:

$$R * V = V * \Delta \qquad \text{bzw.} \qquad V^T * R = \Delta * V^T$$

Bei asymmetrischen Matrizes muss man aber unterscheiden, es gilt hier

$$R * P = P * \Delta \qquad \text{für linksseitige Eigenvektoren,} \tag{2.3}$$

$$Q^T * R = \Delta * Q^T \qquad \text{für rechtsseitige Eigenvektoren.} \tag{2.4}$$

Die Eigenwerte λ sind in beiden Gleichungen die selben. Aus 2.3 und 2.4 lässt sich zeigen, dass — vorausgesetzt, die Eigenvektoren sind auf den Betrag Eins normiert — Folgendes gilt:

$$P^{-1} = Q^T \qquad \text{und} \qquad P = \left(Q^T\right)^{-1}$$

Die Matrix der rechtsseitigen Eigenvektoren ist also die Inverse der Matrix der linksseitigen Eigenvektoren und umgekehrt. Dies gilt jedoch nur, wenn die Matrix R quadratisch ist.

Zur Verdeutlichung ein Beispiel zur Berechnung von Eigenwerten und Eigenveketoren einer nicht-symmetrischen, quadratischen Matrix:

$$R = \begin{pmatrix} 8 & -2 & 0 \\ 0 & 8 & -2 \\ -12 & 22 & -4 \end{pmatrix}$$

Nach 2.2 muss gelten:

$$\det(R - \lambda * I) = \det \begin{pmatrix} 8-\lambda & -2 & 0 \\ 0 & 8-\lambda & -2 \\ -12 & 22 & -4-\lambda \end{pmatrix} = 0$$

Damit ergibt sich die Determinante

$$\begin{aligned}\det(R-\lambda * I) &= (8-\lambda)(8-\lambda)(-4-\lambda)+(-2)(-2)(-12)+(0)(0)(22)\\ &\quad -(-12)(8-\lambda)(0)-(-4-\lambda)(0)(-2)-(22)(-2)(8-\lambda)\\ &= (64-16\lambda+\lambda^2)(-4-\lambda)-48+44(8-\lambda)\\ &= -256+12\lambda^2-\lambda^3-48+352-44\lambda\end{aligned}$$

und hieraus folgt das charakteristische Polynom

$$-\lambda^3+12\lambda^2-44\lambda+48=0$$

mit den Lösungen

$$\begin{aligned}\lambda_1 &= 6\\ \lambda_2 &= 4\\ \lambda_3 &= 2\end{aligned}$$

Das sind die Eigenwerte. Um jetzt die zugehörigen Eigenvektoren zu erhalten, sind die Eigenwerte jeweils in die Gleichung

$$(R-\lambda_i * I) * p_i \qquad \text{bzw.} \qquad q_i^T * (R-\lambda_i * I)$$

einzusetzen. Um also den linksseitigen Eigenvektor zu $\lambda = 6$ erhalten, löst man

$$\begin{pmatrix} 8-6 & -2 & 0 \\ 0 & 8-6 & -2 \\ -12 & 22 & -4-6 \end{pmatrix} \cdot \begin{pmatrix} p_1 & \\ p_2 & s \\ p_3 & \end{pmatrix} = 0$$

Daraus erhält man die drei Gleichungen

$$\begin{aligned} 2p_1 - 2p_2 &= 0\\ 2p_2 - 2p_3 &= 0\\ -12p_1 + 22p_2 - 10p_3 &= 0\end{aligned}$$

Hierbei muss man nur eine Lösung für die unterste Gleichung finden. Diese ergibt sich als

$$p_1 = 1, \qquad p_2 = 1, \qquad p_3 = 1$$

Wenn man dies auch für die anderen zwei Werte von λ berechnet, kommt man zu folgenden Ergebnissen:

$$p_1 = \begin{pmatrix}1\\1\\1\end{pmatrix} \qquad p_2 = \begin{pmatrix}1\\2\\4\end{pmatrix} \qquad p_3 = \begin{pmatrix}1\\3\\9\end{pmatrix}$$

Die Matrix der linksseitigen Eigenvektoren lautet also

$$P = \begin{pmatrix}1 & 1 & 1\\ 1 & 2 & 3\\ 1 & 4 & 9\end{pmatrix}$$

Geht man bei den anderen Gleichungen genauso vor, bekommt man

$$Q = \begin{pmatrix} 6 & -3 & 2 \\ -5 & 4 & -3 \\ 1 & -1 & 1 \end{pmatrix}$$

Berechnet man die Inverse von P (s. 2.2.1, Seite 14),

$$P^{-1} = \begin{pmatrix} 6 & -5 & 1 \\ -6 & 8 & -2 \\ 2 & -3 & 1 \end{pmatrix}$$

so stellt man fest, dass sich P^{-1} und Q^T bis auf einen Normierungsfaktor gleichen und die Beziehung gilt:

$$Q^T = \frac{1}{2} \cdot P^{-1}.$$

Dadurch erhält man für R:

$$\Rightarrow R = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 9 \end{pmatrix} * \begin{pmatrix} 6 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{pmatrix} * \begin{pmatrix} 3 & -\frac{5}{2} & \frac{1}{2} \\ -3 & 4 & -1 \\ 1 & -\frac{3}{2} & \frac{1}{2} \end{pmatrix} \Rightarrow R = \begin{pmatrix} 8 & -2 & 0 \\ 0 & 8 & -2 \\ -12 & 22 & -4 \end{pmatrix}$$

2.3 Diskriminanzanalyse

In diesem Kapitel werden die theoretischen Grundlagen der Diskriminanzanalyse dargestellt. Die mathematische Darstellung wird im Kapitel 2.4 (ab Seite 25) beschrieben.
Die Diskriminanzanalyse ist als strukturprüfendes Verfahren ein multivariates Verfahren zur Analyse von Gruppenunterschieden. Sie untersucht also die Unterschiedlichkeit zweier oder mehrerer Gruppen hinsichtlich mehrerer Variablen. Dabei muss bei den Daten allerdings eine Zuordnung der Art "Merkmalsvariable → Gruppenzugehörigkeit" vorliegen. Hierin liegt auch der große Unterschied zu den anderen taxonomischen (= gruppierenden) Verfahren z.B. der Clusteranalyse (S. 7): Die Diskriminanzanalyse geht *nicht* von ungruppierten Daten aus!

Obwohl man vermuten könnte, dass das doch eher wissenschaftliche Anwendungsgebiet der Analyse von Gruppenunterschieden nur selten Verwendung findet, kann man feststellen, dass auch durchaus praktische Anwendungsgebiete existieren. Hierbei ist besonders die Prognose der Gruppenzugehörigkeit zu nennen. Einige Anwendungsbeispiele sind in der Tabelle 2.5 auf Seite 24 zu sehen, die [BEPW96] entnommen wurde.
Ein bekanntes Beispiel zur Anwendung der Diskriminanzanalyse ist die Analyse der Kontendaten für eine Kreditwürdigkeitsprüfung.
Durch die Diskriminanzanalyse wird anhand verschiedener Konto-Daten (z.B. Verhältnis von Soll zu Haben) einerseits die Diskriminanzfunktion berechnet, die ''gute" von ''schlechten" Konten trennt und andererseits der *Diskriminanzpunkt* bestimmt, der angibt, bis zu welchem Wert ein Konto als ''schlecht" und ab welchem Wert ein Konto als ''gut" angesehen wird. Dieser Diskriminanzpunkt ist auf der Diskriminanzachse zu finden, wie in Abb. 2.1 gezeigt.

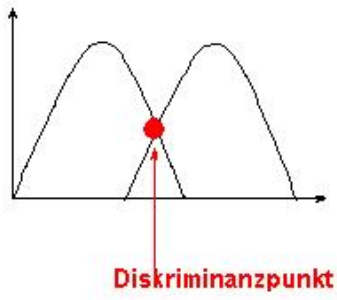


Abbildung 2.1: Lage des Diskriminanzpunkts

Hat man die Diskriminanzfunktion errechnet, kann für jedes neue auszuwertende Konto die Diskriminanzfunktion berechnet und das Konto durch Berechnung des Abstands vom Diskriminanzpunkt in die jeweilige Gruppe eingeordnet werden.(Abb. 2.2)

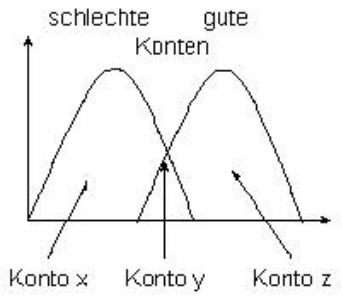


Abbildung 2.2: Einordnung von Werten aufgrund der Diskriminanzfunktion

2.3.1 Prinzipielles Vorgehen

1. Erhebe eine erhärtete Stichprobe[1]
2. Entwickle die Zuordnungsvorschrift
3. Erhebe an weiteren Personen nur noch den Klassifikationsvektor und bestimme mit Hilfe der Zuordnungsvorschriften die Gruppenvariable

Da ein Ziel der Diskriminanzanalyse die Zuordnung neuer Fälle ist, interessiert hauptsächlich der Wert der Gruppenvariablen. Um diesen Schluss von Klassifikationsvektor auf Gruppenvariable zu ermöglichen, müssen vorher noch einige Voraussetzungen erfüllt werden:

- In der Stichprobe dürfen keine Elemente vorkommen, die gleichzeitig zu mehr als einer Gruppe gehören (z.B. Person mit 2 Berufen)
- Der Umfang der Stichprobe sollte wenigstens doppelt so groß sein wie die Anzahl der Merkmalsvariablen
- Die Anzahl der Merkmalsvariablen sollte größer sein als die Anzahl der Gruppen

2.3.2 Anforderungen an ein Diskriminanzanalyse-Programm

1. Beschreibung der Populationen mit Basisstatistiken
   - Mittelwerte, Häufigkeiten
   - Teststatistiken (t-Test, Chi-Quadrat-Test)
   - Korrelationsdiagramme
   - Kovarianz-Matrizen
2. Bestimmung der Zuordnungsregeln

[1] Stichprobe, bei der Klassifikationsvektor und zugehörige Gruppenvariable schon bestimmt wurden

3. Bewertung der Güte der Zuordnungsregeln
   (=Durchführung einer Fehlerratenschätzung)
   Resubstitution } nicht ausschließlich
   Leaving-One-Out } wünschenswert
   Train-and-Test }
   aber auch Bootstrap und Cross-Validation

4. Zuordnung neuer Fälle

5. Auswahl der Merkmale

6. Graphische Darstellung der Resultate

2.3.3 Prüfung der Klassifikation

Zur Prüfung der Klassifikation haben sich mehrere Verfahren etabliert:

- **Resubstitution**
  Schätzung und Prüfung der Daten werden mit dem Ausgangsdatensatz durchgeführt, ohne dass aus diesem Daten entfernt oder hinzugefügt werden. Die Fehlerrate, die man bei diesem Test erhält, ist die gesuchte Fehlerrate.
  Allerdings neigt die Resubstitution zur optimistischen Unterschätzung der Fehlerrate, je kleiner die Stichprobe ist.

  Vorgehensweise:

  1. Bestimme die Zuordnungsregeln
  2. Wende die Zuordnungsregeln auf alle Elemente der Stichprobe an und bestimme die Anzahl der falsch zugeordneten Datensätze

- **Train-and-Test**
  Der Ausgangsdatensatz wird aufgeteilt in:

  – Trainingsdatensatz - dieser enthält die Daten für die Schätzung
  – Testdatensatz - hierin sind 20 bis 30 % der Ausgangsdaten enthalten

  Die gesuchte Fehlerrate ergibt sich in diesem Fall aus der Fehlerrate des Testdatensatzes. Sie wird im Mittel richtig geschätzt.
  Nachteil dieser Methode ist, dass man zwei Stichproben benötigt. Dadurch wird der Umfang des Trainingsdatensatzes kleiner, und die Qualität der Zuordnung leidet, weil bei der Diskriminanzanalyse gilt:
  Je mehr Daten vorhanden sind, desto besser können die Zuordnungsregeln bestimmt werden.

  Vorgehensweise:

  1. Bestimme die Zuordnungsregeln aufgrund des Trainingsdatensatzes

2. Wende die Zuordnungsregeln auf alle Elemente der Teststichprobe an und bestimme die Anzahl der falsch zugeordneten Datensätze

- **Cross-Validation**
  Hierbei werden die Ausgangsdaten in m (gleichgroße) Stichproben aufgeteilt. Dabei gelten bei jedem Durchgang
  (m-1) als Trainingsdatensatz und
  1 als Testdatensatz
  Die Fehlerrate ergibt sich hier aus dem Durchschnitt aller Testdaten-Fehlerraten.

- **Leaving-One-Out**
  Bei dieser Test-Methode nimmt man jeweils einen Datensatz aus den Ausgangsdaten und verwendet diesen als Test-Datensatz. Die gesuchte Fehlerrate findet man durch Durchschnittsbildung der Fehlerraten der einzelnen Stichproben.

  Vorgehensweise:

  1. Nimm den i-ten Fall als Teststichprobe, die restlichen n-1 Fälle bilden die Trainingsstichprobe
  2. Bestimme die Zuordnungsregeln mit der Trainingsstichprobe
  3. Wende die Zuordnungsregeln auf den einen Fall der Teststichprobe an; stelle fest, ob er richtig zugeordnet wurde
  4. Führe die Schritte 1 - 3 für i=1, 2, ... n durch und zähle die Anzahl der falsch zugeordneten Fälle

- **Bootstrap**
  Hierbei erzeugt man m Testdatensätze, die die gleiche Anzahl an Daten enthalten wie der Ausgangsdatensatz. Jeder Testdatensatz wird hier unter Verwendung folgender Regeln erzeugt:

  – Entferne zufällig einzelne Elemente (ca. $\frac{1}{e} \equiv 37\%$)
  – Füge zufällig bereits vorhandene Elemente hinzu

  Mit diesen Datensätzen wird nun jeweils eine Diskriminanzanalyse durchgeführt. Die Fehlerrate ergibt sich aus dem Durchschnitt aller m Fehlerraten.

2.3.3.1 Zusammenfassung

- Die *Leaving-one-out-Methode* kann als Standardmethode empfohlen werden, da sie – nach Deichsel / Trampisch ([DT85]) – im Mittel die richtigen Schätzwerte liefert.

- Die *Train-and-Test-Methode* muss verwendet werden, wenn eine schrittweise Auswahl von Variablen mittels der Diskriminanzanalyse durchgeführt wurde. In diesem Fall ist sie unerlässlich zur Angabe der Fehlerrate der im letzen Schritt ausgewählten Variablenkombination.

- Die *Resubstitutions-Methode* ist alleine nicht anzuwenden! Sie dient nur zum Erhalten der Vorstellung einer Verlässlichkeit der Leaving-one-out-Methode.
  Sind beide Werte gleich, ist der Stichproben-Umfang groß genug zur verlässlichen Bestimmung von Zuordnungsregeln und zur Angabe von Fehlerraten.
  Unterscheiden sich die beiden Werte, ist der Stichproben-Umfang nicht groß genug dazu.

Problemstellung	**Gruppierung**	**Merkmalsvariablen**
Prüfung der Kreditwürdigkeit	Risikoklasse: – hoch – niedrig	Soziodemographische Merkmale (Alter, Einkommen, etc.) Anzahl weiterer Kredite, Beschäftigungsdauer, etc.
Auswahl von Außendienstmitarbeitern	Verkaufserfolg: – hoch – niedrig	Ausbildung, Alter, Perönlichkeitsmerkmale, körperliche Merkmale etc.
Analyse der Markenwahl beim Autokauf	Marke: – Mercedes – BMW – Audi, etc.	Wünsche zu Eigenschaften von Autos, z.B.: Aussehen, Straßenlage, Höchstgeschwindigkeit, Wirtschaftlichkeit etc.
Wähleranalyse	Partei – CDU – SPD – FDP – Grüne	Einstellung zu politischen Themen wie Abrüstung, Atomernergie, Tempolimit, Besteuerung, Wehrdienst, Mitbestimmung etc.
Diagnose von Atemnot bei Neugeborenen	Überleben: – ja – nein	Geburtsgewicht, Geschlecht, pH-Wert des Blutes, postmenstruales Alter der Mutter etc.
Erfolgsaussichten von neuen Produkten	Wirtschaftl. Erfolg: – Gewinn – Verlust	Neuigkeitsgrad des Produktes, Marktkenntnis des Unternehmens, Preis/Leistungs-Verhältnis, technolog. Know-how etc.
Analyse der Diffusion von Innovationen	Adoptergruppen: – Innovatoren – Imitatoren	Risikofreudigkeit, soziale Mobilität, Einkommen, Statusbewußtsein etc.

Tabelle 2.5: Anwendungsgebiete der Diskriminanzanalyse

2.4 Formeln der Diskriminanzanalyse

Um dem Leser die Deutung der Formeln zu vereinfachen, werde ich hier die bei allen Formeln verwendeten Indizes erklären:

g, h Bezeichnung für ein Merkmal ($g = 1, \ldots, m$)

m steht für die Merkmals-Anzahl

i repräsentiert die Gruppen ($i = 1, \ldots, p$)

p ist die max. Anzahl von Gruppen

l steht für einen Datensatz (z.B. eine Versuchsperson) ($l = 1, \ldots, n$)

n ist die Datensatz-Anzahl

n_i ist die Anzahl von Datensätzen in der Gruppe i

y Messergebnis

So wird das Meßergebnis der l-ten Person der i-ten Gruppe für das g-te Merkmal dargestellt durch y_{gil}.

2.4.1 Vorarbeiten

Um die später folgenden Berechnungen durchführen zu können und die Formeln später vereinfacht darstellen zu können, werden am Anfang die Mittelwerte der einzelnen Merkmale pro Gruppe

$$\bar{y}_{gi} = \frac{\sum_{l=1}^{n_i} \bar{y}_{gil}}{n_i}$$

und die Mittelwerte über alle Gruppen berechnet

$$\bar{y}_{g} = \frac{\sum_{i=1}^{p} \sum_{l=1}^{n_i} \bar{y}_{gil}}{n}.$$

Um die zur späteren Berechnung benötigte Kovarianzmatrix berechnen zu können, fehlen nun noch die Summen der Produkte der Abweichungen vom Mittelwert zwischen g-tem und h-tem Merkmal über alle Gruppen. Diese berechnen sich nach folgender Formel:

$$q_{gh} = \sum_{i=1}^{p} \sum_{l=1}^{n_i} (\bar{y}_{gil} - \bar{y}_{gi})(\bar{y}_{hil} - \bar{y}_{hi})$$

Nach dieser Vorarbeit lassen sich die Elemente der Kovarianzmatrix durch

$$s_{gh} = \frac{q_{gh}}{n - p}$$

ermitteln, sodass die *Kovarianzmatrix* über alle Gruppen wie folgt aussieht:

$$S = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1m} \\ s_{21} & s_{22} & \dots & s_{2m} \\ \vdots & \vdots & & \vdots \\ s_{m1} & s_{m2} & \dots & s_{mm} \end{pmatrix}$$

Die *inverse Kovarianzmatrix* wird durch $\hat{S}$ dargestellt und ihre Elemente mit $\hat{s}$ bezeichnet.

2.4.2 Beurteilung der Trennwirksamkeit

Das *multivariate Trennmaß* T^2 erhält man durch

$$T^2(Y_1, \dots, Y_m) = \frac{1}{n-p} \sum_{g=1}^{m} \sum_{h=1}^{m} \hat{s}_{gh} \sum_{p=1}^{i} n_i (y_{gi} - y_g)(y_{hi} - y_h)$$

Ob diese Trennung wirksamm ist, lässt sich an Hand einer F-Verteilung mit den Freiheitsgraden

$$\begin{aligned} v_1 &= \begin{cases} \frac{(p-1)\cdot m \cdot (n-p-m)}{n-(p-1)\cdot m-2} & \text{, falls } n-(p-1)\cdot m-2 > 0 \\ \infty & \text{, falls } n-(p-1)\cdot m-2 \leq 0 \end{cases} \\ v_2 &= n-p-m+1 \end{aligned}$$

ermitteln. Der so berechnete F-Wert ($F_{\alpha;v_1;v_2}$) wird nun mit der errechneten Prüfgröße $\hat{F}$ verglichen.

$$\hat{F} = \frac{n-p-m+1}{(p-1)\cdot m} \cdot T^2$$

Danach entscheidet man folgendermaßen:

$$\begin{aligned} \hat{F} \leq F_{\alpha;v_1;v_2} \quad &\longrightarrow \quad \text{Trennung unwirksam} \\ \hat{F} > F_{\alpha;v_1;v_2} \quad &\longrightarrow \quad \text{Trennung wirksam} \end{aligned}$$

2.4.3 Unentbehrlichkeiten berechnen

Die *Unentbehrlichkeit* eines Merkmals (U_g) beschreibt das Maß, um das das Trennmaß T^2 absinkt, wenn man das dazugehörige Merkmal (g) entfernt.

$$U_g = \frac{1}{(n-p)} \cdot \hat{s}_g g \cdot \sum_{i=1}^{p} n_i c_{gi}^2 \qquad (g = 1, \dots, m)$$

Ob ein Merkmal entbehrlich ist oder nicht, lässt sich wieder mit Hilfe der F-Verteilung feststellen. Man vergleicht in diesem Fall den F-Wert von $F_{\alpha;v1;v2}$ mit $v_1 = p-1$ und $v_2 = n-p-m$ mit dem zu errechnenden Prüfwert $\hat{F}_g$.

$$\hat{F}_g = \frac{n-p-m+1}{p-1} \cdot \frac{U_g}{1+T^2(y_1, \dots, y_m) - U_g}$$

Ist bei diesem Vergleich $\hat{F}_g \leq F$, so ist das Merkmal Y_g entbehrlich. Andernfalls bezeichnet man das Merkmal Y_g als unentbehrlich.

2.4.4 Zuordnung von Individuen

Zur Berechnung der Zuordnung eines Individuums zu einer Gruppe müssen die Kovarianzmatrix und die Mittelwert-Vektoren berechnet sein. Zusätzlich benötigt man noch die sog. *Fehlermatrix*

$$S_e = (n - p) * S$$

und die *Hypothesenmatrix*.

$$S_h = \sum_{i=1}^{p} n_i * (\bar{y}_i - \bar{y})(\bar{y}_i - \bar{y})^T$$

Bei der generalisierten linearen Diskriminanzfunktion bezeichnet α einen beliebigen Eigenvektor zum größten Eigenwert λ_G der Matrix $S_e^{-1} * S_h$, d.h.

$$S_e^{-1} * S_h * \alpha = \lambda_G * \alpha$$

dann wird ein neues Objekt mit Beobachtungsvektor y in die i-te Gruppe eingeteilt, falls für alle $j \neq i$ gilt:

$$|\alpha^T * (\bar{y}_i - y)| < |\alpha^T * (\bar{y}_j - y)|$$

Bei der linearen Fisherschen Diskriminanzfunktion wird das Objekt mit Beobachtungsvektor y in die i-te Gruppe eingeordnet, wenn für alle $j \neq i$ gilt:

$$h_{ij}(y) = (\bar{y}_i - \bar{y}_j) * \hat{S} * y - \frac{1}{2}(\bar{y}_i - \bar{y}_j) * \hat{S} * (\bar{y}_i + \bar{y}_j) > 0$$

Hierbei lässt sich bei der konkreten Berechnung der Aufwand etwas durch die Beziehung

$$h_{ij}(y) = -h_{ji}(y)$$

reduzieren.

2.5 Vergleich Neuronale Netze ./. Diskriminanzanalyse

Sowohl Neuronale Netze als auch die Diskriminanzanalyse gehören zu den (selbst-) lernenden Verfahren; d.h. es gibt keinen fest vorgeschriebenen Algorithmus, sondern das System versucht selbständig aus den vorhandenen Daten Klassifikationsregeln zu erstellen, die einen möglichst kleinen Fehler aufweisen. Weil beide Verfahren unterschiedliche Ansätze benutzen, ist auf den ersten Blick nicht erkennbar, welcher Ansatz der bessere ist.

Bei den Neuronalen Netzen liegt das Potential in den unterschiedlichen Möglichkeiten der Benutzung. Es ist möglich, den Hypothesenraum durch Einschränkung der Neuronenzahl einzugrenzen, um ihn so den vorhandenen Informationen besser anpassen zu können. Außerdem kann man aber durch Hinzunahme neuer Neuronen den Eingaberaum beliebig fein strukturieren.
Darüber hinaus kann man durch eine Vergrößerung der Netzstruktur die Anpassung an die Trainingsdaten erhöhen, bekommt dadurch aber gleichzeitig einen erhöhten Testfehler. Es läßt sich erst in der Testphase feststellen, ob das Verhältnis von "guter Anpassung" und "großer Klassifizierungsfehler" gelungen ist oder nicht.

Die Diskriminanzanalyse ermöglicht es schon während der Trainingsphase, den Klassifikationsfehler zu minimieren.
Die Diskriminanzanalyse geht außerdem von normalverteilten Daten aus, wobei hierzu bei Neuronalen Netzen keine Beschränkungen existieren.

Der Unterschied in der Wirkungsweise hängt einerseits von den Zusammenhängen zwischen Messung und Klassifizierung ab und andererseits von Art und Weise sowie Grad der Zufälligkeit.

2.6 Problemstellung

Die Diskriminanzanalyse ist, wie bereits auf Seite 4 beschrieben, ein statistisches Verfahren. Hierbei wird versucht, aus einer gegebenen Datenmenge, bei der schon eine Einteilung in Gruppen erfolgt ist, eine Funktion zu berechnen. Mit Hilfe dieser Funktion ist es später möglich, neue Datensätze ebenfalls in diese Gruppen einzuordnen.

Meine Aufgabe bestand darin, ein Programm zu entwickeln, das dem Anwender auf einfache Weise gestattet, eine Diskriminanzanalyse durchzuführen.

Es sollte also möglich sein, in meinem Programm Daten einzugeben bzw. zu ändern. Mein Programm sollte die Diskriminanzfunktion, die Unentbehrlichkeiten, das Trennmaß und die Trennkriterien anzeigen.
Zusätzlich musste es eine Möglichkeit geben, die errechnete Diskriminanzfunktion auf beliebige Daten anzuwenden.
Für die Nutzer, die daran interessiert sind, existiert eine Kontroll-Seite, auf der sie sich die Mittelwertmatrix, die Kovarianzmatrix sowie die inverse Kovarianzmatrix ansehen können.
Als letztes bleibt noch die Statistik-Seite zu erwähnen, auf der eine Zusammenfassung zu finden ist, wie viele Datensätze nach der errechneten Diskriminanzfunktion und den Trennkriterien korrekt zugeordnet wurden; dieser Test wird nach zwei verschiedenen Verfahren berechnet

- Resubstitutions-Methode (S. 21) und
- Leaving-one-Out-Methode (S. 22).

Um die Ergebnisse auch weitergeben zu können, kann man diese auch alle ausdrucken.
Meine geplanten Realisierungen habe ich in Form von Tabelle 2.6 (S. 30) festgehalten.

Anforderung
Beschreibung mit Basisstatistiken
Mittelwerte
Häufigkeiten
Teststatistiken
Bestimmung der Zuordnungsregeln
Bewertung der Güte der Zuordnungsregeln
Leaving-one-out
Train-and-Test
Resubstitution
Zuordnung neuer Fälle möglich
Reduktion der Merkmale
manuell
automatisch
Auswahl der zu betrachtenden Merkmale
Druck der Ergebnisse
Graphische Darstellung der Resultate

Tabelle 2.6: Anforderungen an ein Diskriminanzanalyse-Programm

Kapitel 3

Softwaretechnische Realisierung

In diesem Kapitel gehe ich auf die softwaretechnischen Aspekte meines Programmes ein, nachdem die mathematischen Hintergründe erklärt wurden.

Mein Programm besteht im Wesentlichen aus vier Anzeige-Seiten oder "Ansichten". Es sind das: die "Ergebnis-", die "Daten-", die "Kontroll-" und die "Statistik-Ansicht". Wie diese Seite zusammenhängen, ist in Abbildung 3.1 zu sehen.

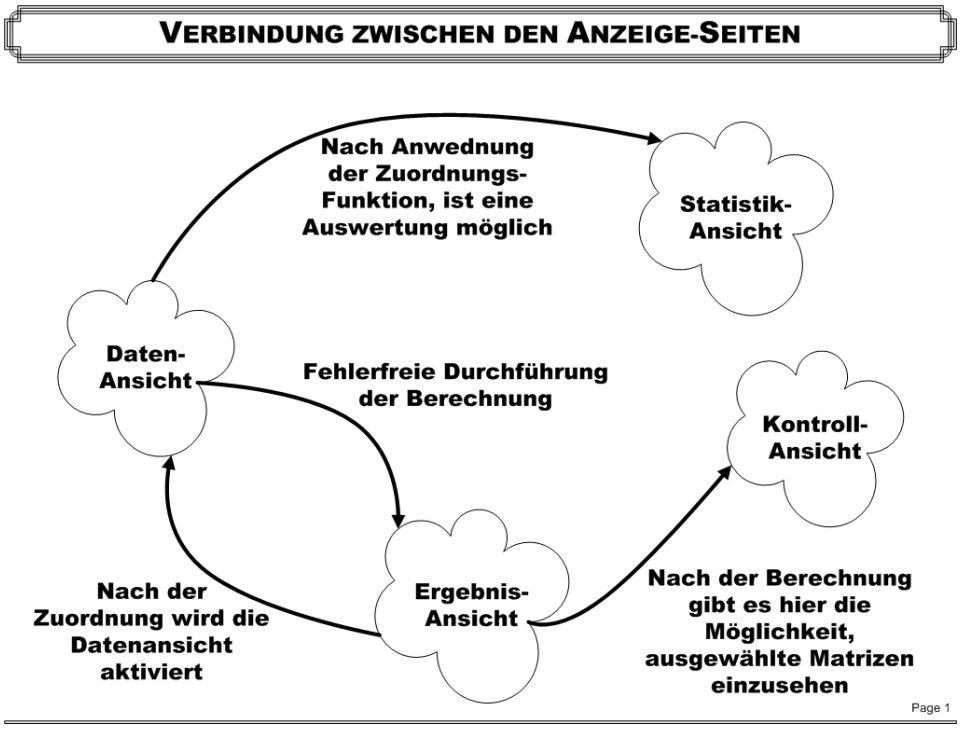


Abbildung 3.1: Verbindungen zwischen den Anzeige-Seiten

Wie diese Daten-Ansichten in einem "normalen"Porgrammablauf verwendet werden, kann man Abbildung 3.2 entnehmen. Der Benutzer legt entweder eine neue Datei an oder öffnet eine bestehende. Dann beginnt die Berechnung, indem er aus der nun erscheinenden Auswahl die Merkmale markiert, die für ihn interessant erscheinen. Mit der so erhaltenen Daten-Matrix wird nun die entsprechende Kovarianz-Matrix erstellt. Ist deren Diskriminante Null, so wird die Berechnung abgebrochen, das Verfahren kann dann nicht angewendet werden. Wird die Berechnung fortgesetzt, werden dem Benutzer nun auf der Ergebnis-Ansicht die Resultate präsentiert. Er hat jetzt die Möglichkeit, die Diskriminanzfunktion zu speichern, um sie später auch auf andere Daten anwenden zu können oder sie auf die momentan geladenen Daten anzuwenden. Nach der Anwendung der Diskriminanzfunktion sieht der Benutzer wieder die Daten-Ansicht. Inzwischen hat sich die ursprüngliche Daten-Matrix um zwei Spalten vergrößert, in denen der Diskriminanzwert und die neue Zuordnung für den jeweiligen Datensatz enthalten sind.

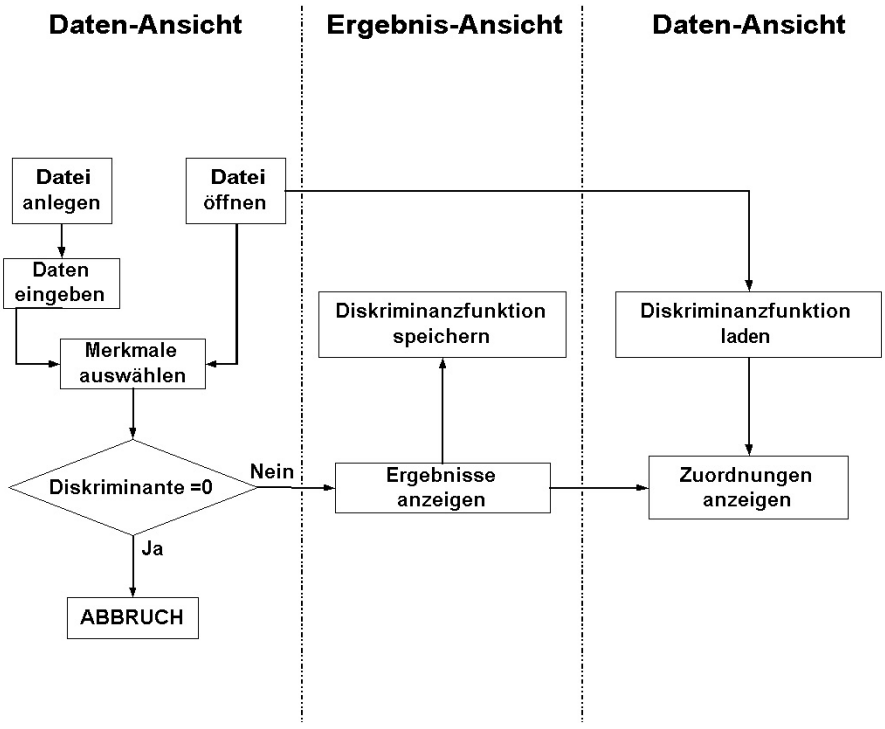


Abbildung 3.2: Struktur des Programm-Ablaufs

Die letzte Graphik in diesem Abschnitt (Abbildung 3.3) soll einen Überblick darüber geben, wie die einzelnen Berechnungs-Funktionen meines Programms zusammenhängen.

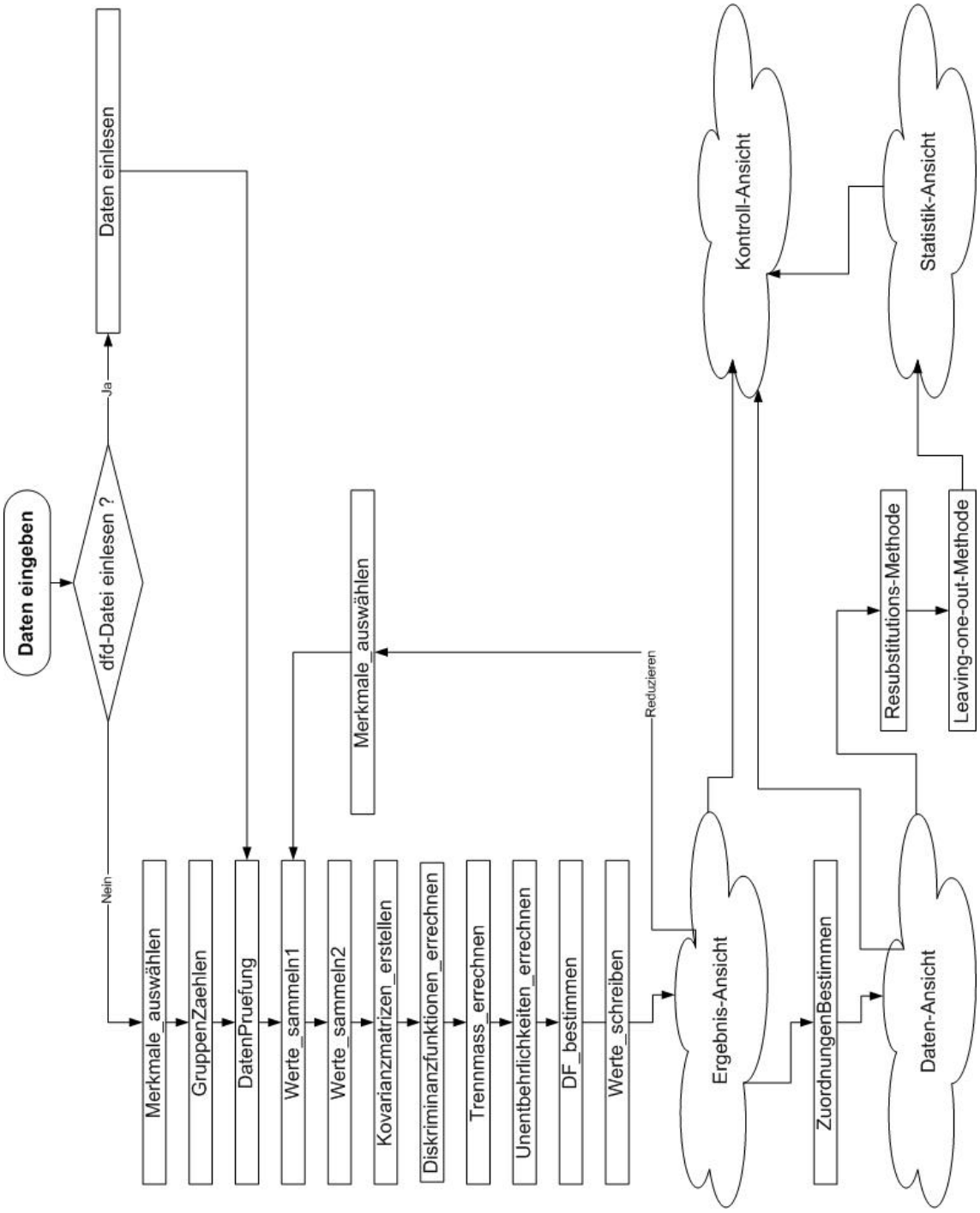


Abbildung 3.3: Zusammenhang zwischen den Berechnungs-Funktionen

3.1 Entwicklungsumgebung

Meine Wahl der Entwicklungsumgebung fiel - nach kurzer Überlegung - auf Delphi.
Im Laufe des Studiums hatte ich Gelegenheit einige Programmiersprachen kennenzulernen und auszuprobieren. So lernte ich C, C++, Java, VisualBasic, Lisp und Delphi kennen. Außerdem hatte ich Kontakt zur Assembler-Programmierung. Nachdem ich meine ersten Programmier-Schritte aber mit TurboPascal gemacht hatte, kam ich immer wieder auf diese Sprache zurück. Dabei hatte ich allerdings das Problem, dass sich mit TurboPascal graphische Oberflächen nur sehr aufwändig erstellen lassen, und es inzwischen mit Delphi einen geeigneten Nachfolger gibt.

Außerdem gefiel mir die Aussicht, durch Kylix (= Delphi für Linux) eine Portierung auf Linux zu erreichen. Auf die Probleme der Portierung gehe ich in einem späteren Kapitel (s. Kapitel 3.6) gesondert ein.

3.2 Entwurf / Aufbau der Software

3.2.1 Aufbau der Oberfläche

Beim Entwurf der Oberfläche hatte ich zuerst die *Daten-Eingabe* im Kopf. Hier hatte ich schon eine relativ klare Vorstellung, wie sie aussehen sollte. Ich wusste, dass auf diesem Formular ein Gitter zur Daten-Eingabe und -Änderung existieren musste, eine Möglichkeit Zeilen zu löschen und hinzuzufügen sowie eine Möglichkeit Spalten zu entfernen bzw. hinzuzufügen.
Bei den anderen Oberflächen hatte ich zu diesem Zeitpunkt noch keine klare Vorstellung.
Abbildung 3.4 zeigt den letzten Stand der Daten-Ansicht, nachdem eine Datei geladen wurde.

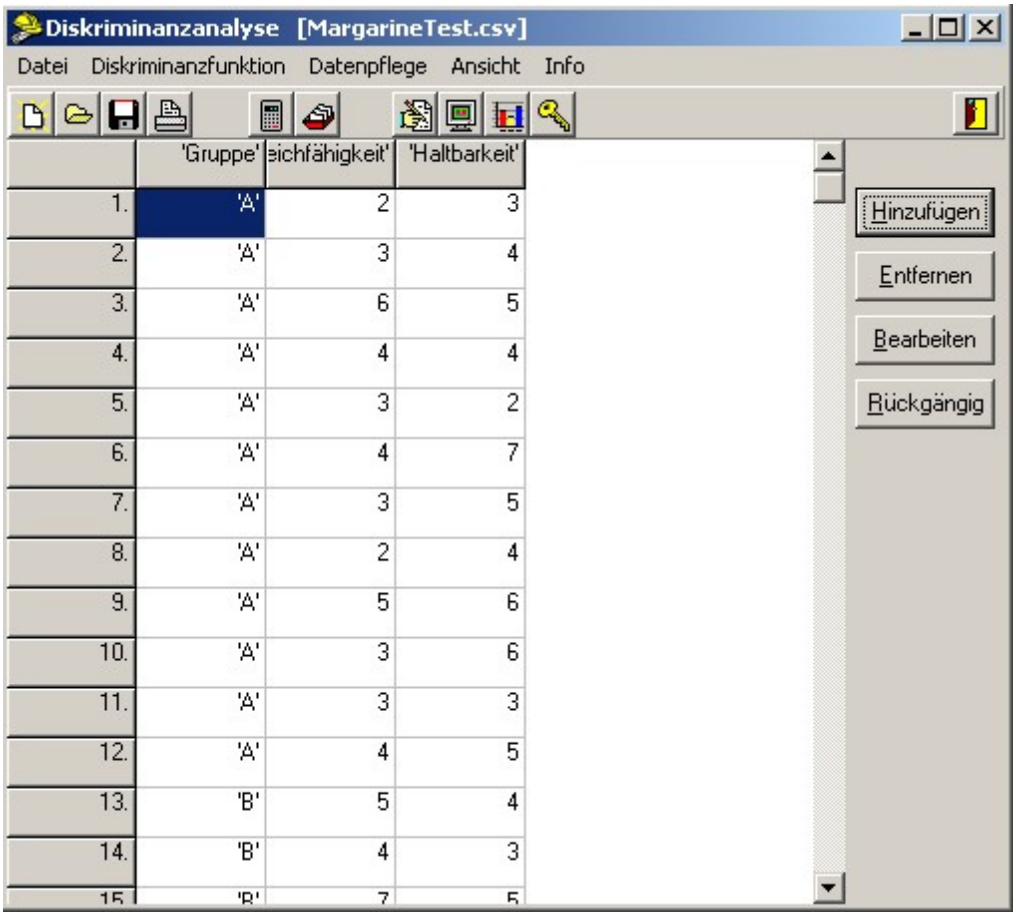


Abbildung 3.4: Endgültiges Aussehen der Daten-Ansicht

Danach machte ich mich an den Entwurf der *Ergebnis-Seite*. Ich wollte eine Tabelle haben, in der ich Informationen zu den einzelnen Gruppen darstellen konnte. Darin stehen – die Gruppenbezeichnungen, – wie häufig die Gruppen in dem vorhandenen Datensatz vorkommen und – nach der Analyse – die Zuordnungsvorschrift.

Später kamen Anzeigen für das Trennmaß und die Prüfgröße dazu.
Darüber hinaus benötigte ich eine Tabelle, um die Unentbehrlichkeiten darzustellen. Auch hierzu verwendete ich eine Tabelle, in der die Merkmalsnamen und die Unentbehrlichkeits-Werte angezeigt werden. An die Spalte für die Unentbehrlichkeits-Werte schließt sich noch eine Spalte an, in der gezeigt wird, ob das entsprechenden Merkmal ohne großen Informationsverlust von der Berechnung ausgeschlossen werden kann oder nicht.
Durch diese Anzeige wurde mir bewußt, dass es auch eine Möglichkeit geben sollte, Merkmale zu reduzieren - sie aus der Berechnung auszuschließen, falls sie sich bei der Berechnung als entbehrlich erweisen sollten.
Außerdem sollte es die Chance geben, einmal ausgeschlossene Merkmale wieder in die Berechnung einzubinden - die Merkmals-Reduktion sollte also auch wieder rückgängig zu machen sein.
In Abbildung 3.5 ist die letzte Version der Ergebnis-Ansicht im Einsatz zu sehen.

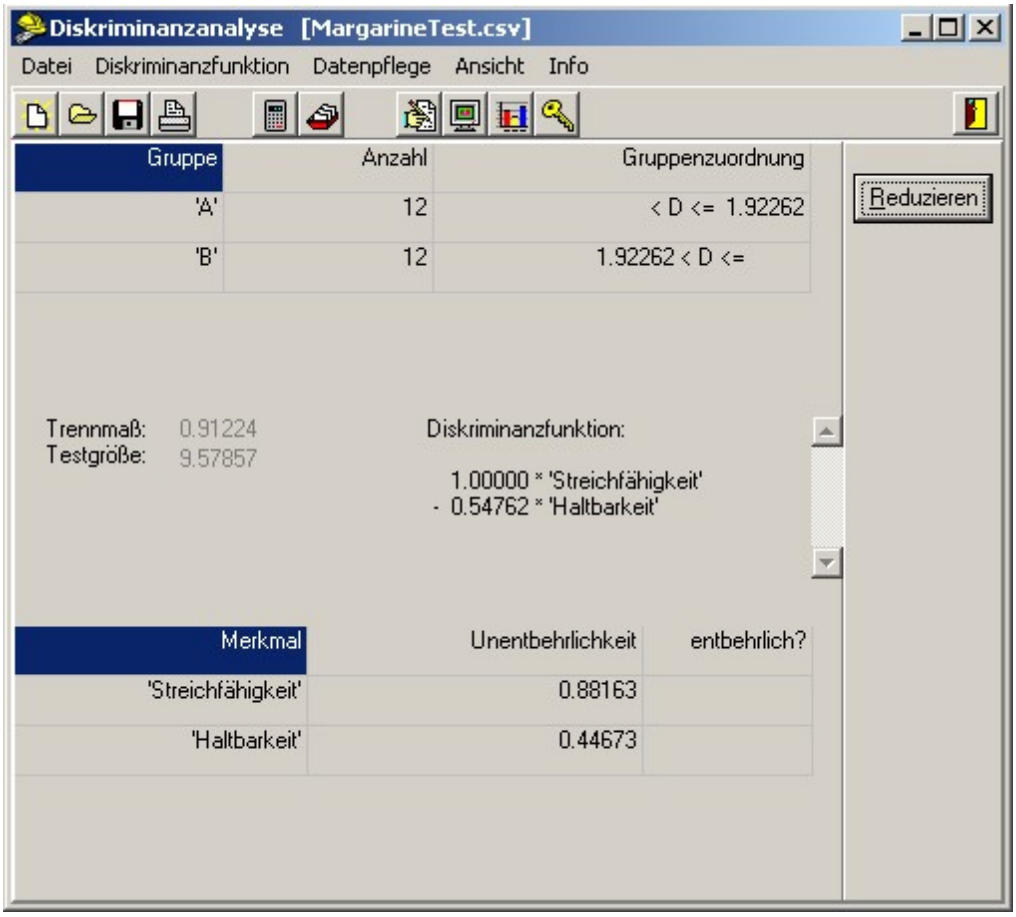


Abbildung 3.5: Endgültiges Erscheinungsbild der Ergebnis-Ansicht

Die *Kontroll-Seite* diente anfangs noch dazu, mir die Ausgabe einzelner Zwischenergebnisse zu erleichtern. Sie wurde später fester Bestandteil, als es

damit leichter wurde, die Matrizen mit den Angaben aus Büchern oder anderen Programmen zu vergleichen. Es werden jetzt die Mittelwert-Matrix, die Kovarianz-Matrix und die inverse Kovarianz-Matrix ausgegeben.
Diese Seite hat die meisten Veränderungen hinter sich (Abbildung 3.6).

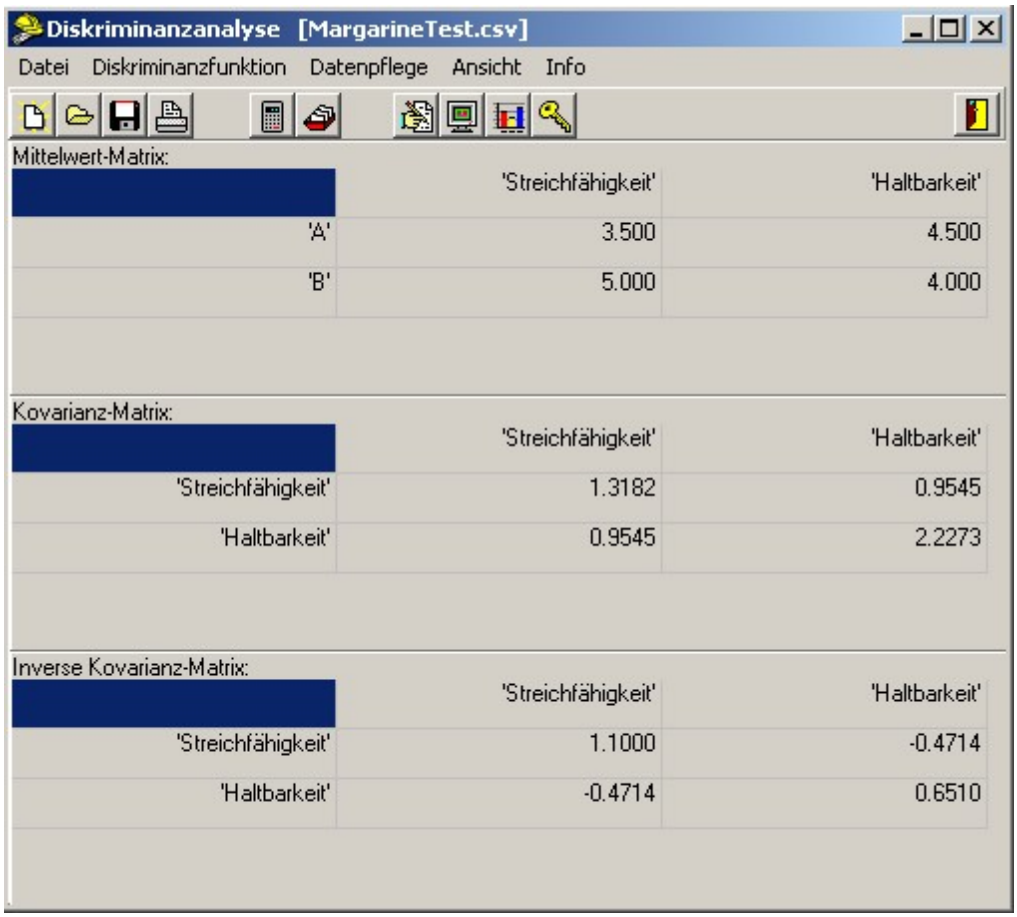


Abbildung 3.6: Endgültiges Erscheinungsbild der Kontroll-Ansicht

Die *Statistik-Seite* kam zuletzt. Hier sieht der Benutzer wieder die Gruppen-Informationen, die auch auf der Ergebnis-Seite zu finden sind. Außerdem wird hier zusätzlich ausgewertet, wieviele Datensätze der erhärteten Stichprobe richtig zugeordnet wurden. Die Auswertung erfolgt nach zwei verschiedenen Methoden: der Resubstitutions-Methode und der Leaving-one-out-Methode.
Nachdem bei den vorherigen Seiten jeweils ein aktueller Stand gezeigt wird, kommt auch hier ein Bild der endgültigen Version (Abb. 3.7).

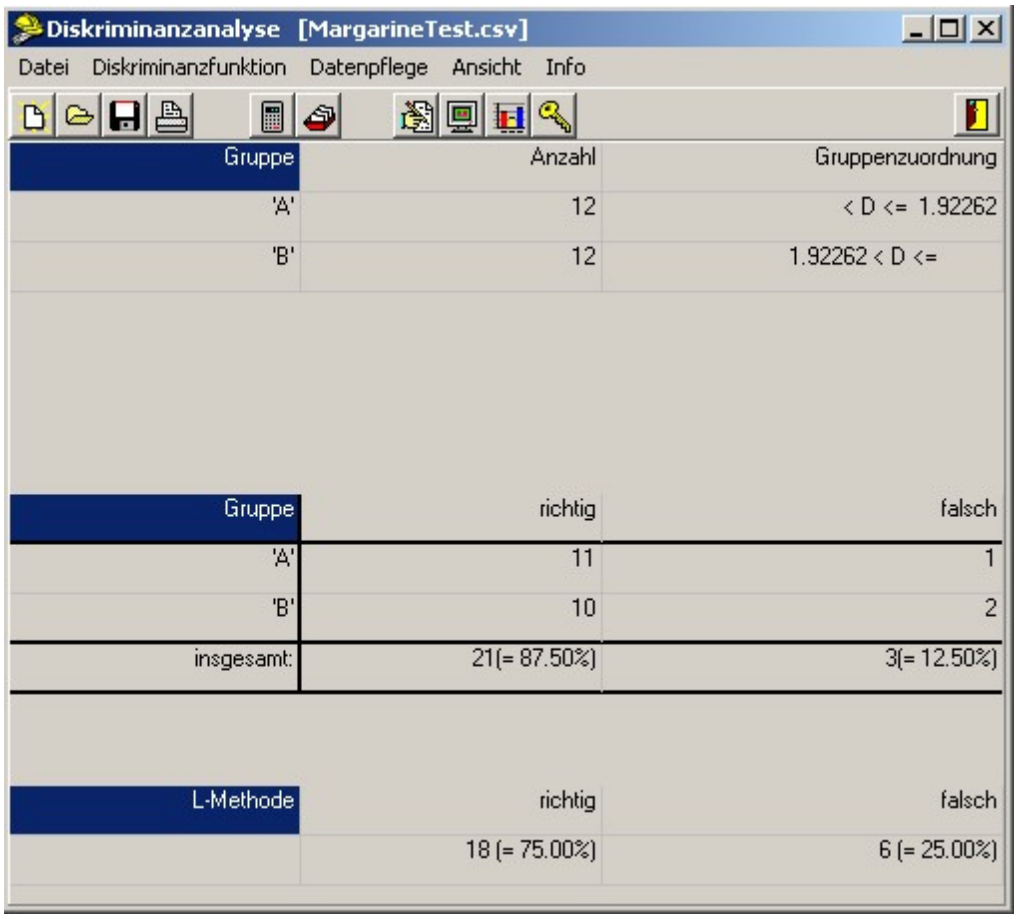


Abbildung 3.7: Endgültiges Erscheinungsbild der Statistik-Ansicht

3.2.2 Typischer Programm-Ablauf

Im typischen Fall hat der Anwender eine erhärtete Stichprobe, deren Daten entweder noch eingegeben werden müssen oder schon als csv-Datei vorliegen. Bei dieser Datei sind die Gruppen-Zuordnungen schon erfolgt.

Aufgrund dieser Daten wird die Diskriminanzfunktion berechnet, und die Zuordnungsvorschriften werden angezeigt. Jetzt hat der Anwender mehrere Möglichkeiten:

- Der Anwender ist mit dem Trennmaß zufrieden:
  - Der Anwender nutzt die Möglichkeit, die Diskriminanzfunktion auf die Daten anzuwenden, um so Ausgaben auf der Statistik-Seite zu erhalten.
  - Der Anwender speichert die eben berechnete Diskriminanzfunktion.
- Der Anwender ist noch nicht mit dem Trennmaß zufrieden:

- Der Anwender versucht, das Trennmaß durch neue Auswahl der zu berücksichtigenden Merkmale anzupassen.
- Der Anwender versucht, das Trennmaß durch Reduktion der einbezogenen Merkmale anzupassen.

Nachdem die Diskriminanzfunktion berechnet und evtl. gespeichert wurde, ist es jetzt möglich, sie auf andere Datensätze anzuwenden, bei denen noch keine Gruppenzugehörigkeit ermittelt wurde, um so die Möglichkeit der Prognose auszunutzen.

3.3 Realisierung (SourceCode-Beispiele)

3.3.1 Matrix-Invertierung

Bei Müller ([Mül69]) ist eine Algol-Prozedur zu finden, die die Invertierung einer Matrix mit Hilfe des Gauß-Jordan-Verfahren mit Pivotisierung beschreibt.

```
1   Inversion einer Matrix mit Berechnung der Determinante
2   ------------------------------------------------------------
3
4   Die Prozedur INVERS(N, A, EPS, ALARM, DELTA) invertiert die
5   N-reihige quadratische Matrix A durch Zeilenoperationen und
6   berechnet gleichzeitig den Wert ihrer Determinante.
7   Es bezeichnen
8   A       die zu invertierende Matrix,
9   N       die Ordnung der Matrix A,
10  EPS     die untere Schranke für ein Pivot-Element während des
11          Rechenganges und für die Determinante. Wird diese Schranke
12          unterschritten, so erfolgt ein Sprung zu der formalen Marke
13          ALARM.
14  DELTA   den Parameter, dem der Wert der Determinante zugewiesen
15          werden soll.
16
17  Nach Ablauf der Prozedur enthält das Feld A die invertierte Matrix.
18
19  Programmiersprache: ALGOL
20
21  PROCEDURE INVERS(N, A, ALARM, DELTA);
22  VALUE N;
23  INTEGER N;
24  REAL EPS, DELTA;
25  ARRAY A;
26  LABEL ALARM;
27  BEGIN
28          ARRAY B, C[1:N];
29          REAL W, Y;
30          INTEGER ARRAY Z[1:N];
31          INTEGER I, J, K, L, P;
32          DELTA:=1.0;
33          FOR J:=1 STEP 1 UNTIL N DO
34                  Z[J] := J;
35          FOR I:=1 STEP 1 UNTIL N DO
36          BEGIN
37                  K:= I;
38                  Q:=A[I, I];
39                  L:= I - 1;
40                  P:= I + 1;
41                  FOR J:= P STEP 1 UNTIL N DO
```

```
42                  BEGIN
43                          W:= A[I, J];
44                          IF ABS(W) GREATER ABS(Y) THEN
45                          BEGIN
46                                  K:= J;
47                                  Y:= W;
48                          END;
49                  END;
50                  DELTA := DELTA * Y;
51                  IF ABS(Y) LESS EPS THEN
52                          GOTO ALARM;
53                  FOR J:= 1 STEP 1 UNTIL N DO
54                  BEGIN
55                          C[J] := A[J, K];
56                          A[J, K] := A[J, I];
57                          A[J, I] := -C[J] * Y;
58                          B[J] := A[I,  J] := A[I, J] * Y;
59                  END;
60                  A[I, I] := Y;
61                  J := Z[I];
62                  Z[I] := Z[K];
63                  Z[K] := J;
64                  FOR K:=1 STEP 1 UNTIL L,P STEP 1 UNTIL N DO
65                          FOR J:= 1 STEP 1 UNTIL L,P STEP 1 UNTIL N DO
66                                  A[K, J] := A[K, J] - B[J] * C[K];
67          END;
68          FOR I:= 1 STEP 1 UNTIL N DO
69          BEGIN
70                  REPEAT:
71                          K := Z[I];
72                          IF K EQUAL I THEN GOTO ADVANCE;
73                          FOR J := 1 STEP 1 UNTEL N DO
74                          BEGIN
75                                  W := A[I,J];
76                                  A[I,J] := A[K,J];
77                                  A[K,J] := W;
78                          END;
79                          P := Z[I];
80                          Z[I] := Z[K];
81                          Z[K] := P;
82                          DELTA := -DELTA;
83                          GOTO REPEAT;
84                  ADVANCE:
85                          END;
86          END;
```

Diese Prozedur ist in ALGOL geschrieben. Sie musste noch nach ObjectPascal umgesetzt werden und sah danach folgender maßen aus:

```
1     -- Umsetzung: Matrix-Invertierung --
2
3   // Typ-Definition
4   type TMatr = Array of Array of Real;
5
6   // globale Variablen, die im Programm schon initialisiert
7   // wurden und hier verwendet werden
8   var m : integer;   // Anzahl der Merkmale wird nach
9                      // Auswahl-Dialog gesetzt
10      s     : TMatr; // Kovarianzmatrix aller Gruppen
11      s_inv : TMatr; // Inverse Kovarianzmatrix aller Gruppen
12
13  ------------------------------------------------------------
14
15  {** Invertierung einer Matrix und Berechnung der Determinante
16        sinv  - vor Berechnung: die zu invertierende Matrix,
17                nach der Berechnung: die invertierte Matrix
18        delta - Determinante}
19  procedure matrixinvert();
20  var i, j, k, l, p: integer;
21      w, y, eps, delta: real;
22      text : string;
23      z: array of integer;
24      b, c: array of real;
25  begin
26       // Initialisierungen
27       SetLength(sinv, m+1, m+1);
28       SetLength(b, m+1);
29       SetLength(c, m+1);
30       SetLength(z, m+1);
31       w := 0;
32       eps := 0;
33
34       // Matrix sinv mit den Werten von s belegen,
35       // damit im weitern Verlauf mit sinv gearbeitet
36       // werden kann
37       for i := 1 to m do
38           for j := 1 to m do
39               sinv[i, j] := s[i, j];
40
41       delta := 1.0;
42       for j := 1 to m do
43           z[j] := j;
44
```

```
45      // Berechnung startet
46      for i:= 1 to m do
47      begin
48          k:= i;
49          y:= sinv[i, i];
50          l := i - 1;
51          p := i + 1;
52
53          // Pivotisierung
54          for j := p to m do
55          begin
56              w := sinv[i, j];
57              if abs(w) > abs(y) then
58              begin
59                  k := j;
60                  y := w;
61              end;
62          end;
63          delta := delta * y;
64          if abs(w) < eps then
65              break;
66          if y = 0 then
67          begin
68              // Dem Anwender mitteilen, dass Merkmal x Probleme macht
69              // (in Zeile 63: delta := delta * y wird 0, wenn y=0)
70              Form1.memDF.Lines.Clear;
71              Form1.memDF.Lines.Add('Fehler in der Matrix-Invertierung');
72              text := 'bei Merkmal ' + IntToStr(i);
73              Form1.memDF.Lines.Add(text);
74              Form1.memDF.Visible := TRUE;
75              errmatrix := TRUE;
76              exit;
77          end
78          else
79              y := 1 / y;
80          for j := 1 to m do
81          begin
82              c[j] := sinv[j, k];
83              sinv[j, k] := sinv[j, i];
84              sinv[j, i] := -c[j] * y;
85              b[j] := sinv[i, j] * y;
86              sinv[i, j] := sinv[i, j] * y;
87          end;
88          sinv[i, i] := y;
89          j := z[i];
90          z[i] := z[k];
91          z[k] := j;
```

```
92          // Zeilen-Additionen durchführen
93          for k := 1 to l do
94          begin
95              for j := 1 to l do
96              begin
97                  sinv[k, j] := sinv[k, j] - b[j] * c[k];
98              end;
99              for j := p to m do
100             begin
101                 sinv[k, j] := sinv[k, j] - b[j] * c[k];
102             end;
103         end;
104         for k := p to m do
105         begin
106             for j := 1 to l do
107             begin
108                 sinv[k, j] := sinv[k, j] - b[j] * c[k];
109             end;
110             for j := p to m do
111             begin
112                 sinv[k, j] := sinv[k, j] - b[j] * c[k];
113             end;
114         end;
115     end;
116
117     // Zeilen wieder in ursprüngliche Reihenfolge bringen
118     for i := 1 to m do
119     begin
120         k := z[i];
121         while k <> i do
122         begin
123             for j:= 1 to m do
124             begin
125                 w := sinv[i, j];
126                 sinv[i, j] := sinv[k, j];
127                 sinv[k, j] := w;
128             end;
129             p := z[i];
130             z[i] := z[k];
131             z[k] := p;
132             delta := -delta;
133             k := z[i];
134         end;
135     end;
136 end;
```

3.3.2 Eigenwert-Problem

Bei Müller ([Mül69]) war außerdem eine Umsetzung für die Lösung des Eigenwert-Problems zu finden. Allerdings geht diese Variante davon aus, dass die ursprüngliche Matrix symmetrisch ist. Dabei stieß ich bei vielen Versuchen auf das Problem, dass die Zuordnungen nicht korrekt waren, obwohl die Kontroll-Matrizen stimmten. Nach einigen weiteren Versuchen die Zuordnungen zu berechnen, war der Fehler gefunden: die Daten, die durch die vorherigen Berechnungen meines Programms in der ursprünglichen Matrix stehen, sind nicht symmetrisch!
Da diese also nicht zu verwenden war, musste ein anderer Weg gefunden werden. Dabei stützte sich diese Methode auf die Informationen, die bei Faddejew ([FF64]) beschrieben wird. Hier das Ergebnis:

```
1     -- Umsetzung: Eigenwert-Problem --
2
3   // Typ-Definition
4   type TMatr = Array of Array of Real;
5
6   // Konstante
7   const G_EPS1 = 0.01
8
9   // globale Variablen
10  var m : integer;    // Anzahl der Merkmale wird nach
11                      // Auswahl-Dialog gesetzt
12      lambda : TMatr; // Kovarianzmatrix aller Gruppen
13
14  ------------------------------------------------------------
15  {** Matrixmultiplikation: m1*m2 --> mres.
16        nz1, ns1 = Zeilen-/Spaltenanzahl von m1
17        ns2 = Spaltenanzahl von m2}
18  procedure MMul(var m1,m2,mres:TMatr; nz1,ns1,ns2:integer);
19  var i,j,k: integer;
20       summe: real;
21       mz: TMatr;
22  begin
23      SetLength(mz, nz1+1, ns2+1);
24      fillchar(mz,sizeof(mz),0);
25      for i:=1 to nz1 do
26          for j:=1 to ns2 do
27          begin
28              summe:=0;
29              for k:=1 to ns1 do
30                  summe:= summe + m1[i,k] * m2[k,j];
31              mz[i,j]:=summe;
32          end;
33      mres:=mz;
34  end;
```

```
35  {** ClearM setzt alle Elemente der Matrix 'feld' auf 0}
36  procedure ClearM(feld: TMatr);
37  var i, j: integer;
38  begin
39      for i := 1 to m do
40          for j := 1 to m do
41              feld[i, j] := 0.0;
42  end;
43
44  {** rmaxw gibt das Maximum der beiden Werte zurück }
45  function rmaxw(wert1, wert2 : real) : real;
46  begin
47      if wert1 > wert2 then
48          Result := wert1
49      else
50          Result := wert2;
51  end;
52
53  {** MmaxEw bestimmt den maximalen Eigenwert und den zugehörigen
54      Eigenvektor einer Matrix.
55          maxew - größter Eigenwert
56          mev   - zum Eigenwert gehöriger Eigenvektor
57          m1    - Matrix für die das Eigenwertproblem gelöst
58                  werden soll
59          n     - m1 ist eine nxn-Matrix
60          anzit - Anzahl der Iterationen; falls das Verfahren
61                  versagt, wird anzit=0 zurückgegeben}
62  procedure MmaxEw(var m1,mev:TMatr; n: integer;
63                   var maxew:real; var anzit:integer);
64  var i, j, dl, eaz, mi, mj, mk: integer;
65       l, mxd, a, b, qs, qn, sum: real;
66       mz: TMatr;
67  begin
68      // Initialisierungen
69      SetLength(mz, n+1, 2);
70      dl:=0;
71      maxew:=0;
72      eaz:=0;
73
74      // es wurde nur 1 Merkmal gewählt
75      if n < 2 then
76      begin
77          maxew:=m1[1,1];
78          lambda[1,1]:=1.0;
79          anzit:=1;
80          exit;
81      end;
```

```
82      // es wurde mehr als 1 Merkmal gewählt
83      while dl<2 do
84      begin
85          // Initialisierung von lambda
86          ClearM(lambda);
87          j:=dl;
88          inc(dl);
89          eaz:=0;
90          anzit:=0;
91          l:=0;
92          for i:=1 to n do
93          begin
94              inc(j);
95              if odd(j) then
96              begin
97                  lambda[i,1]:=1.0;
98                  l:=l + 1.0;
99              end;
100         end;
101         l:=sqrt(l);
102         for i:=1 to n do
103             lambda[i,1]:=lambda[i,1] / l;
104
105         // Berechnung startet
106         repeat
107             // Initialisierungen
108             MMul(m1, lambda, mz, n, n, n);
109             inc(anzit);
110             l:=0;
111             mxd:=0;
112             qn:=0;
113             qs:=0;
114
115             // Berechnung der Vektor-Länge
116             for i:=1 to n do
117             begin
118                 a:=mz[i,1];
119                 l:=l + a * a;
120                 b:=lambda[1,1];
121                 if abs(a) > G_EPS1 then
122                 begin
123                     qs:=qs + b / a;
124                     qn:=qn + 1;
125                 end;
126             end;
127             l:=sqrt(l);
128
```

```
129             // Berechnung des Eigenwerts
130             if l > 0 then
131                 l:=1.0 / l
132             else
133                 l:=1.0;
134             if qn > 0 then
135                 maxew:=qs / qn;
136
137             // Berechnung von lambda
138             for i:=1 to n do
139             begin
140                 a:=mz[i,1] * l;
141                 b:=lambda[i,1];
142                 mxd:=rmaxw(mxd, abs(b-a));
143                 lambda[i,1]:=a;
144             end;
145
146             // Abbruchkriterien prüfen:
147             //
148             // wenn mehr als 40 Iterationen, setze eaz = 1
149             // (Prozedur wird normal beendet)
150             if anzit > 40 then
151                 eaz:=1;
152
153             // wenn mxd sehr klein
154             // (Prozedur wird normal beendet, Verfahren aber nicht
155             //  erfolgreich)
156             if mxd < 0.00001 then
157             begin
158                 eaz:=2;
159                 dl:=2;
160              end;
161         until eaz>0;
162     end;
163
164         // normal beendet oder abgebrochen?
165     if eaz<2 then
166         anzit:=0;
167 end;
```

3.4 Online-Hilfe

Wenn man gesagt bekommt, dass das Programm bitte auch eine Online-Hilfe haben soll, hält man die Erstellung zuerst für einfach. Die Probleme kommen aber, sobald man anfängt, die Hilfe zu erstellen.
Es beginnt bei der Struktur, die man dieser Hilfe geben möchte und geht bis zum Inhalt.

Daher hier eine Aufzählung der Probleme, die sich mir bei der Entwicklung der Hilfe gestellt haben:

- Welche Themenbereiche müssen in der Hilfe abgedeckt sein??
- Wie baut man die Hilfe so auf, dass ein späterer Anwender des Programms auch etwas mit der Hilfe anfangen kann??
- Wieviele Bilder kann ich in die Hilfe einbauen, ohne dass die Hilfe zu groß wird??
- Ist der Text, wie er jetzt ist, verständlich genug für einen Anwender, der das Programm noch nicht kennt?
- Ist der Text erklärend genug für jemanden, der sich mit der Materie ''Diskriminanzanalyse“ nur wenig auskennt?
- An welchen Stellen sollten Verweise auf andere Seiten plaziert werden??
- Wo sollte noch eine zusätzliche Information hinterlegt werden??

Eine auführliche Online-Hilfe zu erstellen, hätte den Zeitrahmen dieser Arbeit gesprengt. Dennoch ist hier dem Programm ein kurze – hoffentlich ausreichende – Online-Hilfe beigefügt.

3.5 Testphase

In der Testphase hatte ich vier verschiedene Datensätze zur Verfügung, auf die ich noch eingehen werde:

- MargarineTest.csv
- diskr01.csv
- diskr02.csv
- diskr03.csv

Da mir zu diesen Datensätzen durch die Literatur (Zwischen-)Ergebnisse bekannt waren, war es mir möglich, mit ihrer Hilfe mein Programm zu überprüfen.

3.5.1 "Margarine"-Test

Diese Daten wurden erhoben, weil ein Lebensmittelhersteller herausfinden wollte, ob sich zwei von ihm hergestellte Margarinemarken (Marke A bzw. B) bezüglich der Wahrnehmung ihrer Eigenschaften durch die Konsumenten unterscheiden.
Besonders interessierten ihn hierbei die Eigenschaften „Streichfähigkeit“ und „Haltbarkeit“.

Er führte daher eine Befragung von Stammanhängern beider Marken durch, bei der die Marken bezüglich der ausgewählten Merkmale auf einer 7-stufigen Skala bewertet werden sollten.

Dieses Beispiel wurde Backhaus ([BEPW96]) entnommen.

3.5.2 diskr01

Das folgende Beispiel stammt aus Röhr/Lohse/Ludwig ([RLL83]).

Wirtuk untersuchte 1976 die Abhängigkeit des psychopysiologischen Aktivierungsverlaufs und der Lernleistungen von sozialen und personalen Bedingungen. Aus den ursprünglich 235 Versuchspersonen wurden in diesem Beispiel die Teilpopulation „Arbeit in Gruppen mit persönlichkeitsheterogener, aber leistungshomogener Zusammensetzung“ ausgewählt.
Von den ursprünglich 235 Personen blieben so nur noch 24 Personen, die den drei Gruppen („extravertiert“, „ambivalent“ und „introvertiert“) zugeordnet sind.
Auch die sieben Merkmale lassen sich übergeordneten Bereichen zuordnen:

- Lerntätigkeit:
  - Lernzeitbedarf (Y_6)

  - subjektiv erlebte Arbeitsatmosphäre beim Lernen (Y_5)

- Lernleistung:
  - Leistung unmittelbar nach dem Lernen (Y_3)
  - Behaltensleistung nach vier Monaten (Y_4)
- erzieherischer Effekt der Gruppenarbeit:
  - Änderung der Einstellung zum Gruppenlernen (Y_1)
  - Änderung der Kooperationsbereitschaft (Y_2)
- Mathematische Leistungsvoraussetzung (Y_7)

Wer mehr zu dazu erfahren möchte sollte bei [Wir76] nachlesen.

3.5.3 diskr02

Untersucht werden die drei Irissorten „Iris setosa“, „Iris versicolor“ und „Iris virginica“. Von jeder Art wurden 50 Pflanzen untersucht. Von jeder Pflanze wurden vier Merkmale erfasst: Länge und Breite des Kelchblattes sowie Länge und Breite des Blütenblattes. Dies war mit 150 Datensätzen meine umfangreichste Test-Datei, deren Ergebnisse ich mit Hartung/Elpelt ([HE89]) vergleichen konnte.

3.5.4 diskr03

Diese Datei wurde erzeugt, um das Verhalten meines Programms bei Datensätzen zu prüfen, in denen einzelne Merkmale sich als Linearkombination anderer Merkmale ergeben. Dazu wurde diskr02.csv um zwei Spalten erweitert. Bei diesen Datensätzen ist die Determinante der Kovarianz-Matrix = 0. Es kann zu Instabilitäten in der Berechnung kommen, sodass das Programm keine Berechnung durchführen sollte.

3.6 Portierung

Wie schon vorher erwähnt, reizte mich der Versuch, mein Delphi-Programm auf Linux zu portieren. Mit Kylix sollte das kein Problem mehr sein, hieß es.

Das erste Problem lag darin, einen Rechner zu finden, auf dem sich Kylix installieren und starten ließ. Die Versuche mit SuSE 8.2, Red Hat 9.0 und Mandrake 9.1 blieben erfolglos. Abschließend versuchte ich es mit KNOPPIX; einer Live-Linux-CD, die auf Debian basiert; erfolgreich!
Damit war der erste Schritt getan. Es fehlte nur noch die Portierung des Quelltextes. Hierbei war es (fast nur) nötig, die Bibliotheken anzupassen. Sofern diese bei Kylix existierten, muss vor den Delphi-Bibliothek-Namen ein 'Q' gesetzt werden. Natürlich existieren momentan noch nicht alle Eigenschaften und Prozeduren / Funktionen in der Kylix-Umsetzung. So fehlten beispielsweise die Eigenschaften 'BevelInner', 'BevelOuter', 'BorderStyle' und 'IsControl'. Bei den Prozeduren / Funktionen fehlten für mein Programm 'GetSystemMenu', 'DeleteMenu', 'SendMessage', und 'HelpJump'. Bei manchen Prozeduren / Funktionen ist es notwendig, sie durch andere zu ersetzen. So war es in meinem Programm notwendig, die Funktion 'StrComp' durch 'CompareStr' zu ersetzen. 'CompareStr' existiert auch unter Delphi, so dass dies eine Änderung ist, die ohne Probleme durchgeführt werden kann. Wenn man bereit ist, solch "kleine Differenzen" hinzunehmen, ist es also durchaus möglich, Windows-Quelltext auf Linux zu portieren.
Ein weiteres Problem machte mir eine Prozedur, die den Menü-Eintrag 'Info' am rechten Rand des Fensters anordnen sollte. Weil ich hier auf Windows-API-Funktionen zurückgreife, ist es verständlicherweise nicht möglich, diese Prozedur unter Linux unbearbeitet zu lassen.
Bisher ist mir dafür leider noch keine Umsetzung gelungen.

Nach einigen Anpassungen und Verzicht auf Ausrichtung des Eintrags 'Info' war es möglich, die Anwendung aus der Kylix-IDE zu übersetzen. Aus der IDE war es auch möglich, das Programm auszuführen. Wenn man allerdings versucht, das Programm ohne die IDE zu starten, bricht es mit Fehlermeldungen ab.
Hoffentlich findet sich dafür noch eine Lösung!

3.7 Anwendungsbeispiele

Um einen kleinen Überblick über mein Programm zu geben, zeige ich hier folgend zwei Beispiele, wie der Ablauf aussehen kann.

3.7.1 "Margarine"-Test (MargarineTest.csv)

Wie in Abschnitt 3.5.1 (S. 50) bereits beschrieben, sollten Konsumenten zwei Margarinesorten in Bezug auf die Eigenschaften "Streichfähigkeit" und "Haltbarkeit" in einer 7-stufigen Skala bewerten.

Der Anwender öffnet die Datei und sieht dann Abbildung 3.8.

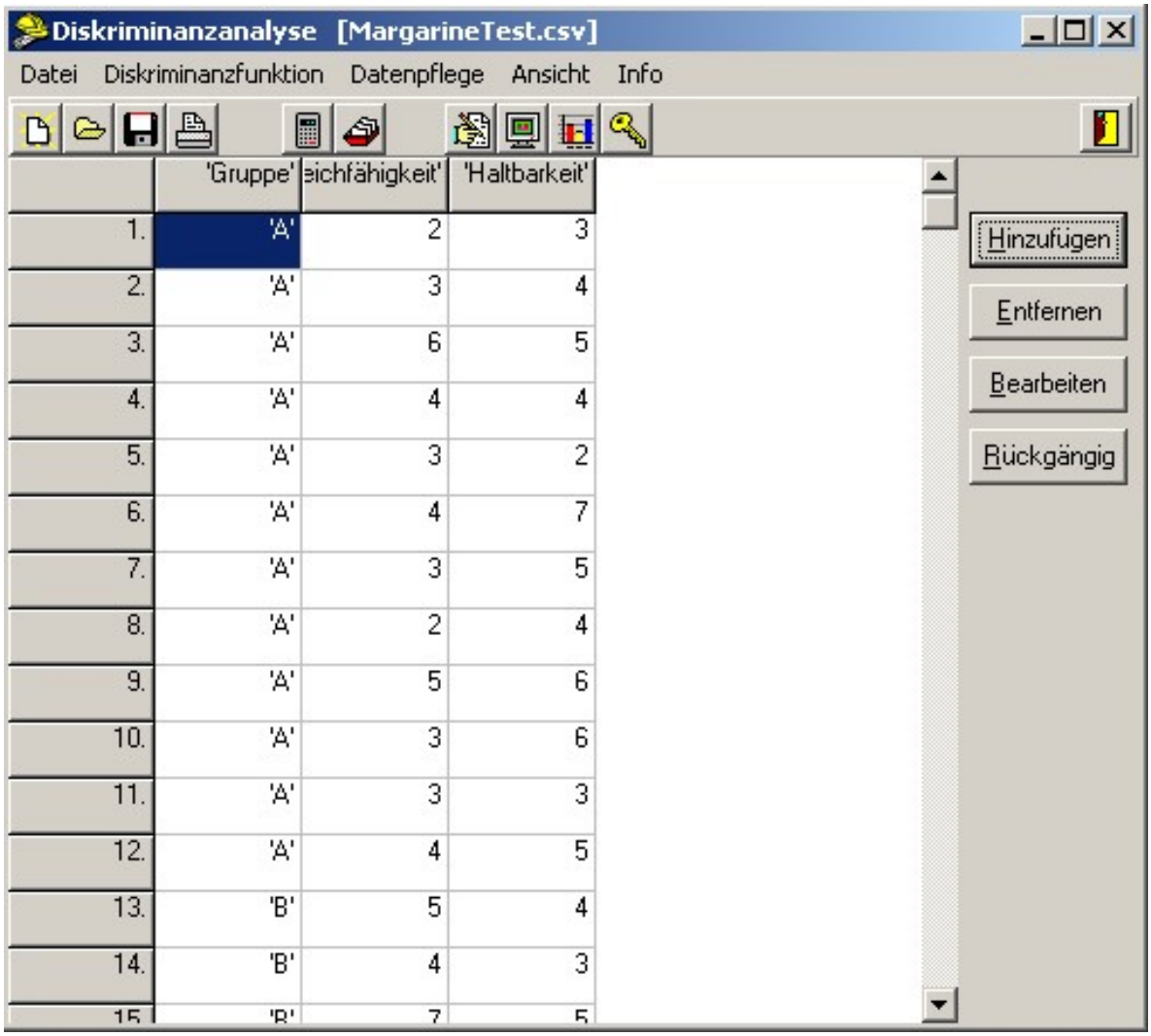


Abbildung 3.8: Margarine 1: Die geöffnete Datei wird in der Daten-Ansicht angezeigt.

Nach dem Start der Berechnung müssen die Merkmale ausgewählt werden, die in die Berechnung einfließen sollen (Abbildung 3.9).

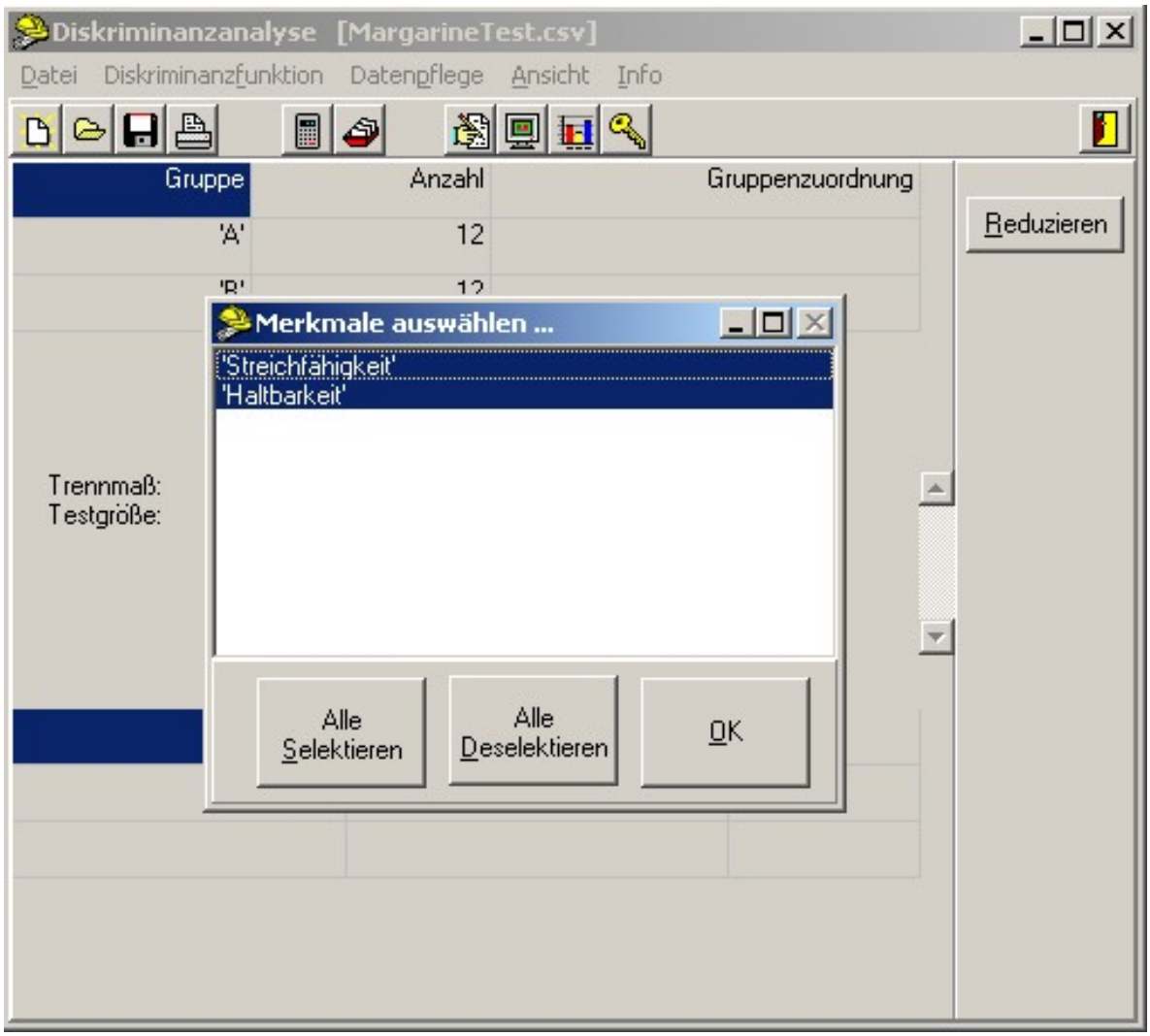


Abbildung 3.9: Margarine 2: Zum Start der Berechnung muss der Anwender die Merkmale auswählen.

Mit dem Ende der Berechnung sieht der Anwender die Ergebnis-Ansicht (Abbildung 3.10, S. 56).
Hier sieht er unter anderem die Diskriminanzfunktion und das Trennmaß. Wenn er mit diesem zufrieden ist, kann er die Funktion anwenden. Danach befindet sich der Anwender wieder auf der Daten-Ansicht. Hier sind zwei Spalten dazugekommen: für die neue Zuordnung und den Diskriminanzwert (Abbildung 3.11, S. 57).
Anschließend hat er noch die Möglichkeit die Statistik-Anzeige aufzurufen, um einen Überblick zu erhalten, wie gut die Funktion die Daten trennen kann (Abbildung 3.12, S. 58).

An diesem Beispiel wird deutlich, dass bei den Daten darauf geachtet werden sollte, dass keine gleichen Werte unterschiedlichen Gruppen zugeordnet werden. So ist der Datensatz (3—3) einmal der Gruppe A zugeordnet (Nr. 11) und auch der Gruppe B (Nr. 16). Genauso ist es auch bei den Datensätzen Nr. 4 und Nr. 17. Beide haben die Ausprägungen (4—4).
Datensatz Nr. 3 ist zwar nicht doppelt belegt, wird aber trotzdem fälschlicherweise der Gruppe B zugeordnet, weil diese Ausprägungen viel eher zur Gruppe B passen als zur Gruppe A.

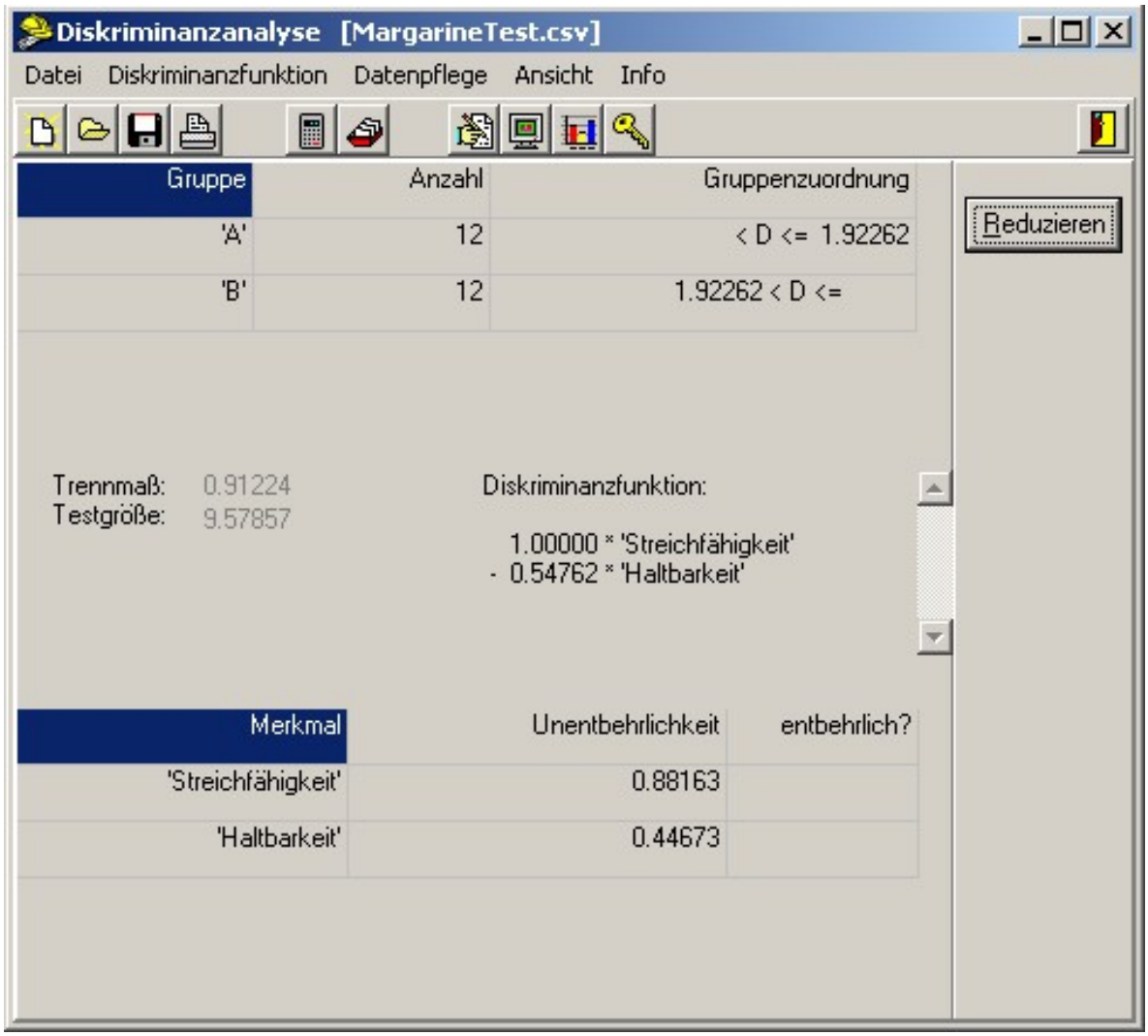


Abbildung 3.10: Margarine 3: Nach erfolgter Berechnung werden die Ergebnisse angezeigt.

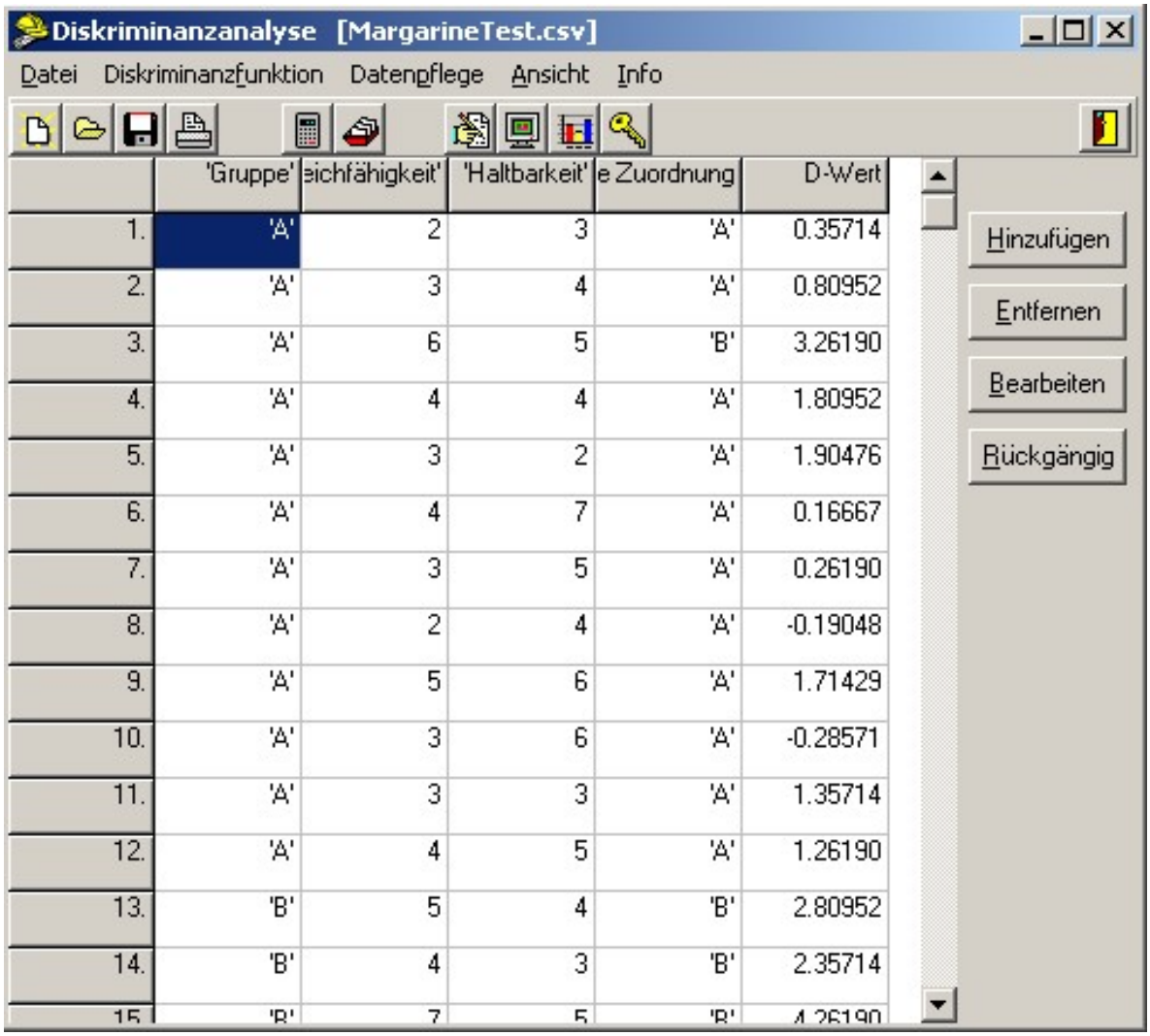


Abbildung 3.11: Margarine 4: Nach der Anwendung der Diskriminanzfunktion werden die neuen Zuordnungen in der Daten-Ansicht gezeigt.

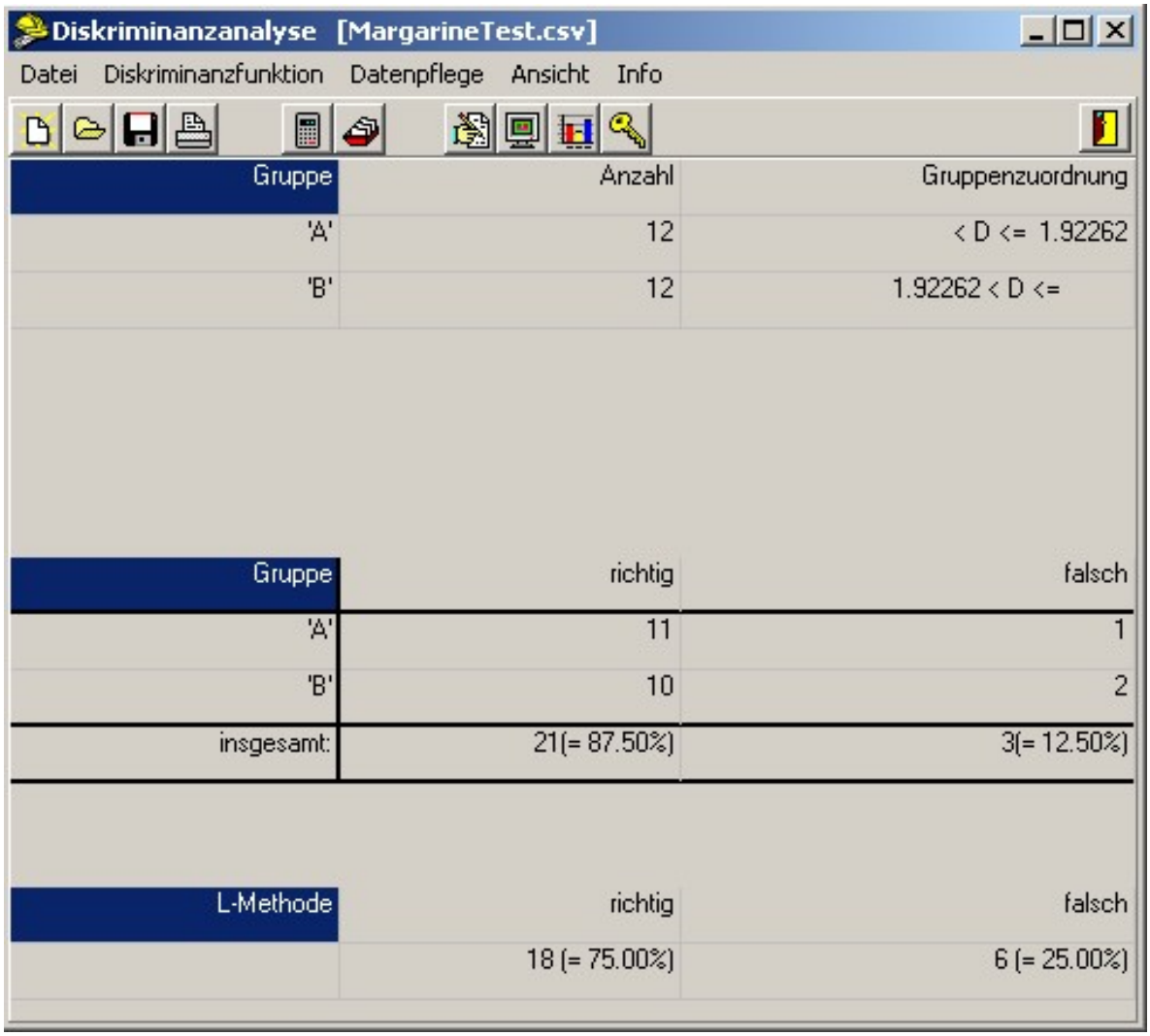
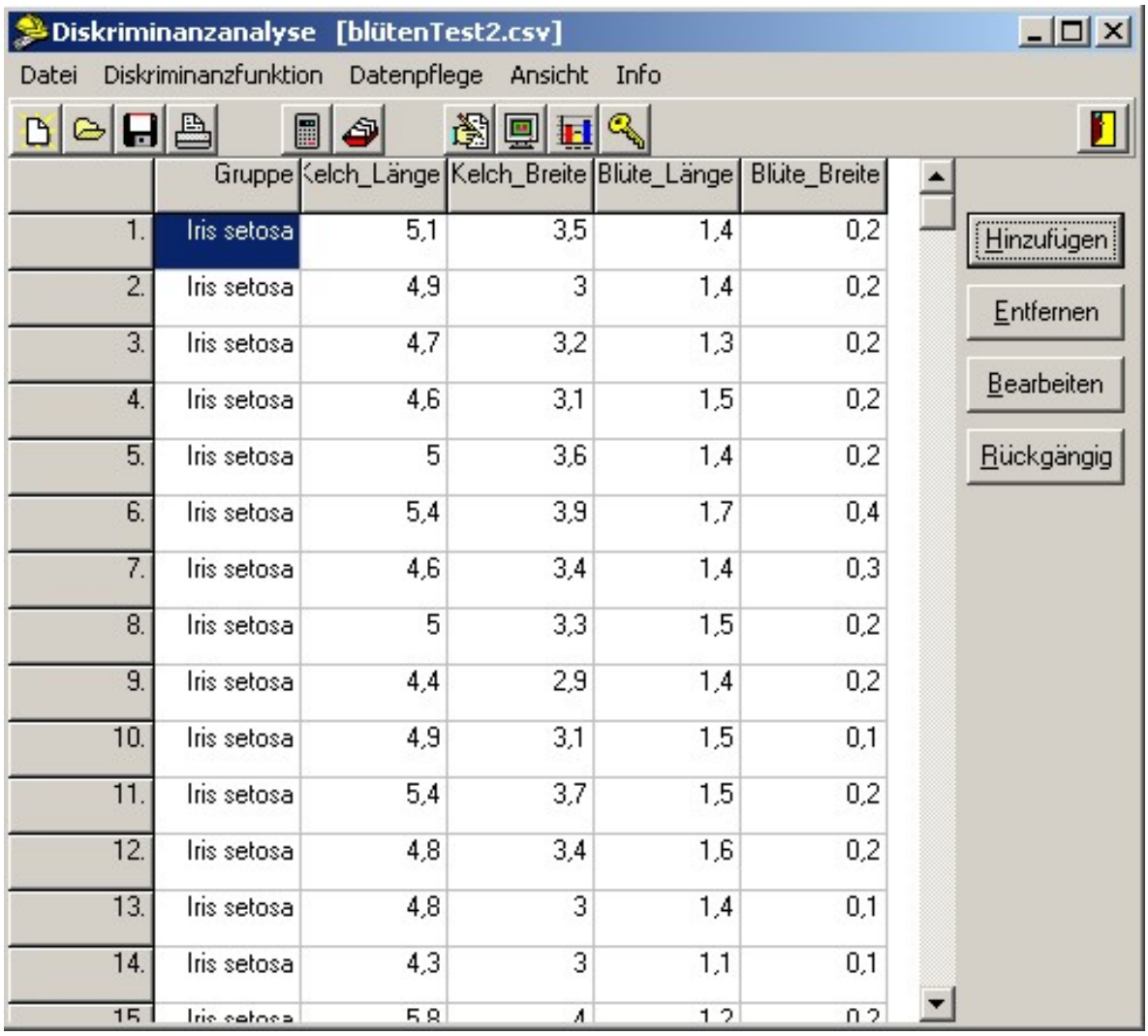


Abbildung 3.12: Margarine 5: Die Statistik zeigt, wie gut die Diskriminanzfunktion trennen kann.

3.7.2 Blüten (diskr02.csv)

Bei diesem Beispiel ist das Vorgehen sehr ähnlich. Nach dem Öffnen der Datei (Abbildung 3.13, S. 59)



	Gruppe	Kelch_Länge	Kelch_Breite	Blüte_Länge	Blüte_Breite
1.	Iris setosa	5,1	3,5	1,4	0,2
2.	Iris setosa	4,9	3	1,4	0,2
3.	Iris setosa	4,7	3,2	1,3	0,2
4.	Iris setosa	4,6	3,1	1,5	0,2
5.	Iris setosa	5	3,6	1,4	0,2
6.	Iris setosa	5,4	3,9	1,7	0,4
7.	Iris setosa	4,6	3,4	1,4	0,3
8.	Iris setosa	5	3,3	1,5	0,2
9.	Iris setosa	4,4	2,9	1,4	0,2
10.	Iris setosa	4,9	3,1	1,5	0,1
11.	Iris setosa	5,4	3,7	1,5	0,2
12.	Iris setosa	4,8	3,4	1,6	0,2
13.	Iris setosa	4,8	3	1,4	0,1
14.	Iris setosa	4,3	3	1,1	0,1
15.	Iris setosa	5,8	4	1,2	0,2

Abbildung 3.13: Blüten 1 - Die geöffnete Datei wird in der Daten-Ansicht angezeigt.

werden die Merkmale ausgewählt (Abbildung 3.14, S. 61)
und das Ergebnis angezeigt (Abbildung 3.15, S. 62).
Dann wurden die Zuordnungen bestimmt (Abbildung 3.16, S. 63)
und die Statistik erstellt (Abbildung 3.17, S. 3.17).

Dieses Vorgehen wurde danach noch zweimal durchgeführt, um einmal zu zeigen, wie sich die Auswahl der Merkmale auf die Zuordnungsbestimmung auswirken kann. Die Abbildungen 3.18 (S. 65) bis 3.20 (S. 67) zeigen eine Möglichkeit,

die Auswahl zu verringern und nur die Blüten-Werte in die Berechnung einfließen zu lassen.
Bei den Abbildungen 3.21 (S. 68) bis 3.23 (S. 70) wurden nur die Kelch-Werte berücksichtigt.
Vergleicht man nun die drei Statistik-Seiten, so fällt auf, dass die wenigsten falschen Zuordnungen bei der ersten Berechnung getroffen wurden. Am schlechtesten funktionierte die Trennung, wenn man nur die Daten der Blüten-Blätter zur Berechnung verwendete.

Merkmale	richtig zugeordnet		falsch zugeordnet	
	Resubstituion	Leaving-one-out	Resubstituion	Leaving-one-out
alle	148	99	2	51
Kelch	117	33	86	64
Blüte	50	50	100	100

Tabelle 3.1: Vergleich der Zuordnungs-Güte in Abhängigkeit von der Merkmalsauswahl

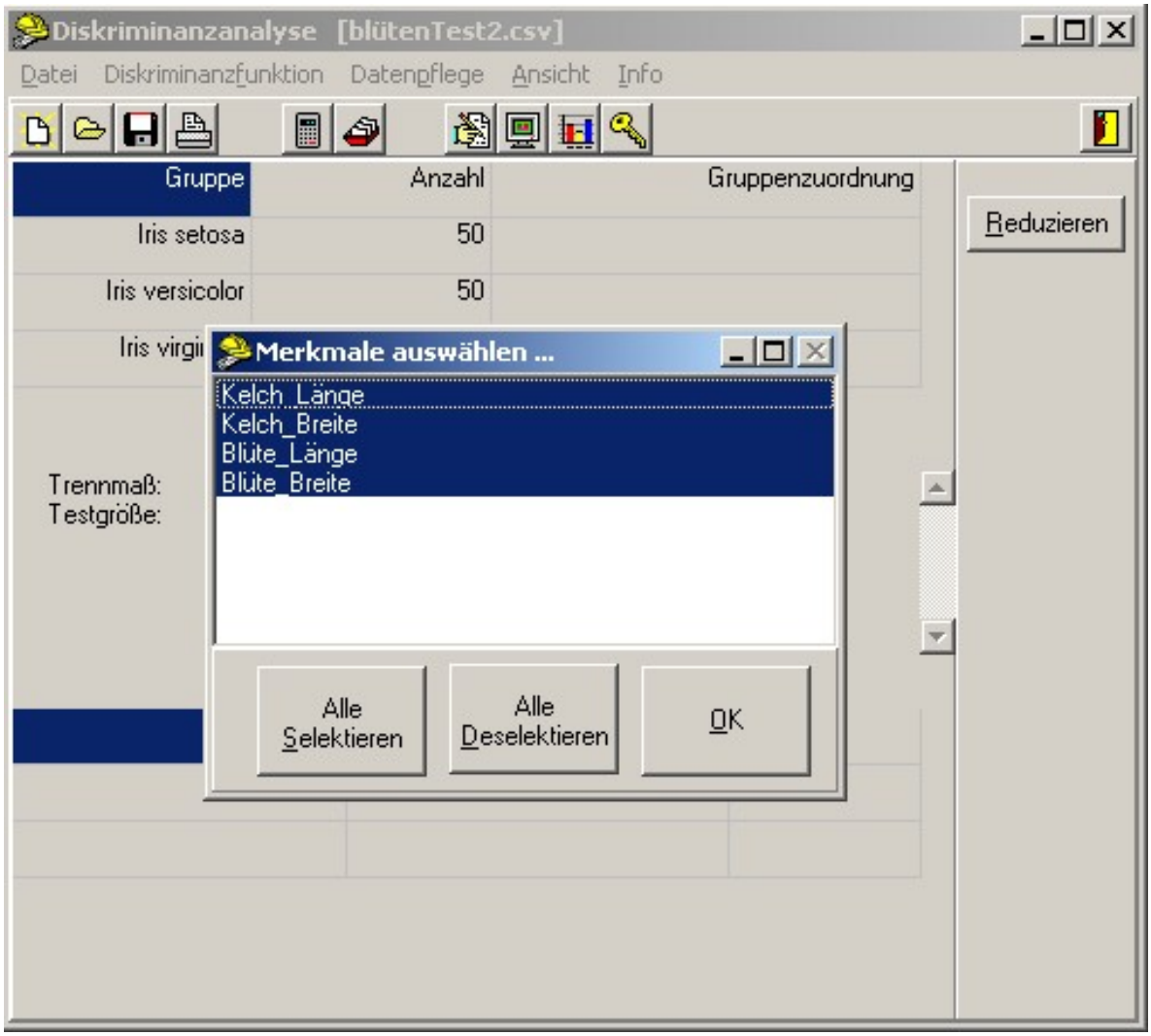


Abbildung 3.14: Blüten 2 - Zum Start der Berechnung muss der Anwender die Merkmale auswählen.

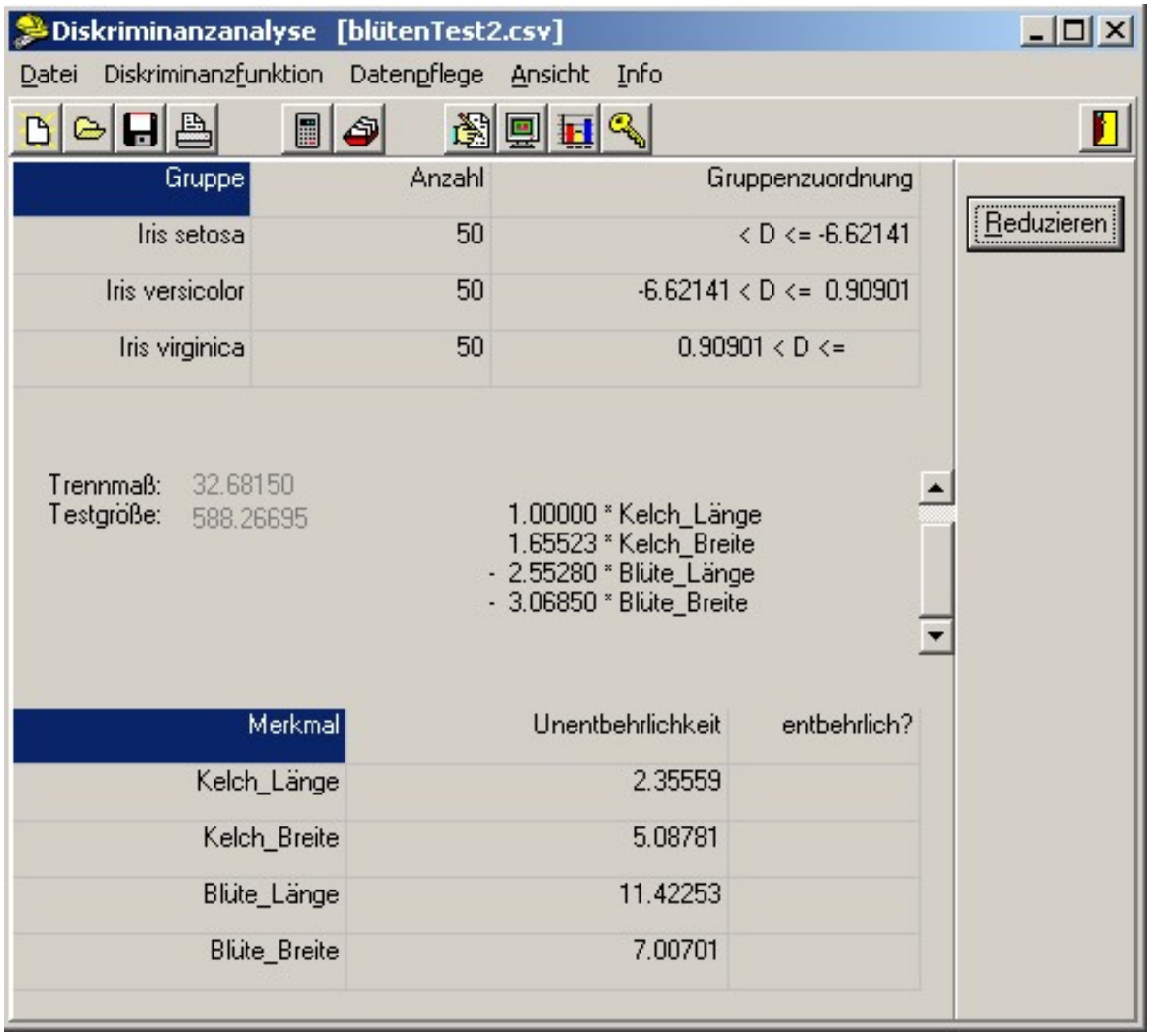
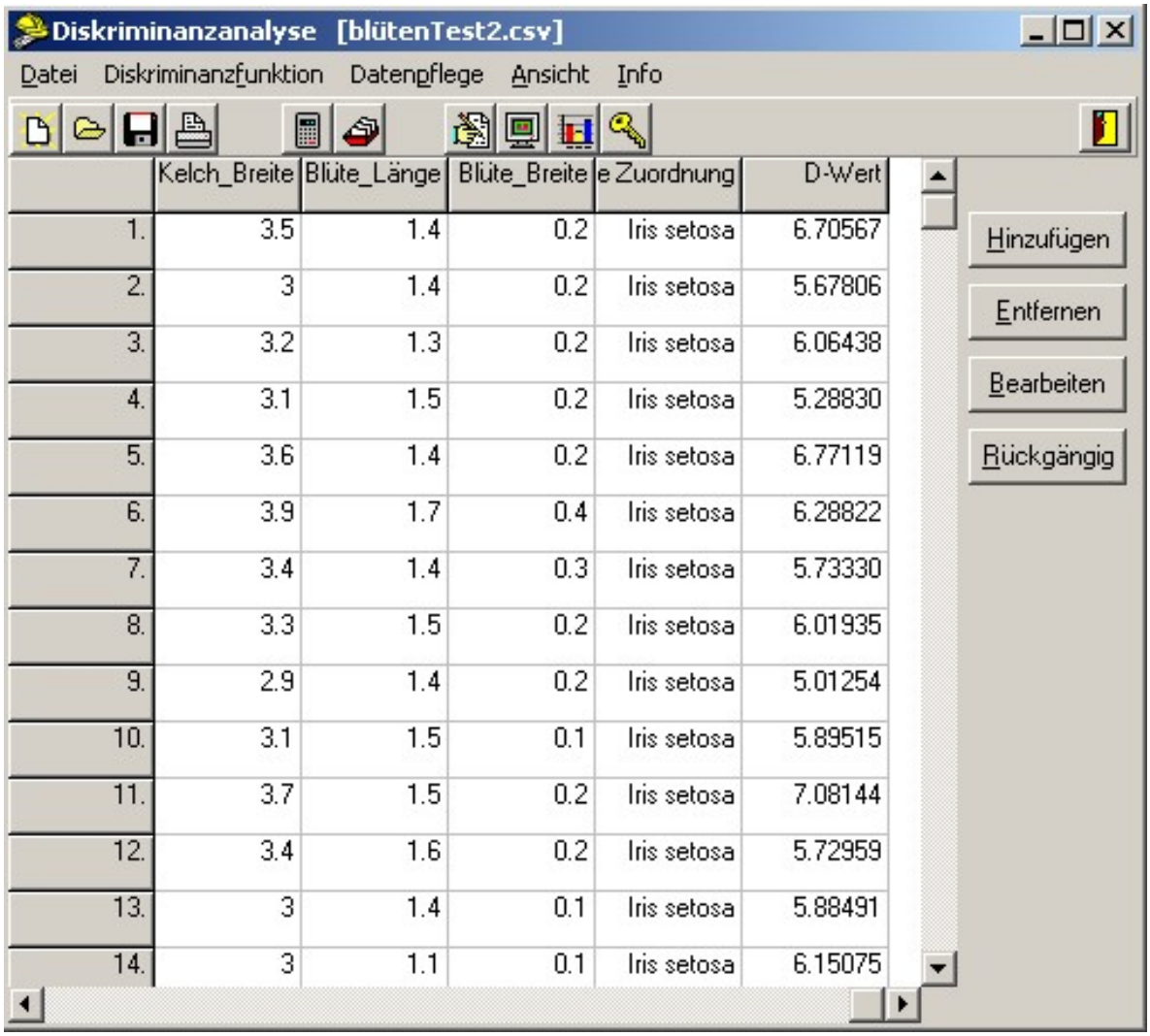


Abbildung 3.15: Blüten 3 - Nach erfolgter Berechnung werden die Ergebnisse angezeigt.



	Kelch_Breite	Blüte_Länge	Blüte_Breite	e Zuordnung	D-Wert
1.	3.5	1.4	0.2	Iris setosa	6.70567
2.	3	1.4	0.2	Iris setosa	5.67806
3.	3.2	1.3	0.2	Iris setosa	6.06438
4.	3.1	1.5	0.2	Iris setosa	5.28830
5.	3.6	1.4	0.2	Iris setosa	6.77119
6.	3.9	1.7	0.4	Iris setosa	6.28822
7.	3.4	1.4	0.3	Iris setosa	5.73330
8.	3.3	1.5	0.2	Iris setosa	6.01935
9.	2.9	1.4	0.2	Iris setosa	5.01254
10.	3.1	1.5	0.1	Iris setosa	5.89515
11.	3.7	1.5	0.2	Iris setosa	7.08144
12.	3.4	1.6	0.2	Iris setosa	5.72959
13.	3	1.4	0.1	Iris setosa	5.88491
14.	3	1.1	0.1	Iris setosa	6.15075

Abbildung 3.16: Blüten 4 - Nach der Anwendung der Diskriminanzfunktion werden die neuen Zuordnungen in der Daten-Ansicht gezeigt.

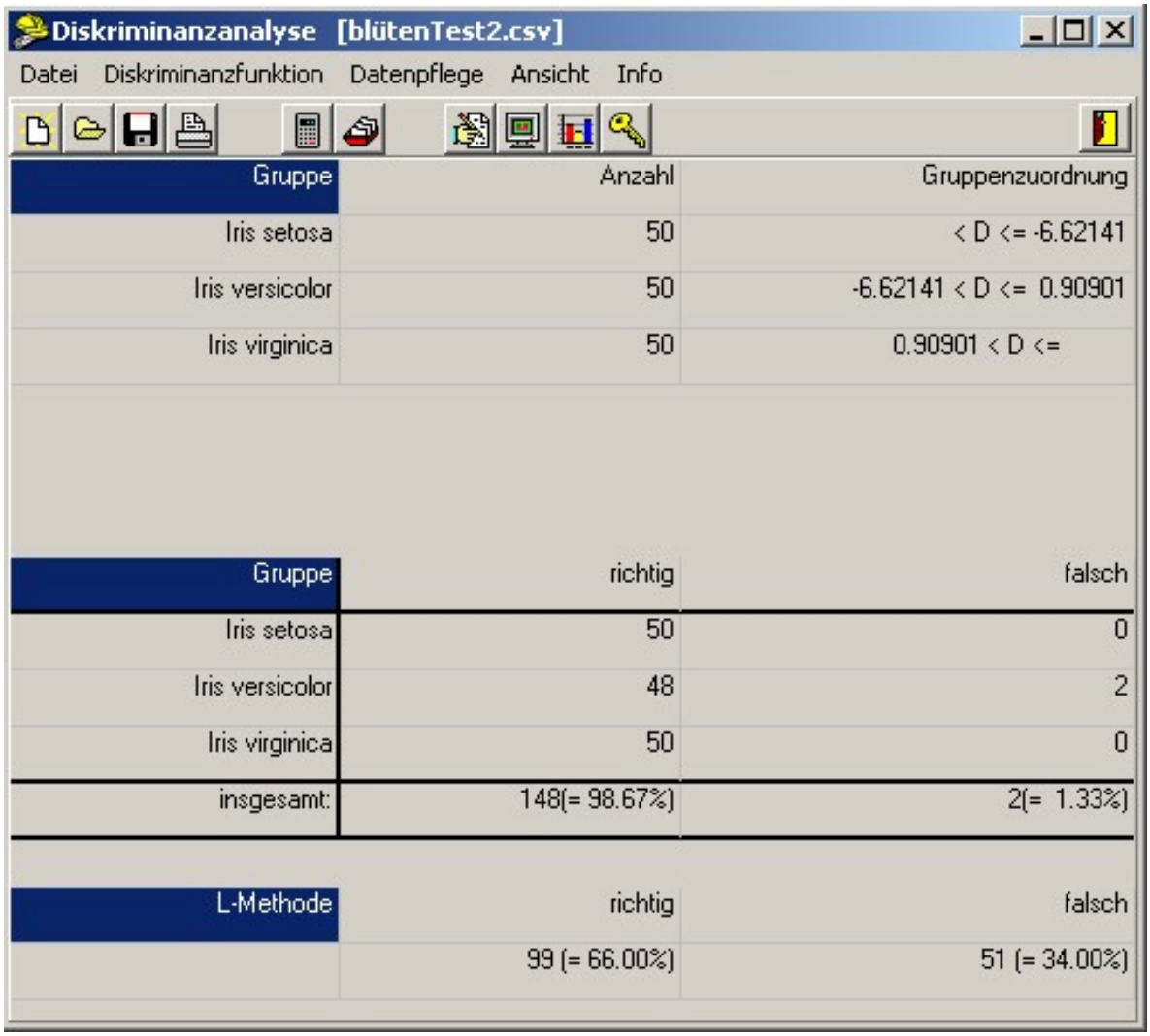


Abbildung 3.17: Blüten 5 - Die Statistik zeigt, wie gut die Diskriminanzfunktion trennen kann.

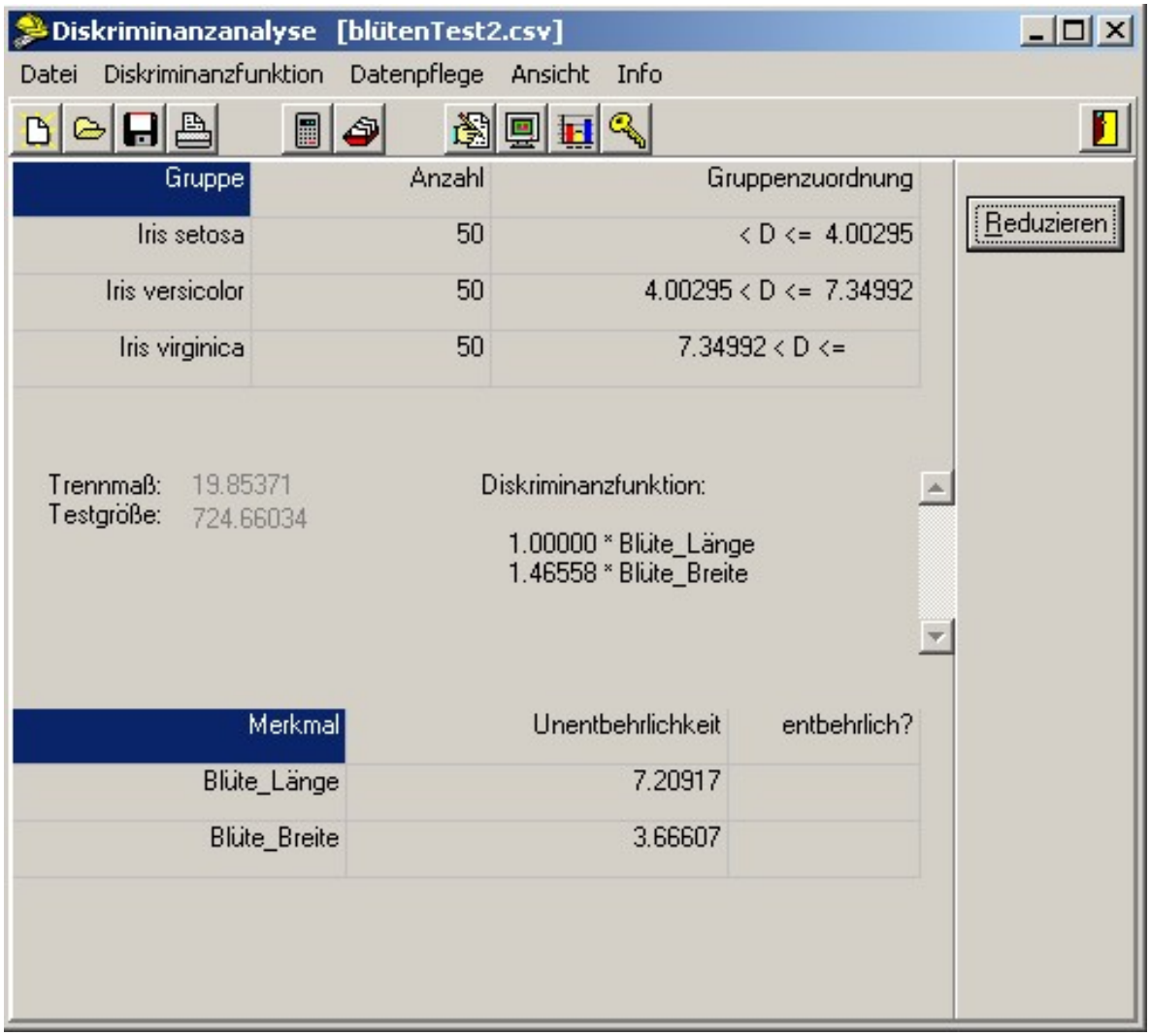
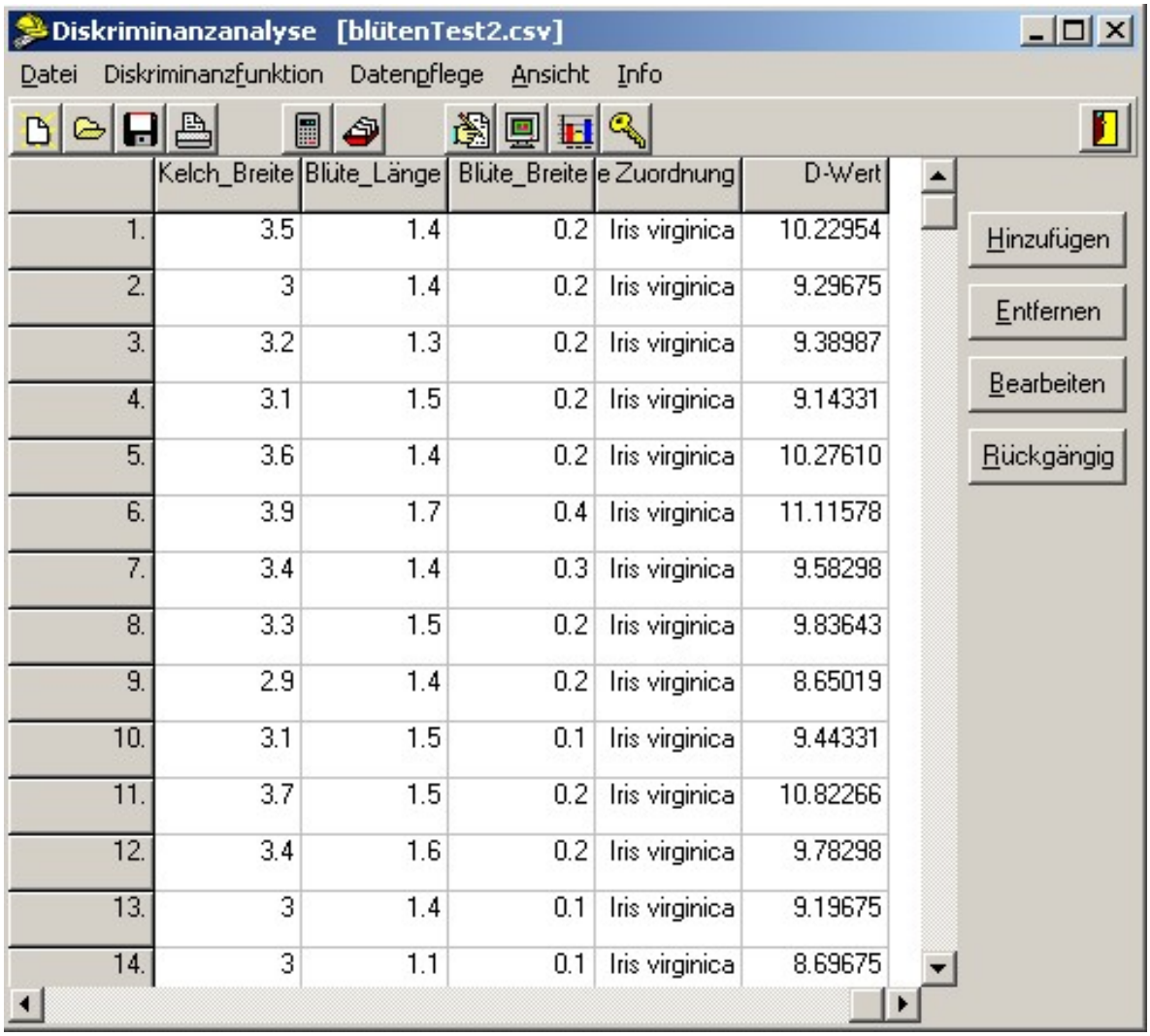
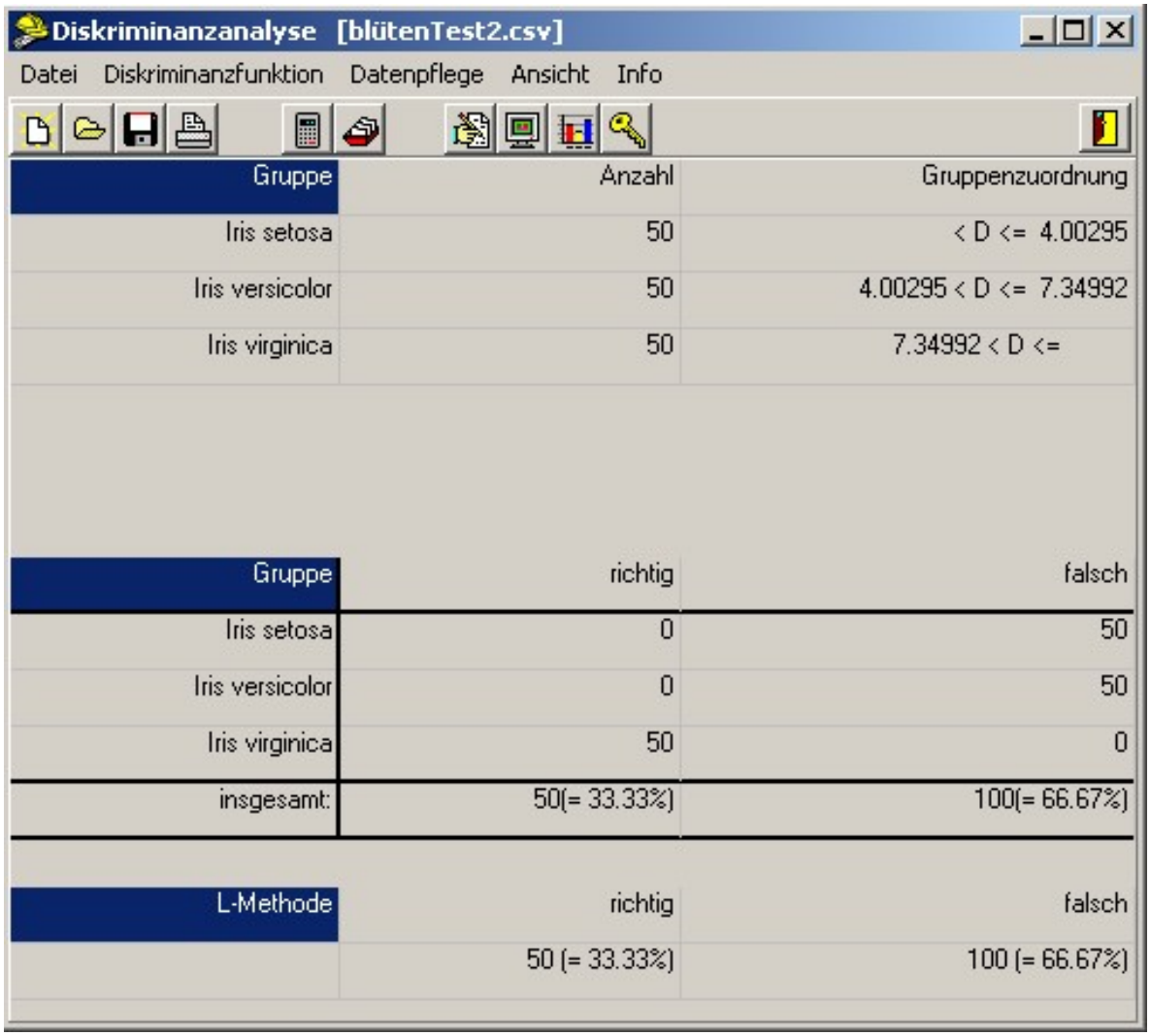


Abbildung 3.18: Blüten 6 - Nach erfolgter Berechnung werden die Ergebnisse angezeigt.



	Kelch_Breite	Blüte_Länge	Blüte_Breite	e Zuordnung	D-Wert
1.	3.5	1.4	0.2	Iris virginica	10.22954
2.	3	1.4	0.2	Iris virginica	9.29675
3.	3.2	1.3	0.2	Iris virginica	9.38987
4.	3.1	1.5	0.2	Iris virginica	9.14331
5.	3.6	1.4	0.2	Iris virginica	10.27610
6.	3.9	1.7	0.4	Iris virginica	11.11578
7.	3.4	1.4	0.3	Iris virginica	9.58298
8.	3.3	1.5	0.2	Iris virginica	9.83643
9.	2.9	1.4	0.2	Iris virginica	8.65019
10.	3.1	1.5	0.1	Iris virginica	9.44331
11.	3.7	1.5	0.2	Iris virginica	10.82266
12.	3.4	1.6	0.2	Iris virginica	9.78298
13.	3	1.4	0.1	Iris virginica	9.19675
14.	3	1.1	0.1	Iris virginica	8.69675

Abbildung 3.19: Blüten 7 - Nach der Anwendung der Diskriminanzfunktion werden die neuen Zuordnungen in der Daten-Ansicht gezeigt.



Gruppe	Anzahl	Gruppenzuordnung
Iris setosa	50	< D <= 4.00295
Iris versicolor	50	4.00295 < D <= 7.34992
Iris virginica	50	7.34992 < D <=

Gruppe	richtig	falsch
Iris setosa	0	50
Iris versicolor	0	50
Iris virginica	50	0
insgesamt:	50(= 33.33%)	100(= 66.67%)

L-Methode	richtig	falsch
	50 (= 33.33%)	100 (= 66.67%)

Abbildung 3.20: Blüten 8 - Die Statistik zeigt, wie gut die Diskriminanzfunktion trennen kann.

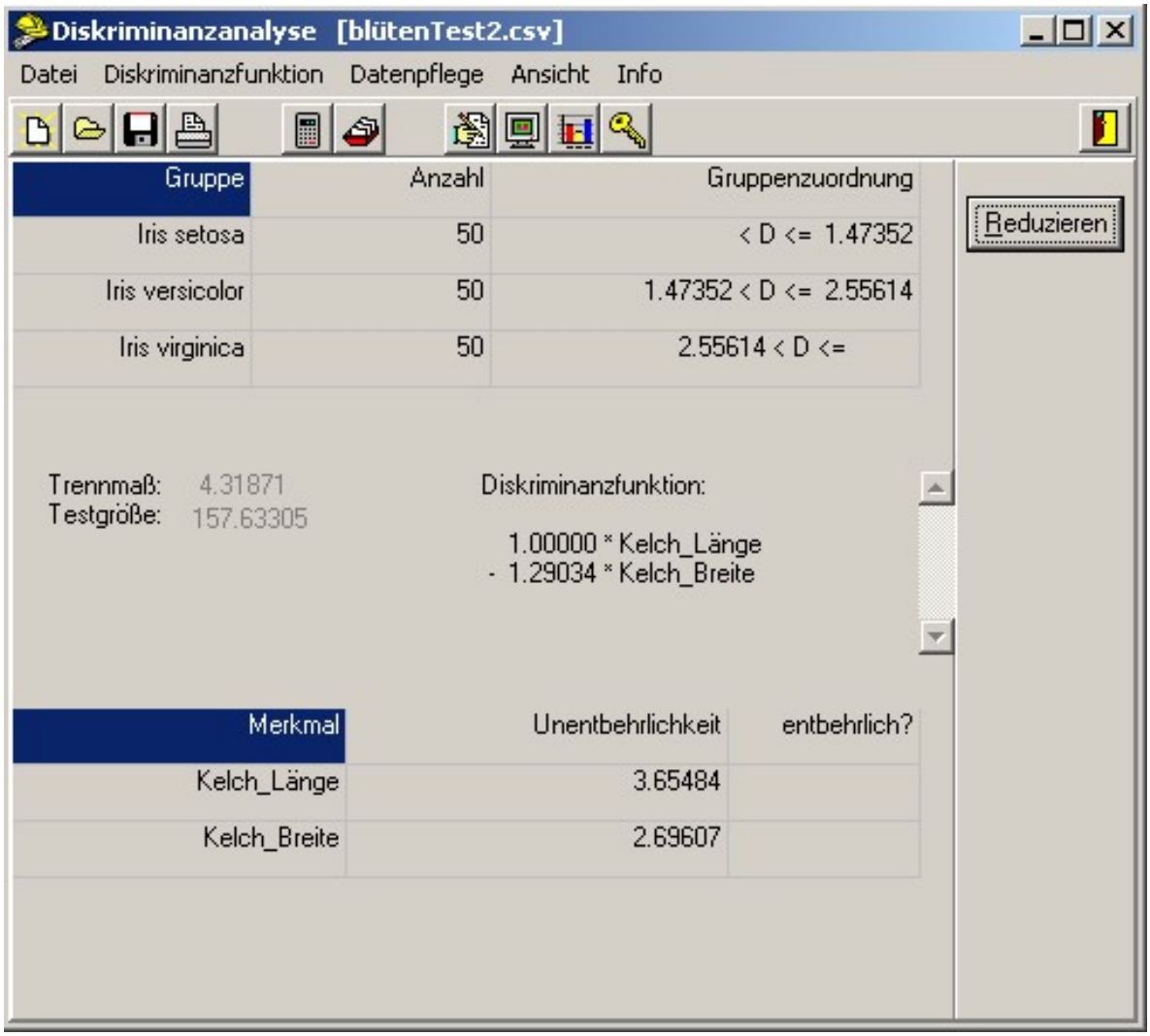
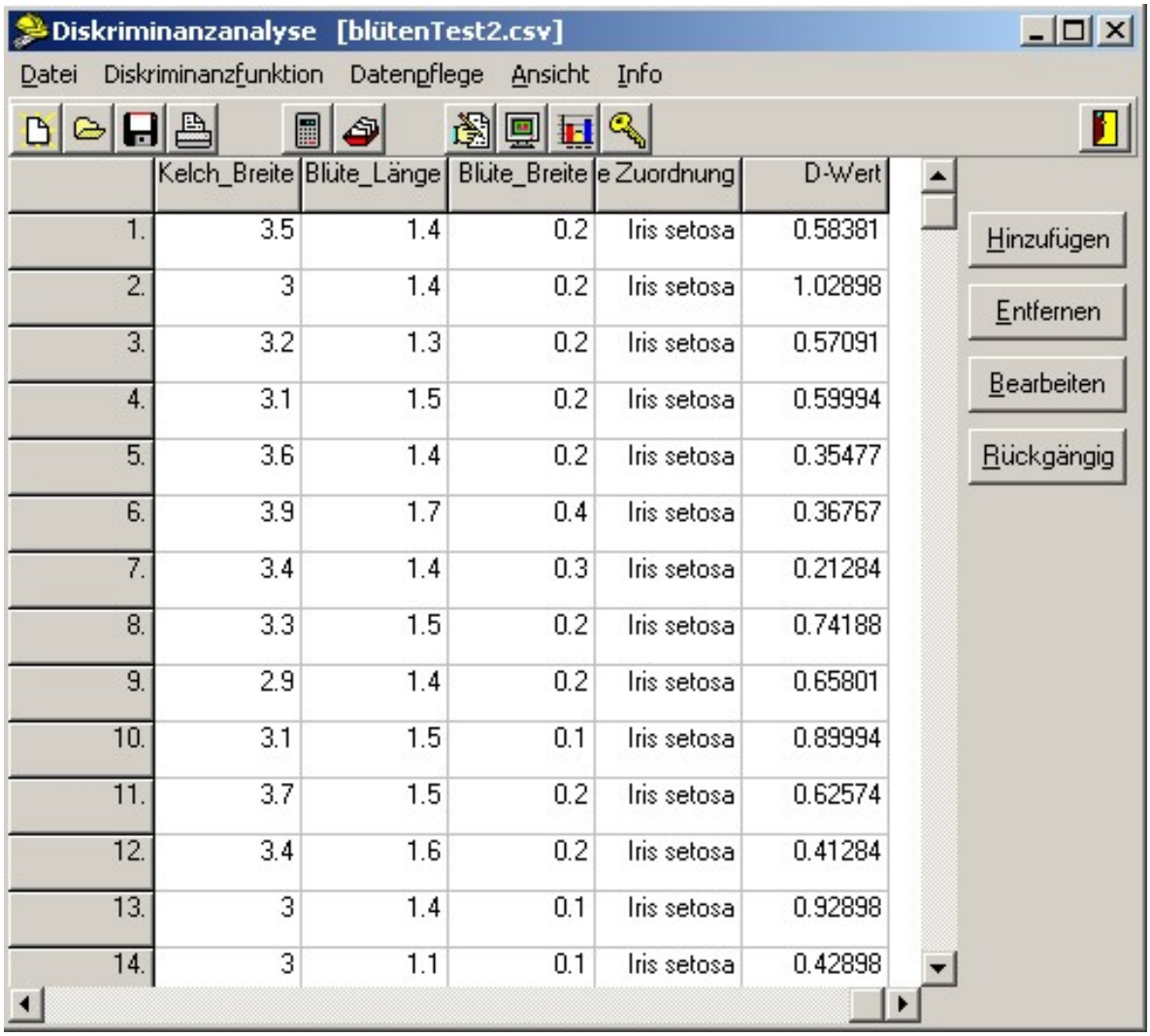
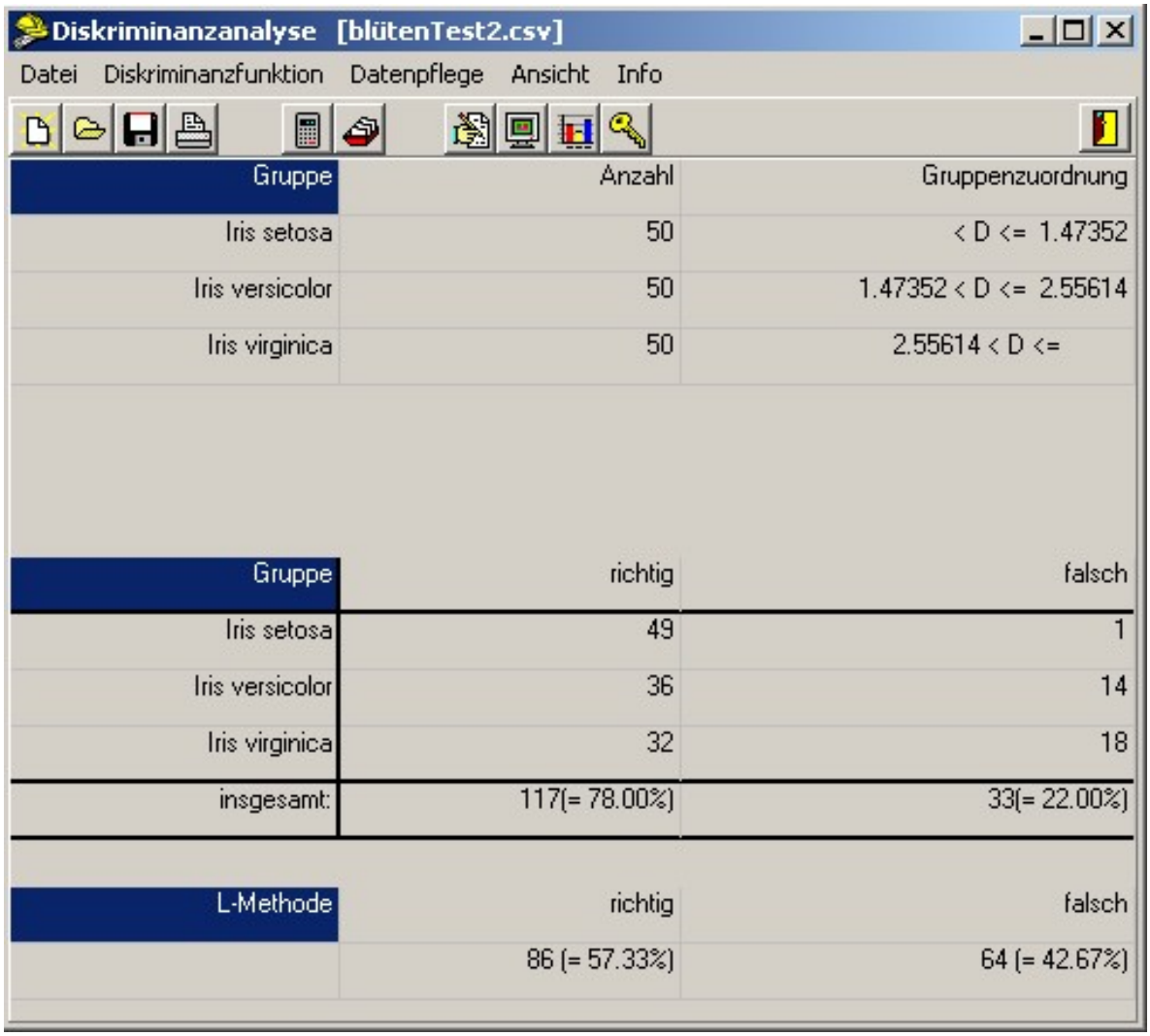


Abbildung 3.21: Blüten 9 - Nach erfolgter Berechnung werden die Ergebnisse angezeigt.



	Kelch_Breite	Blüte_Länge	Blüte_Breite	e Zuordnung	D-Wert
1.	3.5	1.4	0.2	Iris setosa	0.58381
2.	3	1.4	0.2	Iris setosa	1.02898
3.	3.2	1.3	0.2	Iris setosa	0.57091
4.	3.1	1.5	0.2	Iris setosa	0.59994
5.	3.6	1.4	0.2	Iris setosa	0.35477
6.	3.9	1.7	0.4	Iris setosa	0.36767
7.	3.4	1.4	0.3	Iris setosa	0.21284
8.	3.3	1.5	0.2	Iris setosa	0.74188
9.	2.9	1.4	0.2	Iris setosa	0.65801
10.	3.1	1.5	0.1	Iris setosa	0.89994
11.	3.7	1.5	0.2	Iris setosa	0.62574
12.	3.4	1.6	0.2	Iris setosa	0.41284
13.	3	1.4	0.1	Iris setosa	0.92898
14.	3	1.1	0.1	Iris setosa	0.42898

Abbildung 3.22: Blüten 10 - Nach der Anwendung der Diskriminanzfunktion werden die neuen Zuordnungen in der Daten-Ansicht gezeigt.



Gruppe	Anzahl	Gruppenzuordnung
Iris setosa	50	< D <= 1.47352
Iris versicolor	50	1.47352 < D <= 2.55614
Iris virginica	50	2.55614 < D <=

Gruppe	richtig	falsch
Iris setosa	49	1
Iris versicolor	36	14
Iris virginica	32	18
insgesamt:	117(= 78.00%)	33(= 22.00%)

L-Methode	richtig	falsch
	86 (= 57.33%)	64 (= 42.67%)

Abbildung 3.23: Blüten 11 - Die Statistik zeigt, wie gut die Diskriminanzfunktion trennen kann.

Kapitel 4

Zusammenfassung / kritische Würdigung

Zu Anfang meiner Arbeit stellte ich eine Liste der Punkte, die ich gerne realisieren wollte. Das Ergebnis war Tabelle 2.6 (S. 30). Um einen Überblick über die tatsächlich umgesetzten Punkte zu geben, ist sie hier nocheinmal abgebildet. Allerdings wurde sie um die Spalten "erfüllt" bzw. "nicht erfüllt" erweitert.

Anforderung	**erfüllt**	**nicht erfüllt**
Beschreibung mit Basisstatistiken		
Mittelwerte	√	
Häufigkeiten	√	
Teststatistiken		X
Bestimmung der Zuordnungsregeln	√	
Bewertung der Güte der Zuordnungsregeln		
Leaving-one-out	√	
Train-and-Test		X
Resubstitution	√	
Zuordnung neuer Fälle möglich	√	
Reduktion der Merkmale		
manuell	√	
automatisch		X
Auswahl der zu betrachtenden Merkmale	√	
Druck der Ergebnisse	√	
Graphische Darstellung der Resultate		X

Tabelle 4.1: Anforderungen an ein Diskriminanzanalyse-Programm und deren Umsetzung

Der Hauptgrund für das Fehlen diverser Teilaufgaben war Zeitmangel. Anfangs hielt ich die Zeit für ausreichend, mit Beginn der Programmier-Phase stellte sich jedoch heraus, dass ich wohl doppelt so viel Zeit benötigt hätte alles zu realisiern, als mir zur Verfügung stand. Aus diesem Grund implementierte ich

zuerst die für die Berechnung unerlässlichen Teile wie – die Basis-Statistiken, – die Bestimmung der Zuordnungsregel und – die Auswahl der zu betrachtenden Merkmale. Jetzt kam die Zuordnung der Daten, aus denen ich die Funktion berechnet hatte. Nachdem diese funktionierte war auch die Zuordnung neuer Fälle kein Problem mehr.
Bis zur Zuordnung war jetzt also die Berechnung fertiggestellt; jetzt konnten also "Zusätze" kommen.
Um einen Überblick zu erhalten, wie gut die Zuordnung funktionierte, versuchte ich mich an der Resubstitutionsmethode. Da diese aber, wie in der Literatur mehrfach zu lesen ist, als einzige Überprüfungsmethode nicht ausreicht, musste noch mindestens eine weitere Methode angewandt werden. Ich entschied mich für die Leaving-one-out-Methode. Diese war auch noch in der gegebenen Zeit fertig.
Alles Weitere musste jedoch "gestrichen" werden, da das den mir zur Verfügung stehenden Zeitrahmen gesprengt hätte. Die Testphase braucht eben auch noch Zeit. Einige Tests waren zwar schon parallel zur Entwicklung gelaufen, diese beschränkten sich jedoch immer nur auf Teilgebiete und nie auf das komplette Programm.

Glossar

abhängige Variable Größe, deren Veränderung beobachtet wird

Ausprägung festgestellter Merkmalswert

dependente Verfahren untersucht werden der Einfluß einer Gruppe von Variablen auf eine Reihe interessierender, beobachteter Merkmale ODER der Zusammenhang zwischen zwei Merkmalsgruppen (Regressionsanalyse, Varianzanalyse, Kovarianzanalyse, Korrelationsanalyse)

diskrete Merkmale können nur abzählbar viele Werte annehmen (z.B. Geschlecht, Anzahl)

Eigenvektor Ein vom Nullvektor verschiedener Vektor x ist ein Eigenvektor der quadratischen Matrix A, wenn es einen Skalar λ gibt, mit $Ax = \lambda x$

Eigenwert Ein Skalar heißt Eigenwert der quadratischen Matrix A, wenn das lineare Gleichungssystem $(A-\lambda E)x = 0$ eine nichttriviale Lösung besitzt.

Eigenwertaufgabe Es seien A, B (n, n)-Matrizen. Dann versteht man unter der zugehörigen Eigenwertaufgabe das Problem: Bestimme λ so, dass das lineare Gleichunggsysstem $(A - \lambda B)x = 0$ eine nichttriviale Lösung besitzt.
Zur Unterscheidung nennt man dies auch die *allgemeine Eigenwertaufgabe* und den Spezialfall $B = E$ die *spezielle Eigenwertaufgabe*.

Einheitsmatrix Die (n,n)-Matrix mit den Elementen

$$e_{ij} = \begin{cases} 1 & \text{, falls} \quad i = j \\ 0 & \text{, falls} \quad i \neq j \end{cases}$$

heißt Einheitsmatrix und wird mit E bezeichnet.

elementare Zeilenoperationen Die elementaren Zeilenoperationen sind:

1. Vertauschen zweier Zeilen
2. Multiplikation einer Zeile mit einem von 0 verschiedenen Skalar
3. Addition eines Vielfachen einer Zeile mit einer anderen Zeile

erhärtete Stichprobe Stichprobe, bei der der Klassifikationsvektor und die zugehörige Gruppenvariable schon bestimmt wurden

homogemes Gleichungssystem Ein lineares Gleichungssystem $Ax = b$ heißt homogen, wenn b der Nullvektor ist.

IDE **I**ntegrated **D**evelopment **E**nvironment

interdependente Verfahren alle beobachteten bzw. erhobenen Merkmale sind „gleichberechtigt“; Aussagen und Ergebnisse gelten für alle Merkmale auf gleiche Weise (Clusteranalyse, Faktoranalyse)

Inverse einer Matrix Eine Matrix B heißt Inverse der quadratischen Matrix A, wenn $BA = AB = E$ gilt.

invertierbare Matrix Eine quadratische Matrix heißt invertierbar, wenn sie eine Inverse besitzt.

Klassifikationsvektor Darstellungsform für das Meßergebnis einer Person

KNOPPIX eine auf Debian basierende Live-Linux-CD (http://www.knopper.net/knoppix)

linear abhängig Die Vektoren $v^1, \dots v^m$ heißen linear abhängig, wenn das lineare Gleichungssystem $\alpha_1 v^1 + \cdots + \alpha_m v^m = 0$ eine nichttriviale Lösung besitzt, d.h. eine Lösung $\alpha_1, \dots, \alpha_m$, für die wenigstens ein α_j von Null verschieden ist.

linear unabhängig Die Vektoren $v^1, \dots v^m$ heißen linear unabhängig, wenn das lineare Gleichungssystem $\alpha_1 v^1 + \cdots + \alpha_m v^m = 0$ nur die triviale Lösung besitzt, d.h. aus dieser Gleichung folgt, dass alle α_j gleich Null sind.

Matrix Eine (m, n)-Matrix A ist ein rechteckiges Schema reeller (oder komplexer) Zahlen mit m Zeilen und n Spalten.

Merkmal Größe, die gemessen wird bzw. nach der gefragt wird

metrische Skala Abstände zwischen Werten sind interpretierbar (z.B. Größe, Gewicht)

nichtsinguläre Matrix Eine (n, n)-Matrix heißt nichtsingulär oder regulär, wenn sie den Rang n besitzt.

normale Matrix Die (n, n)-Matrix A heißt normal, falls $AA^T = A^T A$.

Nominal-Skala Merkmale haben keine Reihenfolge und sind nicht vergleichbar (z.B. Beruf)

Ordinal-Skala Merkmale haben eine Reihenfolge, unterscheiden sich aber in der Intensität (z.B. Noten)

quadratische Matrix Eine (m, n)-Matrix heißt quadratisch, wenn m=n gilt, wenn also die Zeilenzahl und die Spaltenzahl übereinstimmen.

quantitative Merkmale in *Größe* unterscheidbar (z.B. Alter, Gewicht etc.)

qualitative Merkmale in *Art* unterscheidbar (z.B. Farbe, Beruf etc.)

Rang einer Matrix Ist A eine (m, n)-Matrix, so ist die maximale Anzahl linear unabhängiger Zeilen gleich der maximalen Anzahl linear unabhängiger Spalten. Diese Maximalzahl heißt: der Rang von A.

reguläre Matrix Eine (n, n)-Matrix heißt regulär, wenn ihr Rang n ist.

singuläre Matrix Eine (n, n)-Matrix heißt regulär, wenn ihr Rang kleiner als n ist.

Skalar Unter Skalaren versteht man im Zusammenhang mit reellen Vektorräumen die reellen Zahlen, im Zusammenhang mit komplexen Vektorräumen die komplexen Zahlen.

stetige Merkmale können beliebigen Wert in einem bestimmten Bereich annehmen (z.B. Länge, Lebensdauer)

symmetrische Matrix Die quadratische Matrix A heißt symmetrisch, wenn sie mit ihrer transponierten übereinstimmt.

transponierte Matrix Ist $A = (a_{ij})$, $(i = 1, \ldots, m,\ j = 1, \ldots, n)$ eine (m, n)-Matrix, so heißt die (n, m)-Matrix B mit den Elementen $B = (a_{ji})$, $(i = 1, \ldots, m,\ j = 1, \ldots, m)$ die transponierte Matrix von A und wird mit $A^T := B$ bezeichnet.

unabhängige Variable Einflußgröße, deren Wirkung auf die abhängige Variable untersucht wird

Literaturverzeichnis

[BEPW96] Backhaus, K. ; Erichson, B. ; Plinke, W. ; Weiber, R.: *Multivariate Analysemethoden: eine enwendungsorientierte Einführung.* 8. Auflage. Springer Verlag, Berlin, 1996. – ISBN 3–540–60917–2

[Boh03] Bohl, H.: *S37 Eigenwerte und Eigenvektoren.* http://www.mathematik.uni-stuttgart.de/HM/HMD/aufgaben01/node536.html, 07.10.2003

[Bor93] Bortz, J.: *Statistik für Sozialwissenschaftler.* 4. Auflage. Springer-Verlag, Berlin, 1993. – ISBN 3–540–56200–1

[Bök03a] Böker, F.: *Multivariate Verfahren – Kapitel 1 Überblick.* http://www.math.fu-berlin.de/~bioinf/download/mvsec1.pdf, 08.10.2003

[Bök03b] Böker, F.: *Multivariate Verfahren – Kapitel 2 Multivariate Verteilungen.* http://www.math.fu-berlin.de/~bioinf/download/mvsec2.pdf, 08.10.2003

[Bök03c] Böker, F.: *Multivariate Verfahren – Kapitel 3 Erste Schritte der Datenanalyse.* http://www.math.fu-berlin.de/~bioinf/download/mvsec3.pdf, 08.10.2003

[Bök03d] Böker, F.: *Multivariate Verfahren – Kapitel 4 Hauptkomponentenanalyse.* http://www.math.fu-berlin.de/~bioinf/download/mvsec4.pdf, 08.10.2003

[Bök03e] Böker, F.: *Multivariate Verfahren – Kapitel 5 Faktorenanalyse.* http://www.math.fu-berlin.de/~bioinf/download/mvsec5.pdf, 08.10.2003

[Bök03f] Böker, F.: *Multivariate Verfahren – Kapitel 6 Die multivariate Normalverteilung.* http://www.math.fu-berlin.de/~bioinf/download/mvsec6.pdf, 08.10.2003

[Bök03g] Böker, F.: *Multivariate Verfahren – Kapitel 7 Verfahren, die auf Normalverteilung basieren.* http://www.math.fu-berlin.de/~bioinf/download/mvsec7.pdf, 08.10.2003

[Bök03h] Böker, F.: *Multivariate Verfahren – Kapitel 8 Diskriminanzanalyse.* http://www.math.fu-berlin.de/~bioinf/download/mvsec8.pdf, 08.10.2003

[DT85] Deichsel, G. ; Trampisch, H.J.: *Clusteranalyse und Diskriminanzanalyse.* Gustav Fischer Verlag, Stuttgart, 1985. – ISBN 3–437–20342–8

[Ern01] Ernst, O.: *Numerische Mathematik - Kapitel 3 Direkte Verfahren zur Lösung linearer Gleichungssysteme.* http://igel.mathe.tu-freiberg.de/~ernst/Lehre/Techniker/folien3.pdf, 17.10.2001

[FF64] Faddejew, D.K. ; Faddejewa, W.N.: *Numerische Methoden der linearen Algebra.* R. Oldenbourg Verlag, München, 1964

[Han03] Handl, A.: *Multivariate Verfahren.* http://www.quantlet.com/mdstat/scripts/mst/pdf/mstpdf, 24.10.2003

[HE89] Hartung, J. ; Elpelt, B.: *Multivariate Statistik: Lehr- und Handbuch der angewandten Statistik.* 3. Auflage. R. Oldenbourg Verlag, München, 1989. – ISBN 3–486–21430–6

[Hen03] Henningsen, M.: *Kapitel 6 Tutorium am 1. Juni 2001.* http://www.math.tu-berlin.de/~hennings/tutorium_6.pdf, 07.10.2003

[Hoh03a] Hohenester, U.: *Kapitel 3 Matrixinvertierung.* http://physik.uni-graz.at/~uxh/lineare-algebra/Kapitel3.pdf, 08.10.2003

[Hoh03b] Hohenester, U.: *Kapitel 4 Gauß-Jordan-Verfahren.* http://physik.uni-graz.at/~uxh/lineare-algebra/Kapitel4.pdf, 08.10.2003

[Läu92] Läuter, J.: *Stabile multivariate Verfahren: Diskriminanzanalyse - Regressionsanalyse - Faktoranalyse.* Akademie Verlag GmbH, Berlin, 1992. – ISBN 3–05–501419–7

[Mar90] Marinell, G.: *Multivariate Verfahren: Einführung für Studierende und Praktiker.* 3. Auflage. R. Oldenbourg Verlag, München, 1990. – ISBN 3–486–21623–6

[Mey90] Meyer: *Meyers großes Taschenlexikon in 24 Bänden.* Bd. 5. 3. Auflage. Bibliographisches Institut & F.A. Brockhaus AG, Mannheim, 1990. – ISBN 3–411–11053–8

[Mey03] Meyer, H.: *Matrizen- und Vektorrechnung*. http://www.uni-bamberg.de/~ba2dp2/lehrmaterial/methodenlehre/Matrizenrechnung/, 10.10.2003

[Mör03a] Möring, M.: *Data Mining und statistische Datenanalyse*. http://www.uni-koblenz.de/~moeh/public/dmfol98.ps, 24.10.2003

[Mör03b] Möring, M.: *Grundlagen des Data Minig*. http://www.uni-koblenz.de/~moeh/public/dmfol.ps, 24.10.2003

[Mül69] Müller, D.: *Programmierung elektronischer Rechenanlagen*. Hochschultaschenbücher Bibliographisches Institut, Mannheim Wien Zürich, 1969. – ISBN 341100049X

[Pap88] Papula, L.: *Mathmatik für Ingenieure*. Bd. 2. 4. Auflage. Vieweg Verlag, 1988. – ISBN 3–528–34237–4

[RLL83] Röhr, M. ; Lohse, H. ; Ludwig, R.: *Statistik für Soziologen, Pädagogen, Psychologen und Mediziner*. Bd. 2. Verlag Harri Deutsch, Thun und Frankfurt am Main, 1983. – ISBN 3–87144–596–7

[VM02] Voss, H. ; Mackens, W.: *Grundlagen der Numerischen Mathematik*. http://www.tu-harburg.de/mat/LEHRE/material/grnummath_03.pdf, 2002

[Web03] Weber, O.: *Skript zur Vorlesung Multivariate Methoden der Sozialwissenschaften*. http://www.uns.umnw.ethz.ch/pers/weber/Skript.pdf, 08.10.2003

[Wir76] Wirtuk, E.: *Sozialpersonale Bedingungen der psychopysiologischen Aktivierung und der Lernleistungen bei programmiertem Material*, Karl-Marx-Universität Leipzig, Diss., 1976

Abbildungsverzeichnis

Tabellenverzeichnis

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben. Gleichzeitig versichere ich, diese Arbeit in gleicher oder ähnlicher Form weder veröffentlicht noch einer anderen Prüfungsbehörde vorgelegt zu haben.

Gießen, den 12. Dezember 2003