MATTHIAS KUHN

Generelle Kommentare zum Vortrag für Alle

Motto: Was man wie präsentiert, offenbart Kompetenz, Verständnistiefe und/oder Wissenslücken

*** Angaben zu METHODEN:**

1. Abkürzungen vermeiden (z.B. EDA) oder einmal ausschreiben & kurz erläutern (z.B. Modellnamen wie kNN)
2. klar sagen, wie die Daten bereinigt wurden (nur aufzählen, was gemacht wurde und nicht, was alles möglich ist)
   --> wenn das für die verschiedenen Modelle unterschiedlich war, dann jeweils neu angeben
3. genau angeben, welche Variablen für das jeweilige Modell verwendet wurden
   --> wenn das für die verschiedenen Modelle unterschiedlich war, dann jeweils neu angeben
4. Modelle eindeutig angeben (z.B. GLM mit family="binomial" für logistische Regression)
5. verwendete Statistik-Software samt Versionsnummer & Jahr angeben + wichtige R-Packages
   --> Zusatzfolie für die anschliessende Diskussion (verwendete R-Packages sollten im Code stehen)

*** Angaben zu OUTPUTS von STATISTIK & MACHINE LEARNING (ML):**

1. Resultate bevorzugt durch geeignete Grafiken präsentieren --> i.d.R. besser als Tabellen
   --> am informativsten sind Kombinationen aus Plots und zusätzlich eingetragenen Kennzahlen etc.
2. Interpretation in Worten für wichtige statistische Outputs & Machine Learning Outputs angeben
   --> z.B. was bedeutet kappa = 0.2 ?
3. (statistische) Fachbegriffe richtig und präzise verwenden
4. nur geeignete Kennzahlen / Parameter auswählen & präsentieren
5. wichtige Kennzahlen / Parameter zusammen mit ihren Konfidenzintervallen angeben

*** Erstellen einer PRÄSENTATION**:

1. Vortrag und einzelne Folien klar strukturieren (Zwischenüberschriften, numbers/bullets etc.) und dazu
   - inhaltlich / thematisch (!) informative Folien-Titel verwenden (z.B. nicht nur "Daten-Visulisierung: Boxplot")
   - klar angeben, zu welchem Datensatz oder Modell die Angaben, Diagramme etc. gehören
2. präzise formulieren und nicht zu viel auf eine Folie packen
   --> Ziel: genaue, korrekte Information & rasche Lesbarkeit (geeignete Schriftgrösse)
3. Platz auf den Folien gut nutzen (lesbare Schriftgrösse wählen & möglichst viel Platz für Grafiken verwenden)
4. Stichworte & Kurz-Sätze statt ausführlicher Sätze —> für Zuhörer/innen schnell(er) lesbar/erfassbar

Kommentar zu deiner Präsentation, Analyse und Vorhersage:

2 positive Feedbacks:

* 10 Folien vom 23.11. waren thematisch plus/minus gut gewählt

* Diagramme & Output nutzen den Platz auf den Folien gut

4 Verbesserungsvorschläge:

* Vortragszeit mit 7 min. klar zu kurz --> mit den 10 Folien vom 23.11. hättest du 10min. durchaus füllen können

* Folien mit Interpretationen der Diagramme & Outputs **für dein eigenes Modell**, keine allgemeinen Erläuterungen
--> Vortragsfolien unterscheiden sich klar von (i) einem Bericht (Inhalt & Aufbau) und von Vorlesungsfolien (Anzahl & Inhalt)

* kompletten R-Code nachliefern für ein logistisches Regressionsmodell (trainiert !)
---> prüfen, ob alles generiert wird, was die Folien zeigen, und zusätzlich auch die Datei mit der Vorhersage

* Vorhersage-Datei nachliefern für diese trainierte (!) logistische Regression --> es geht um Machine Learning

Vorschlag

Folie 1

Titel nicht klar verständlich

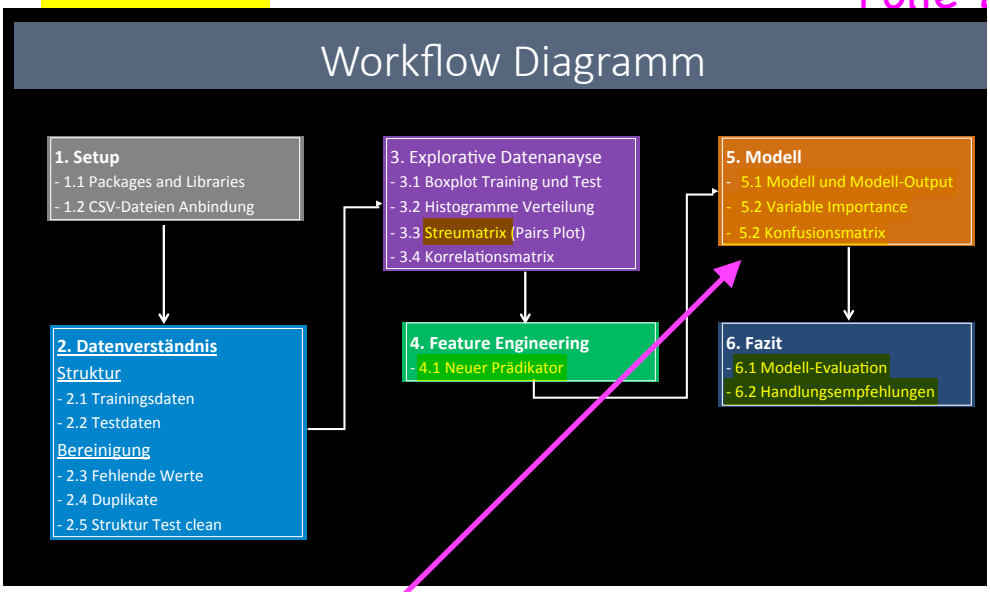
Autorenname ist wichtig & gehört zum Vortragstitel

Vorschlag

Folie 2

Lesbarkeit ist essentell !

-> weniger verschlungene Pfade



erhält man nur als Ergebnis eines Modells

Folie nicht geeignet für 10-minütigen Vortrag

--> Folien explizit für den Vortrag erstellen, kein Copy&Paste von einem (Shiny-)Bericht

... am 23.11. hast du einige Beispiele für gute Vortragsfolien gesehen !!!

Folie 3

1. Setup

1.1 Pakete und Bibliotheken

```
## 1.1 Setup Packages and Libraries ----

Folgende Pakete und Bibliotheken sind im Code enthalten:

- caret     - Für Modelltraining und -bewertung
- corrplot  - Für die Visualisierung der Korrelationsmatrix
- dplyr     - Für Datenmanipulation
- DT        - Für die Anzeige von Daten in Tabellen
- e1071     - Für die Berechnung von Streumaßen
- ggplot2   - Für die Visualisierung von Daten
- gridExtra - Für die Anordnung von Plots
- here      - Für die Verwaltung von Dateipfaden
- pROC      - Für die Erstellung von ROC-Kurven
- scales    - Für die Formatierung von Achsenbeschriftungen
- shiny     - Für die Erstellung von Shiny-Apps
- shinyjs   - Für JavaScript-Interaktionen in Shiny-Apps
- tidyr     - Für die Datenmanipulation
```

1.2 Anbindung der CSV-Dateien an den Shiny Server

```
Trainingsdaten geladen:  Ja
Testdaten geladen:  Ja
```

Bestätigung der geladenen Daten

1.2.1 Trainingsdaten Vorschau:

Output-Variable

	ID	MonateLetzteSpende	AnzahlSpenden	Gesamtvolumen	MonateErsteSpende	SpendeMaerz2007
1	619	2	50	12500	98	Ja
2	664	0	13	3250	28	Ja
3	441	1	16	4000	35	Ja
4	160	2	20	5000	45	Ja
5	358	1	24	6000	77	Nein

1.2.2 Testdaten Vorschau:

	ID	MonateLetzteSpende	AnzahlSpenden	Gesamtvolumen	MonateErsteSpende
1	659	2	12	3000	52
2	276	21	7	1750	38
3	263	4	1	250	4
4	303	11	11	2750	38
5	83	4	12	3000	34

Vorschlag

Folie 4

Zwischentitel aus Bericht nicht für Folien geeignet

Statistische Kennzahlen für Trainings- und Testdaten

2.1.1 Trainingsdaten Zusammenfassung

Output-Variable

```
       ID         MonateLetzteSpende AnzahlSpenden    Gesamtvolumen    MonateErsteSpende SpendeMaerz2007
 Min.   :  0.0   Min.   : 0.000      Min.   : 1.000   Min.   :  250    Min.   : 2.00     Nein:438
 1st Qu.:183.8   1st Qu.: 2.000      1st Qu.: 2.000   1st Qu.:  500    1st Qu.:16.00     Ja  :138
 Median :375.5   Median : 7.000      Median : 4.000   Median : 1000    Median :28.00
 Mean   :374.0   Mean   : 9.439      Mean   : 5.427   Mean   : 1357    Mean   :34.05
 3rd Qu.:562.5   3rd Qu.:14.000      3rd Qu.: 7.000   3rd Qu.: 1750    3rd Qu.:49.25
 Max.   :747.0   Max.   :74.000      Max.   :50.000   Max.   :12500    Max.   :98.00
```

2.2.1 Testdaten Zusammenfassung

```
       ID         MonateLetzteSpende AnzahlSpenden    Gesamtvolumen    MonateErsteSpende
 Min.   :  1.0   Min.   : 0.000      Min.   : 1.000   Min.   :  250    Min.   : 2.00
 1st Qu.:198.2   1st Qu.: 4.000      1st Qu.: 2.000   1st Qu.:  500    1st Qu.:14.00
 Median :377.5   Median : 7.000      Median : 4.000   Median : 1000    Median :31.00
 Mean   :374.6   Mean   : 9.495      Mean   : 5.935   Mean   : 1484    Mean   :35.48
 3rd Qu.:537.0   3rd Qu.:14.000      3rd Qu.: 8.000   3rd Qu.: 2000    3rd Qu.:52.00
 Max.   :745.0   Max.   :40.000      Max.   :41.000   Max.   :10250    Max.   :98.00
```

2.1-2.2 Stru

2.1.1 Trainingsdaten Zusammenfassung

```
       ID         MonateLetzteSpende AnzahlSpenden    Gesamtvolume
 Min.   :  0.0   Min.   : 0.000      Min.   : 1.000   Min.   : 25
 1st Qu.:183.8   1st Qu.: 2.000      1st Qu.: 2.000   1st Qu.:  50
 Median :375.5   Median : 7.000      Median : 4.000   Median : 100
 Mean   :374.0   Mean   : 9.439      Mean   : 5.427   Mean   : 135
 3rd Qu.:562.5   3rd Qu.:14.000      3rd Qu.: 7.000   3rd Qu.: 1750   3rd Qu.:49.25
 Max.   :747.0   Max.   :74.000      Max.   :50.000   Max.   :12500   Max.   :98.00
```

2.1.2 Struktur der Trainingsdaten

```
'data.frame':   576 obs. of  6 variables:
 $ ID               : int  619 664 441 160 358 335 47 164 736 436 ...
 $ MonateLetzteSpende: int  2 0 1 2 1 4 2 1 5 0 ...
 $ AnzahlSpenden     : int  50 13 16 20 24 4 7 12 46 3 ...
 $ Gesamtvolumen     : int  12500 3250 4000 5000 6000 1000 1750 3000 11500 750 ...
 $ MonateErsteSpende : int  98 28 35 45 77 4 14 35 98 4 ...
 $ SpendeMaerz2007   : Factor w/ 2 levels "Nein","Ja": 2 2 2 2 1 1 2 1 2 1 ...
```

```
3rd Qu.:537.0   3rd Qu.:14.000      3rd Qu.: 8.000   3rd Qu.: 2000    3rd Qu.:52.00
Max.   :745.0   Max.   :40.000      Max.   :41.000   Max.   :10250    Max.   :98.00
```

2.2.2 Struktur der Testdaten

```
'data.frame':   200 obs. of  5 variables:
 $ ID               : int  659 276 263 303 83 500 530 244 249 728 ...
 $ MonateLetzteSpende: int  2 21 4 11 4 3 4 14 20 14 ...
 $ AnzahlSpenden     : int  12 7 1 11 12 21 2 1 2 4 ...
 $ Gesamtvolumen     : int  3000 1750 250 2750 3000 5250 500 250 500 1000 ...
 $ MonateErsteSpende : int  52 38 4 38 34 42 4 14 87 64 ...
```

ACHTUNG: Codierung der Variable "SpenderMaerz2007" ja = 1 und nein = 2 (nicht 0, 1)

Folie nicht geeignet für 10 minütigen Vortrag
--> Folien für Vortrag erstellen, kein Copy&Paste vom Shiny-Bericht

Folie 5

2.3 – 2.5 Datenbereinigung

1.3 Fehlende Werte

Diese Tabelle zeigt die Anzahl der fehlenden Werte in den Trainings- und Testdaten für jede Variable. Fehlende Werte können die Modellleistung beeinträchtigen und müssen daher identifiziert und behandelt werden.

Variable	Training	Test
ID	0	0
MonateLetzteSpende	0	0
AnzahlSpenden	0	0
Gesamtvolumen	0	0
MonateErsteSpende	0	0
SpendeMaerz2007	0	x

1.4 Tabelle Duplikate Testdaten

Diese Tabelle zeigt die Anzahl der doppelten IDs in den Testdaten. Doppelte Einträge können die Analyse verzerren und sollten daher identifiziert und bereinigt werden.

Frequenz	AnzahlZeilen
3	2
2	24

1.5 Zusammenfassung der bereinigten Testdaten

Diese statistische Zusammenfassung der bereinigten Testdaten gibt einen Überblick über die Verteilung und zentrale Tendenzen der Variablen nach dem Entfernen von Duplikaten und fehlenden Werten.

```
Statistische Zusammenfassung der bereinigten Testdaten:
       ID        MonateLetzteSpende AnzahlSpenden    Gesamtvolumen   MonateErsteSpende
 Min.   :  1.0   Min.   : 0.000     Min.   : 1.000   Min.   :  250   Min.   : 2.00
 1st Qu.:195.0   1st Qu.: 4.000     1st Qu.: 2.000   1st Qu.:  500   1st Qu.:14.00
 Median :355.0   Median : 7.500     Median : 3.500   Median :  875   Median :29.00
 Mean   :371.7   Mean   : 9.733     Mean   : 5.808   Mean   : 1452   Mean   :35.
 3rd Qu.:539.0   3rd Qu.:14.000     3rd Qu.: 7.250   3rd Qu.: 1812   3rd Qu.:
 Max.   :745.0   Max.   :40.000     Max.   :41.000   Max.   :10250   Max.   :98.00
```

1.5A Struktur der bereinigten Testdaten

Die Struktur der bereinigten Testdaten zeigt die Datentypen und die ersten paar Einträge jeder Variable nach der Bereinigung. Dies stellt sicher, dass die Daten bereit für die weitere Analyse sind.

```
Struktur der bereinigten Testdaten
'data.frame':   172 obs. of  5 variables:
 $ ID                : int  659 276 263 303 83 500 530 244 249 728 ...
 $ MonateLetzteSpende: int  2 21 4 11 4 3 4 14 23 14 ...
 $ AnzahlSpenden     : int  12 7 1 11 12 21 2 1 2 4 ...
 $ Gesamtvolumen     : int  3000 1750 250 2750 3000 5250 500 250 500 1000 ...
 $ MonateErsteSpende : int  52 38 4 38 34 42 4 14 87 64 ...
```

Doppelte können hier nicht aus den Testdaten entfernt werden, da sie auch in der Submission-datei doppelt sind

Folie 6

Vorschlag

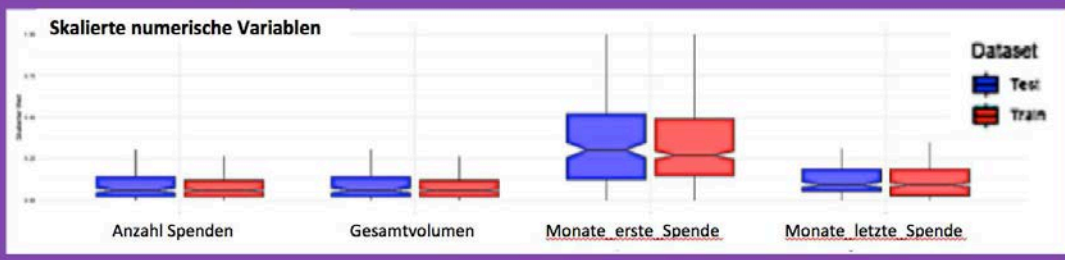
Lesbarkeit ist essentiell

d.h. weniger ist mehr

--> * grössere Schrift

* grössere Legende

Vergleich der Trainings- und Testdaten



Erkenntnisse

- Verteilungen der Variablen sind nicht unterschiedlich für Trainings- und Testdaten
  --> Kerben der Boxen überlappen
- Variabilität ist im Testdatensatz teilweise geringer, z.B. bei "Anzahl Spenden"

Schlussfolgerung

Man kann annehmen, dass Test- und Trainingsdaten Stichproben aus derselben Population sind.

2.1

2.1 Boxplot Trainings- und Testdaten

Der Boxplot zeigt die Verteilung der ausgewählten numerischen Variablen in

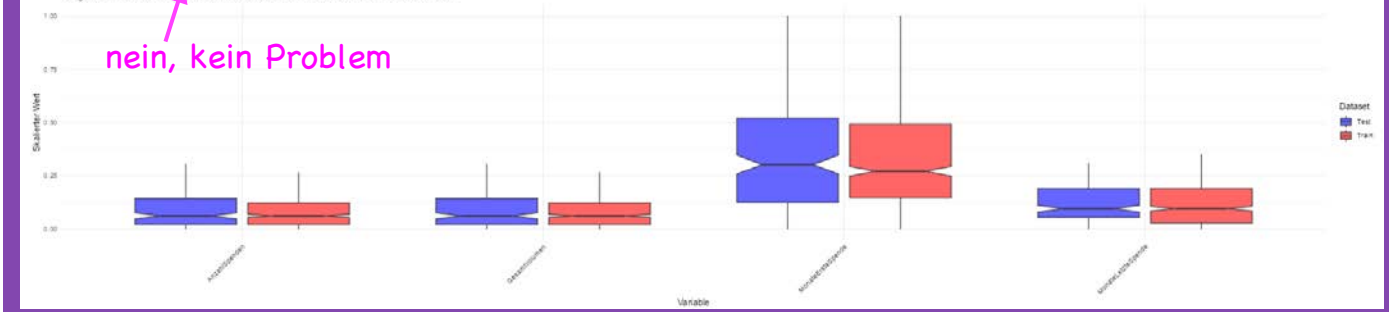
- Ausreißer zu identifizieren, die die Modellleistung beeinflussen kön
- Unterschiede in der mittleren Tendenz (Median) zwischen den Dat
- Potenzielle Variabilitätsunterschiede (Boxlängen) zu analysieren.

Erkenntnisse:

- Die Verteilungen der Variablen sind größtenteils konsistent zwische
- Die Variabilität (Boxlänge) einiger Variablen, wie 'AnzahlSpenden',
- Ausreißer bei 'Gesamtvolumen' könnten auf Besonderheiten in der

Schlussfolgerung: Es könnten Generalisierungsprobleme auftreten, wenn die

Vergleich der skalierten numerischen Variablen zwischen Train- und Te



nein, kein Problem

2.2 Vergleich der Verteilungen Test und Train

2.2 Vergleich der Verteilungen zwischen Train und Test

steht im Folien-Titel

Diese Visualisierung zeigt die Verteilungen der numerischen Variablen in den Trainings- und Testdaten. Ziel ist es, Unterschiede oder Ähnlichkeiten zu erkennen, um potenzielle Herausforderungen für die Modellgeneralisation zu identifizieren.

auf Folie unnötig

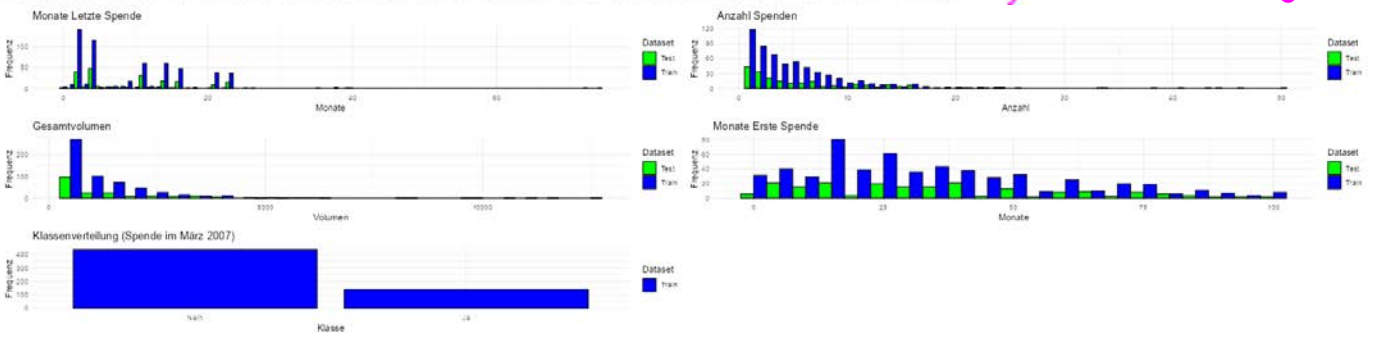
- MonateLetzteSpende: Untersucht die zeitliche Verteilung, wann Spenden zuletzt erfolgten.
- AnzahlSpenden: Zeigt die Häufigkeit der Blutspenden pro Spender.
- Gesamtvolumen: Analysiert die Verteilung des gesamten gespendeten Blutvolumens.
- MonateErsteSpende: Gibt Aufschluss über den Zeitraum der ersten Spende.

Erkenntnisse aus den Plots:

- **"MonateLetzteSpende"**: Die Verteilungen stimmen weitgehend überein, aber leichte Verschiebungen im Testdatensatz könnten auf Unterschiede in der Stichprobenstruktur hinweisen.
- **"AnzahlSpenden"**: Im Testdatensatz gibt es weniger hohe Spendenanzahlen, was auf eine mögliche Stichprobenverzerrung hinweisen könnte.
- **"Gesamtvolumen"**: Ähnliche Muster wie bei der AnzahlSpenden, da das Gesamtvolumen stark von der AnzahlSpenden abhängt.
- **"MonateErsteSpende"**: Die Verteilung im oberen Bereich ist im Testdatensatz geringer, was auf unterschiedliche Verhaltensmuster der Spender hinweisen könnte.

grössere Schrift & stichwortartig

Schlussfolgerung: Unterschiede in den Verteilungen könnten die Modellleistung beeinflussen. Eine Anpassung der Gewichtung oder weitere Feature-Engineering-Maßnahmen könnten notwendig sein.



Inhalte von Folien 8 & 9 & 12 auf einer Folie zusammenfassen

UNPASSEND:
Text bezieht sich nicht auf die abgebildete Streumatrix (pairs), somdern auf Boxplots

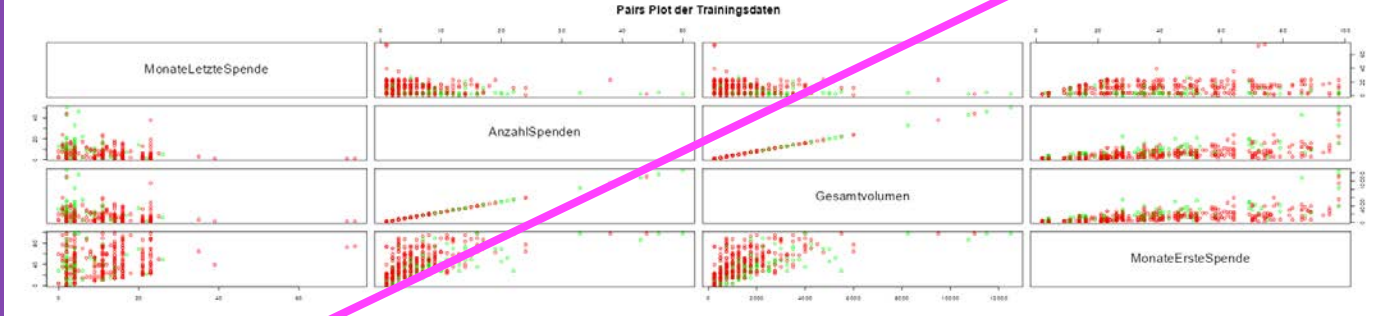
2.3 Pairs Plot

2.3 Pairs Plot

Erkenntnisse:

- MonateLetzteSpende: Der Boxplot zeigt, dass die Verteilung der Werte in den Trainings- und Testdaten ähnlich ist. Beide Datensätze haben eine ähnliche mittlere Tendenz und Streuung. Dies deutet darauf hin, dass diese Variable zwischen den Datensätzen konsistent bleibt, was positiv für die Modellgeneralisation ist.
- AnzahlSpenden und Gesamtvolumen: Beide Variablen zeigen in Testdaten tendenziell weniger Variabilität (engere Boxen) im Vergleich zu den Trainingsdaten. Dies könnte auf ein Sampling-Bias oder Unterschiede in den beiden Datensätzen hinweisen.
- MonateErsteSpende: Auffällige Unterschiede im oberen Quartil der Werte könnten eine systematische Abweichung zwischen den Datensätzen signalisieren, was die Modellleistung beeinträchtigen könnte.

Schlussfolgerung: Unterschiede in den Verteilungen könnten dazu führen, dass ein Modell, das auf den Trainingsdaten trainiert wurde, Schwierigkeiten hat, auf die Testdaten zu generalisieren. Eine weitere Analyse (z. B. Korrekturen oder Feature-Engineering) könnte erforderlich sein.



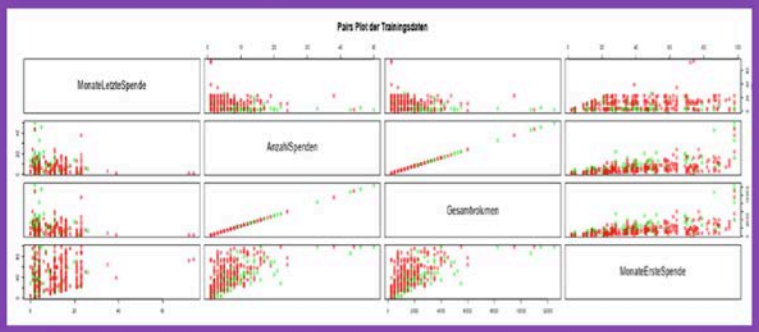
Inhalte von Folien 8 & 9 & 12 auf einer Folie zusammenfassen

Folie 9

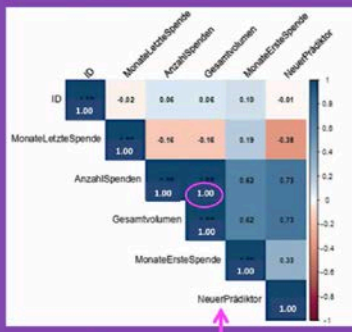
Vorschlag

Trainingsdaten - Beziehung der Variablen

Streumatrix



Korrelationen



Erkenntnisse

- 'Anzahl-Spenden' und 'Gesamtvolumen' haben Korrelation 1 → alle Punkte auf einer Linie in der Streumatrix
- Korrelation von 'NeuerPrädiktor' mit 'Anzahl-Spenden' und mit 'Gesamtvolumen' ist mit 0.73 ebenfalls h hoch

Schlussfolgerung

- 'Anzahl-Spenden' beibehalten und 'Gesamtvolumen' weglassen

$$NeuerPrädiktor = \frac{AnzahlSpenden}{MonateLetzteSpende}$$

2

2.4 Korrelationsmatrix

Die Korrelationsmatrix visualisiert die Stärke und Richtung der lineare
Farbe entspricht der Stärke der Korrelation.

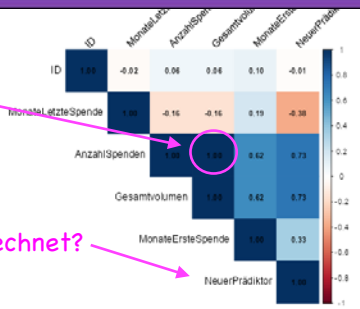
Hohe Korrelationen (absoluter Wert > 0.8) können auf redundante Info
verbessern.

- Hohe Korrelation: "Gesamtvolumen" und "AnzahlSpenden"
- Niedrige Korrelation: "MonateErsteSpende" hat nur eine ger
- Multikollinearität: Variablen mit st r Korrelation sollten un

Empfohlene Maßnahmen:

- Untersuchung hoch korrelierter Variablen: Variablen wie "G
- Berücksichtigung unabhängiger Variablen: "MonateErsteSp
- Verwendung von PCA oder Regularisierung: Falls mehrere V

Korrelation ist 1 nicht 0.62



bei Korrelationen fehlt die Angabe, welche signifikant sind -> nur diese sind relevant

wie ist der NeuerPrädiktor berechnet?
--> wird erst auf Folie 12 erklärt

Diagramm uninteressant - und es ergibt auch keinen Sinn

Folie 10

2.5 Datenstruktur Outputgrössen

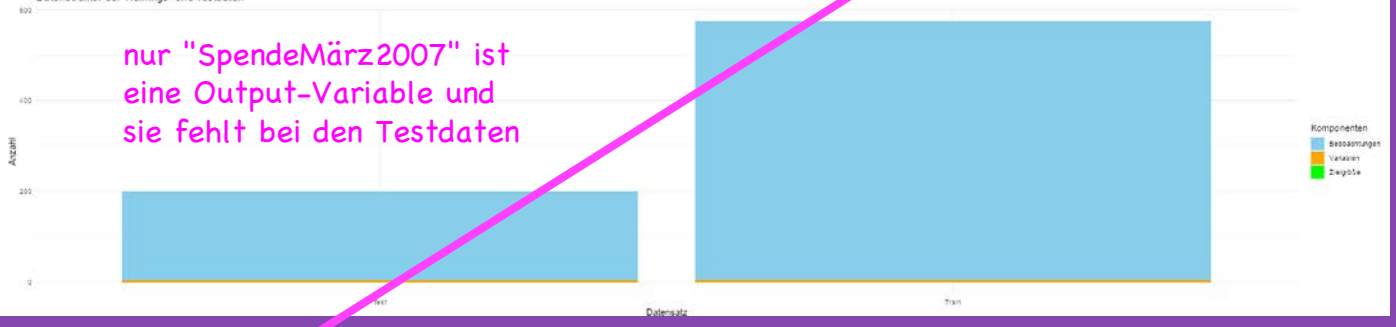
2.5 Datenstruktur Outputgrößen

Dieser Abschnitt visualisiert die Struktur der Trainings- und Testdaten. Er zeigt die Anzahl der Beobachtungen (n), der Variablen (p), sowie das Vorhandensein der Zielgröße (Y) für den Trainingsdatensatz.

Diese Visualisierung ist hilfreich, um sicherzustellen, dass die Datenkonsistenz zwischen Train- und Testdatensatz gewährleistet ist. Unterschiede in der Anzahl der Variablen oder fehlende Zielgrößen in den Testdaten können Einfluss auf die Modellbewertung haben.

- **Beobachtungen (n):** Der Trainingsdatensatz enthält deutlich mehr Beobachtungen als der Testdatensatz. Dies ist in der Regel positiv, da ein größeres Trainingsset eine robustere Modellanpassung ermöglicht.
- **Variablen (p):** Die Anzahl der Prädiktorvariablen (ohne Zielgröße) ist zwischen Trainings- und Testdatensatz konsistent. Dies zeigt, dass die Datensätze korrekt vorbereitet wurden.
- **Zielgröße (Y):** Die Zielvariable ist erwartungsgemäß nur im Trainingsdatensatz vorhanden, da der Testdatensatz ausschließlich für Vorhersagen verwendet wird.

Datenstruktur der Trainings- und Testdaten



nur "SpendeMärz2007" ist eine Output-Variable und sie fehlt bei den Testdaten

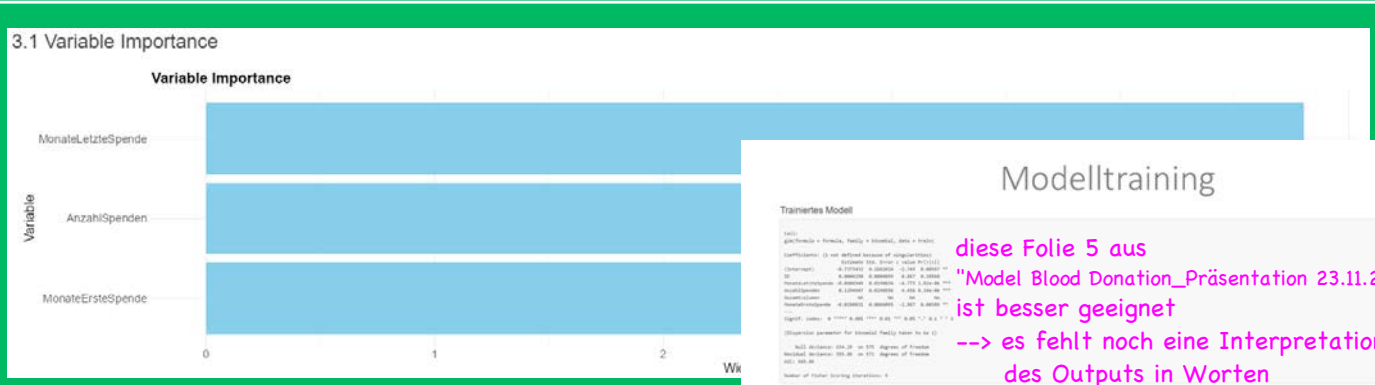
Inhalte von Folien 11 & 13 auf einer Folie zusammenfassen

Variablen-Wichtigkeit erhält man erst als Resultat eines Modells

--> wie wichtig Variablen/Features sind, hängt vom gewählten Modell ab !

Folie 11

3.1 Variable Importance



diese Folie 5 aus "Model Blood Donation_Präsentation 23.11.2024.pdf" ist besser geeignet

--> es fehlt noch eine Interpretation des Outputs in Worten

für welches Modell ?

--> hier das Modell vorstellen, gemäss Vortragstitel & Folie 13 müsste es eine logistische Regression sein

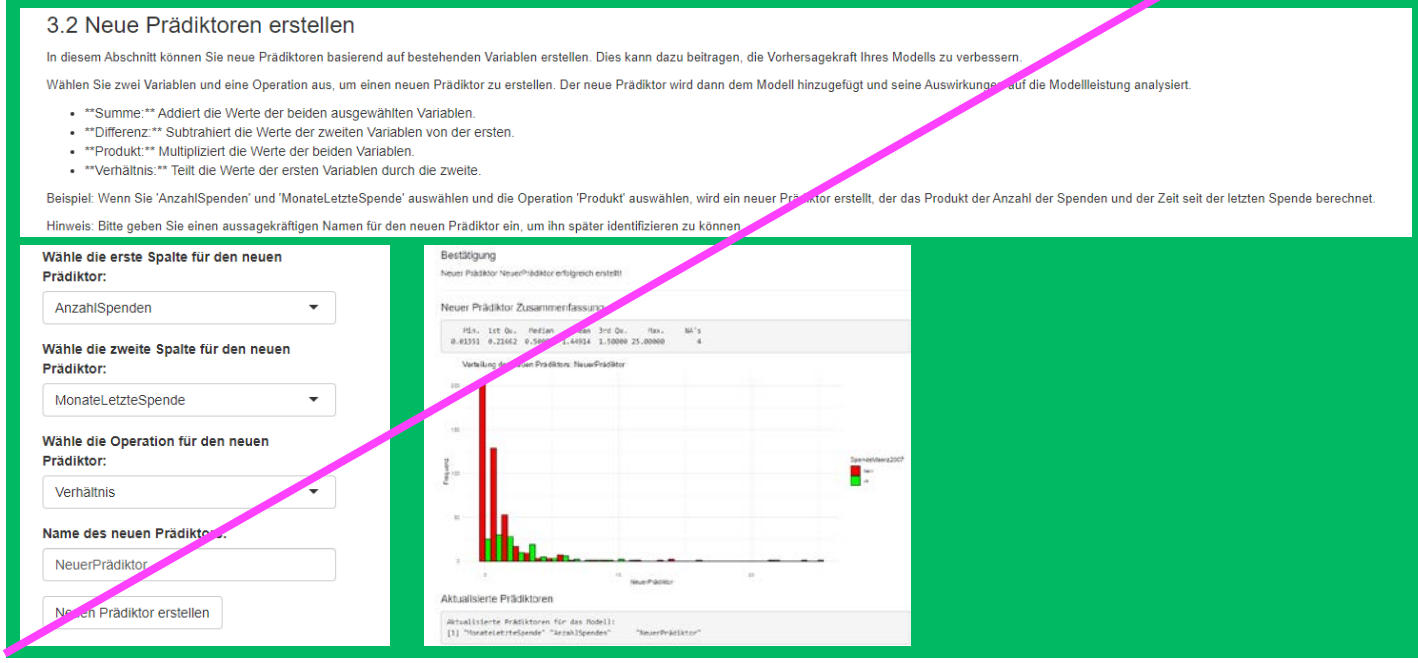
stammt von Folie 13

```
Call:
glm(formula = formula, family = binomial(), data = train_data())

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -0.74578    0.19423  -3.840 0.000123 ***
MonateLetzteSpende -0.10738    0.01818  -5.905 3.52e-09 ***
AnzahlSpenden      0.07180    0.01861   3.859 0.000114 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 634.29  on 575  degrees of freedom
Residual deviance: 565.56  on 573  degrees of freedom
AIC: 571.56

Number of Fisher Scoring iterations: 5
```

Folie 12

Berechnungsformel von 'NeuerPrädiktor' auf "neuer" Folie 9 (Vorschlag)

3.2 Neuer Prädikator



Inhalte von Folien 11 & 13 auf einer Folie zusammenfassen

Folie 13

Resultat der logistischen Regression

4.1 Modellzusammenfassung

4.1 Modellzusammenfassung

In diesem Abschnitt wird die Modellzusammenfassung des trainierten Modells bereitgestellt. Die Zusammenfassung bietet einen Einblick in:

- **Die ausgewählten Prädiktoren:** Welche Variablen wurden für das Modell verwendet und wie wirken sie sich auf die Zielvariable aus?
- **Die geschätzten Koeffizienten:** Diese Koeffizienten zeigen, wie stark und in welche Richtung (positiv oder negativ) jede Variable die Zielvariable beeinflusst.
- **Die statistische Signifikanz der Koeffizienten:** Zeigt, ob ein Prädiktor einen signifikanten Einfluss auf die Zielvariable hat. Dies wird anhand des p-Werts überprüft:
  - **p < 0.05:** Der Prädiktor hat einen statistisch signifikanten Einfluss.
  - **p ≥ 0.05:** Der Einfluss des Prädiktors ist nicht signifikant.
- **Die Modellgüte:** Beurteilung der Anpassungsgüte durch Metriken wie:
  - **AIC (Akaike-Informationskriterium):** Niedrigere Werte deuten auf ein besser angepasstes Modell hin.
  - **Null- und Residual-Deviance:** Zeigen, wie gut das Modell die Daten erklärt.

allgemeine Erläuterungen
-> nicht geeignet für Folien über die Ergebniss deiner Modellierung

Die Modellzusammenfassung ist wichtig, um:

- **Identifikation wichtiger Prädiktoren:** Sie hilft zu erkennen, welche Variablen entscheidend sind, um die Zielvariable vorherzusagen. Nicht signifikante Variablen könnten entfernt werden, um das Modell zu vereinfachen.
- **Interpretation der Beziehungen:** Die Koeffizienten geben Aufschluss darüber, wie sich eine Änderung in den Prädiktoren auf die Zielvariable auswirkt.
- **Bewertung der Modellgüte:** Die zusammenfassenden Statistiken helfen zu beurteilen, ob das Modell gut genug ist, um verlässliche Vorhersagen zu treffen.

Das Modell wird auf Basis einer logistischen Regression (glm) trainiert. Die Zielvariable 'SpendeMaerz2007' ist binär (Ja/Nein). Die Formel wird dynamisch erstellt, basierend auf den vom Benutzer ausgewählten Prädiktoren.

```
Call:
glm(formula = formula, family = binomial(), data = train_data())

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -0.74578    0.19423  -3.840 0.000123 ***
MonateLetzteSpende -0.10738    0.01818  -5.905 3.52e-09 ***
AnzahlSpenden       0.07180    0.01861   3.859 0.000114 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 634.29  on 575  degrees of freedom
Residual deviance: 565.56  on 573  degrees of freedom
AIC: 571.56

Number of Fisher Scoring iterations: 5
```

hier wird nichts trainiert !
das ist eine normale logistische Regression mit den zwei Variablen "MonateLetzteSpende" und "Anzahlspenden"-> "NeuerPrädiktor" nicht verwendet

Logistische Regression mit caret

hier wird eine logistische Regression trainiert

- Wir verwenden wieder cats aus MASS:

```
cats <- MASS::cats
set.seed(3454)
i_train <- createDataPartition(cats$Sex, p = 0.8, list = FALSE) train
<- cats[i_train, ]
test <- cats[-i_train, ]
model <- train(Sex ~ ., data = train, method = "glm", family = "binomial")
pred <- predict(model, newdata = test)
postResample(pred, test$Sex)

##  Accuracy     Kappa
## 0.8928571 0.7307692
```

- Mit kNN hatten wir eine Accuracy von 0.75 erreicht.

Inhalte von Folien 14 & 15 auf einer Folie zusammenfassen

Folie 14

4.2 Modelleistung

4.2 Modellleistung

Dieser Abschnitt präsentiert die Leistungskennzahlen des trainierten Modells. Die Modellleistung wird durch Metriken gemessen, die zeigen, wie gut das Modell die Zielvariable vorhersagen kann.

Die folgenden Metriken werden dargestellt:

- **Genauigkeit (Accuracy):** Der Anteil der korrekt vorhergesagten Beobachtungen an der Gesamtanzahl der Beobachtungen. Eine hohe Genauigkeit deutet darauf hin, dass das Modell insgesamt zuverlässig vorhersagt.
- **Präzision (Precision):** Der Anteil der richtig positiven Vorhersagen (True Positives) an allen positiven Vorhersagen (True Positives + False Positives). Präzision ist besonders wichtig, wenn die Kosten für falsche positive Vorhersagen hoch sind.
- **Recall (Sensitivität):** Der Anteil der richtig positiven Vorhersagen (True Positives) an allen tatsächlichen positiven Beobachtungen (True Positives + False Negatives). Recall ist entscheidend, wenn es wichtig ist, alle positiven Fälle zu identifizieren.

Zusätzlich wird eine Konfusionsmatrix (siehe Abschnitt 4.2A) bereitgestellt, die die tatsächlichen Klassen (Ist-Werte) und die vorhergesagten Klassen (Soll-Werte) gegenüberstellt.

Diese Kennzahlen bieten eine umfassende Analyse der Stärken und Schwächen des Modells und helfen bei der Identifikation potenzieller Verbesserungsbereiche.

Eine hohe Genauigkeit allein garantiert nicht, dass das Modell für alle Anwendungen geeignet ist. Die Präzision und der Recall sind ebenfalls wichtig, insbesondere wenn das Modell auf einen spezifischen Anwendungsfall zugeschnitten ist.

- **Beispiel 1 (hohe Präzision, niedriger Recall):** Das Modell macht kaum Fehler bei den positiven Vorhersagen, identifiziert jedoch nicht alle positiven Fälle.
- **Beispiel 2 (niedrige Präzision, hoher Recall):** Das Modell erkennt fast alle positiven Fälle, macht aber viele Fehler bei den positiven Vorhersagen.

Diese Metriken helfen dabei, das Modell an die spezifischen Anforderungen anzupassen. Zum Beispiel könnte ein Modell mit hohem Recall bevorzugt werden, wenn es darum geht, alle potenziellen Fälle zu identifizieren (z. B. in medizinischen Anwendungen).

Metrik	Wert
Genauigkeit	0.23
Präzision	0.61
Recall	0.10

das Ergebnisse deines Modells, oder ?

woher kommen diese Werte?
jedenfalls nicht von Folie 15

allgemeine Erläuterungen
-> nicht geeignet für Folien über die Ergebniss deiner Modellierung

Inhalte von Folien 14 & 15 auf einer Folie zusammenfassen

Folie 15

4.2A Konfusionsmatrix

4.2A Konfusionsmatrix

Die Konfusionsmatrix gibt eine detaillierte Übersicht über die Vorhersageergebnisse des Modells. Sie vergleicht die tatsächlichen Klassen (Ist-Werte) mit den vorhergesagten Klassen (Soll-Werte).

Die Konfusionsmatrix ist in vier Bereiche unterteilt:

- **True Positives (TP):** Fälle, bei denen das Modell korrekt vorhergesagt hat, dass die Zielvariable zutrifft (z. B. Spende = Ja).
- **True Negatives (TN):** Fälle, bei denen das Modell korrekt vorhergesagt hat, dass die Zielvariable nicht zutrifft (z. B. Spende = Nein).
- **False Positives (FP):** Fälle, bei denen das Modell fälschlicherweise vorhergesagt hat, dass die Zielvariable zutrifft (z. B. Spende = Ja, aber tatsächlich Spende = Nein).
- **False Negatives (FN):** Fälle, bei denen das Modell fälschlicherweise vorhergesagt hat, dass die Zielvariable nicht zutrifft (z. B. Spende = Nein, aber tatsächlich Spende = Ja).

Die Konfusionsmatrix bietet die Grundlage für die Berechnung vieler wichtiger Modellkennzahlen wie Genauigkeit, Präzision und Recall.

Ein Beispiel für eine Konfusionsmatrix:

	Vorhergesagt Ja	Vorhergesagt Nein
Tatsächlich Ja	TP	FN
Tatsächlich Nein	FP	TN

```
Confusion Matrix and Statistics

          Reference
Prediction Nein  Ja
      Nein  429 124
      Ja      9  14

               Accuracy : 0.7691
                 95% CI : (0.7325, 0.8029)
    No Information Rate : 0.7604
    P-Value [Acc > NIR] : 0.3327

                  Kappa : 0.1132

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9795
            Specificity : 0.1014
         Pos Pred Value : 0.7758
         Neg Pred Value : 0.6087
             Prevalence : 0.7604
         Detection Rate : 0.7448
   Detection Prevalence : 0.9601
      Balanced Accuracy : 0.5405

       'Positive' Class : Nein
```

interpretieren

hier sollten die Erkenntnisse aus deinem Modell stehen, nicht allgemeine Erkentnisse

Mögliche Erkenntnisse aus der Konfusionsmatrix:

- Hohe Werte für TP und TN deuten darauf hin, dass das Modell zuverlässig ist.
- Hohe Werte für FP und FN könnten auf Probleme im Modell hinweisen, z. B. eine falsche Gewichtung der Klassen.
- Die Analyse von FP und FN hilft, die Schwächen des Modells besser zu verstehen und mögliche Verbesserungen [illegible]

Die Konfusionsmatrix kann auch verwendet [illegible] ein Klassenungleichgewicht hinzuweisen. Falls eine Klasse deutlich häufiger vorkommt als die andere, könnte dies das Modell verzerren und weitere Maßnahmen wie eine Gewichtung der Klassen oder Sampling erfordern.

Die Konfusionsmatrix wird berechnet, indem die tatsächlichen Klassen aus den Trainingsdaten mit den vom Modell vorhergesagten Klassen verglichen werden. Dies ermöglicht eine präzise Beurteilung der Modellleistung auf den Trainingsdaten.

Lesbarkeit ist essentiell --> zwei Folien daraus machen

Folie 16

5. Fazit

5.1 Modellzusammenfassung

Dieser Abschnitt fasst die wichtigsten Erkenntnisse aus der Modellierung zusammen. Die Analyse berücksichtigt die Genauigkeit, Präzision und Recall sowie Stärken und Schwächen des Modells.

- Das Modell zeigt eine gute Gesamtgenauigkeit, was auf eine zuverlässige Klassifizierung hindeutet.
- Stärken: Hohe Präzision in der Vorhersage der Zielklasse (Spende Ja/Nein).
- Schwächen: Unterschiede in der Variabilität der Testdaten könnten die Generalisierungsfähigkeit des Modells beeinträchtigen.

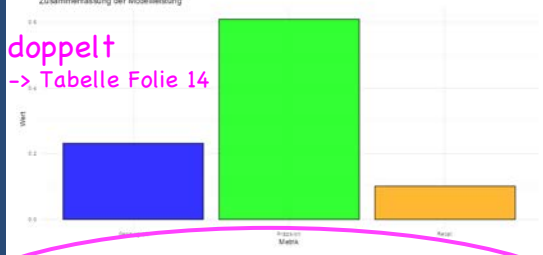
woran hast du das abgelesen?

```
Erkenntnisse aus der Modellzusammenfassung:
--------------------------------------------
1. Die wichtigsten Prädiktoren laut Modell:
                Estimate  Std. Error  z value     Pr(>|z|)
(Intercept)   -1.4978019 0.19423485 -3.839580 1.232451e-04
SpendeJa      -0.10718284 0.01818434 -5.905238 3.511387e-09
AnzahlSpenden  0.07179922 0.01860621  3.858884 1.139061e-04

2. Modellgüte (AIC): 571.5639
--------------------------------------------
```

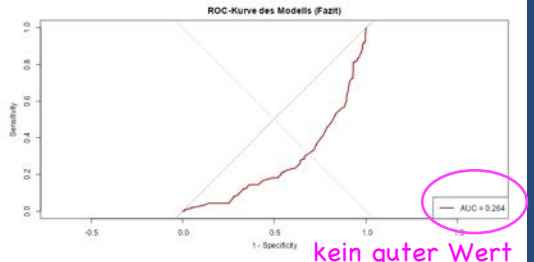
doppelt
--> schon auf Folie 13

5.2 Visualisierung der Modellleistung

Die folgenden Plots visualisieren die Leistung des Modells anhand der wichtigsten Metriken: Genauigkeit, Präzision und Recall sowie die ROC-Kurve.



doppelt
-> Tabelle Folie 14



kein guter Wert

5.3 Handlungsempfehlungen

Basierend auf den Ergebnissen des Modells und der Datenanalyse werden die folgenden Empfehlungen ausgesprochen:

- Weitere Feature-Engineering-Maßnahmen könnten die Modellleistung steigern.
- Untersuche die Verteilung der Testdaten und gleiche mögliche Unterschiede durch Datentransformationen aus.
- Falls mehr Daten verfügbar sind, könnte eine erneute Modellierung mit größerem Trainingsdatensatz die Robustheit verbessern.
- Überlege, ob das Modell in einem iterativen Prozess mit Feedback-Schleifen weiterentwickelt werden sollte.

das ist wichtig --> neue Folie, die den Vortrag prägnant abschliesst