DAV

DEUTSCHE
AKTUARVEREINIGUNG e.V.

Schriftliche Prüfung im Spezialwissen

Actuarial Data Science Advanced

gemäß Prüfungsordnung 5
der Deutschen Aktuarvereinigung e. V.

am 18.10.2024

Hinweise:

- Als Hilfsmittel ist ein Taschenrechner zugelassen.
- Die Gesamtpunktzahl beträgt 180 Punkte. Die Klausur ist bestanden, wenn mindestens 90 Punkte erreicht werden.
- Bitte prüfen Sie die Ihnen vorliegende Prüfungsklausur auf Vollständigkeit. Die Klausur besteht aus 34 Seiten und einer Seite als Anlage, die mit abzugeben ist.
- Alle Antworten sind zu begründen und bei Rechenaufgaben muss der Lösungsweg ersichtlich sein.

Mitglieder der Prüfungskommission:

Axel Kiermaier, Dr. René Külheim, Dr. Jonas Offtermatt,
Dr. Felix Spangenberg

Aufgabe 1 [1.2 Datenschutz 2, 3.3 Visualisierung 1, 3.4 Innovative Produkte 1] (28 Punkte)

In ihrem Unternehmen steht eine Neuerung in der Produktentwicklung an. Erstmalig sollen Data Science Methoden in der Produktentwicklung eingesetzt werden.

a) [6 Punkte] Sie sollen Ihren Kollegen in einer Präsentation die Vorteile von Data Science in der Produktentwicklung nahebringen. Nennen Sie hierfür 4 Vorteile, welche sich durch den Einsatz von Data Science Methoden in der Produktentwicklung ergeben.

b) [5 Punkte] Im Anschluss an Ihre Präsentation spricht Sie der Datenschutzbeauftragte Ihres Unternehmens an. Er hat Bedenken wegen Artikel 25 der DSGVO. Erläutern Sie ihm, wie die Grundsätze des Datenschutzes durch *Privacy by Design* und *Privacy by Default* dabei helfen können, innovative Produkte zu entwickeln, die den Datenschutz respektieren und das Vertrauen der Kunden erhöhen. Erläutern Sie insbesondere, was unter *Privacy by Design* und *Privacy by Default* zu verstehen ist.

c) [8 Punkte] Nach wenigen Wochen Projektlaufzeit haben Ihre Kollegen, das erste Modell gefittet. Sie präsentieren Ihnen folgende Ergebnisse:

Datenpunkt	1	2	3	4	5	6	7	8
y_test	84	80	95	85	85	94	100	92
y_predicted	75	71	90	70	74	88	86	84

Zeichnen Sie auf Basis der Tabelle einen Scatter-Plot y_test gegen y_predicted und einen Residuen-Plot. Bitte achten Sie hierbei auch auf die grundlegenden Anforderungen an eine Visualisierung (bspw. Verwendung eines Lineals, Achsenbeschriftung, Titel, …).

d) [4 Punkte] Interpretieren Sie die Ergebnisse der Visualisierungen.

e) [5 Punkte] Mit welchen Python-Befehlen könnten Sie den ScatterPlot erstellen? (Vergessen Sie nicht die nötigen Imports.)

Lösungsvorschlag:

a) Mögliche Vorteile sind:

Verbesserte Entscheidungsfindung: Durch den Einsatz von Data Science können große Mengen von Daten analysiert werden, um fundierte Entscheidungen während des gesamten Produktentwicklungsprozesses zu treffen. Dies ermöglicht es, Trends zu identifizieren, Kundenbedürfnisse besser zu verstehen und die Produktleistung zu optimieren.

Effizienzsteigerung: Data Science ermöglicht die Automatisierung von Prozessen in der Produktentwicklung, wie z.B. die Vorhersage von Produktionszeiten, die Optimierung von Lieferketten oder die frühzeitige Erkennung von potenziellen Problemen. Dadurch können Zeit und Ressourcen eingespart und die Produktentwicklung beschleunigt werden.

Personalisierte Produktgestaltung: Durch die Analyse von Kundendaten und -verhalten können Produkte individuell an die Bedürfnisse und Vorlieben der Kunden angepasst werden. Data Science ermöglicht die Entwicklung maßgeschneiderter Produkte, die einen höheren Mehrwert bieten und die Kundenzufriedenheit steigern.

Wettbewerbsvorteil: Unternehmen, die Data Science in ihre Produktentwicklung integrieren, können sich einen Wettbewerbsvorteil verschaffen, indem sie innovative Produkte schneller auf den Markt bringen und gleichzeitig die Qualität und Effektivität ihrer Produkte verbessern. Durch die Nutzung von Daten können sie auch Trends frühzeitig erkennen und sich besser an Marktveränderungen anpassen.

Weitere Nennungen von Vorteilen sind möglich. Pro Vorteil 1,5 Punkte.

b) Privacy by Design und Privacy by Default sind Prinzipien, die sicherstellen, dass Datenschutz von Anfang an in die Entwicklung von Produkten integriert wird.

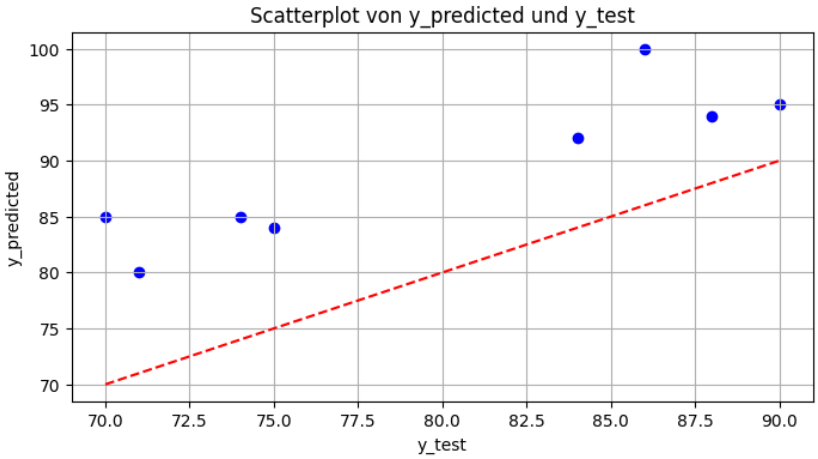
Unter **Privacy by Design** versteht man die Integration von Datenschutzmerkmalen bereits in die Architektur und das Design von Produkten und Dienstleistungen. Das bedeutet, dass Datenschutz von Anfang an berücksichtigt wird, indem beispielsweise datenschutzfreundliche Standardeinstellungen implementiert werden, Datenminimierung und Anonymisierung angewendet werden und die Sicherheit der Datenübertragung gewährleistet wird.

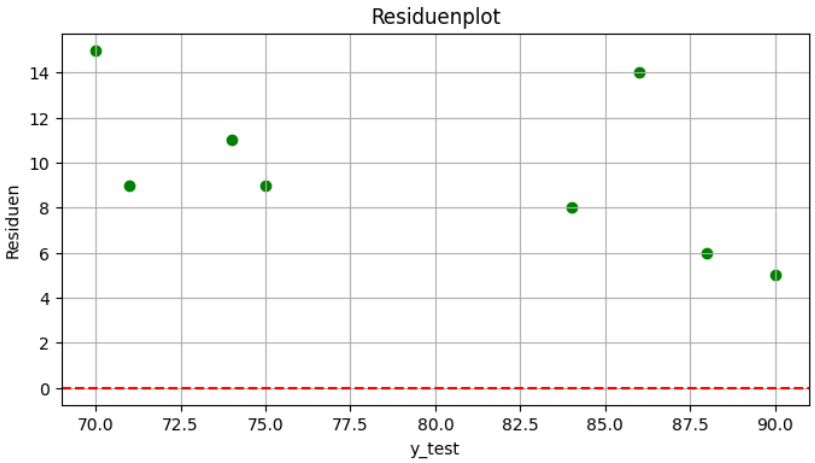
Privacy by Default bedeutet, dass Datenschutz als Standard eingestellt ist, ohne dass der Benutzer manuelle Einstellungen vornehmen muss. Dies bedeutet, dass die strengsten Datenschutzoptionen automatisch aktiviert sind und der Benutzer aktiv andere Einstellungen vornehmen muss, wenn er den Datenschutz ändern möchte.

Beide Grundsätze tragen dazu bei, das Vertrauen der Verbraucher in die Sicherheit ihrer persönlichen Daten zu stärken und das Risiko von Datenschutzverletzungen zu minimieren. Indem Datenschutz von Anfang an in den Entwicklungsprozess integriert wird und als Standard eingestellt ist, können Unternehmen innovative Produkte entwickeln, die den Datenschutz respektieren und gleichzeitig den gesetzlichen Anforderungen, insbesondere den Bestimmungen der DSGVO, entsprechen.

2 Punkte pro Erläuterung eines Prinzips und 1 Punkt für die allgemeine Erklärung, warum innovative Produkte damit möglich sind.

c) Die beiden Schaubilder sollten ungefähr so aussehen:





Wie immer gibt es Abzug für fehlende Beschriftungen, unsaubere Zeichnungen, etc.

d) Da beim Scatterplot zwei getrennte Cluster von Punkten weit weg von der Normalgeraden liegen, deutet dies darauf hin, dass es eine starke nicht-lineare Beziehung zwischen den Variablen gibt. Das bedeutet, dass die Datenpunkte nicht entlang einer geraden Linie streuen, sondern eher eine gekrümmte oder schräge Form aufweisen.

   Die Tatsache, dass die Residuen einen deutlichen Abstand von der Nulllinie aufweisen, deutet darauf hin, dass das Modell nicht alle Variationen der abhängigen Variable erklären kann. Die Residuen sind die Differenz zwischen den beobachteten Werten und den vorhergesagten Werten des Modells. Wenn die Residuen einen großen Abstand von der Nulllinie haben, bedeutet dies, dass das Modell große Fehler bei der Vorhersage der abhängigen Variable macht. Ein großer Abstand von der Nulllinie deutet darauf hin, dass das Modell systematische Fehler aufweist und eine unzureichende Anpassung an die Daten vorliegt.

   Insgesamt deuten der Scatterplot und der Residuenplot darauf hin, dass das aktuelle Modell möglicherweise nicht angemessen ist, um die Beziehung zwischen den Variablen zu beschreiben. Möglicherweise müssen komplexere Modelle oder nicht-lineare Transformationen der Daten in Betracht gezogen werden, um die Beziehung besser zu erfassen und die Vorhersagegenauigkeit zu verbessern.

e) Der Python-Code sollte in etwa so aussehen:

```
import matplotlib.pyplot as plt
# Datenpunkte aus der Tabelle
y_predicted = [84, 80, 95, 85, 85, 94, 100, 92]
y_test = [75, 71, 90, 70, 74, 88, 86, 84]

# Scatterplot
plt.figure(figsize=(8, 4))
plt.scatter(y_test, y_predicted, color='blue')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)],
color='red', linestyle='--')
plt.xlabel('y_test')
plt.ylabel('y_predicted')
plt.title('Scatterplot von y_predicted und y_test')
plt.grid(True)
plt.show()
```

Aufgabe 2 [3.1 Data Mining 2, 4.4 Datenaufbereitung zur Modellerstellung] (16 Punkte)

a) [2 Punkte] Nennen Sie zwei alternative Data-Mining-Prozessmodelle zu CRISP-DM.

b) [9 Punkte] Beschreiben Sie die drei Arten von fehlenden Werten in Datensätzen nach der Typologie von Rubin mit ihren jeweiligen Abkürzungen, Langnamen und geben Sie für jede Art ein Beispiel aus dem Versicherungskontext an.

c) [5 Punkte] Führen Sie auf folgendem Datensatz ein One-Hot-Encoding / eine Dummy-Kodierung auf der Spalte Claim_Location durch.

Year	Contract_ID	Claim_Location	Claim_Amount
2022	1	City	1500
2022	2	Rural	800
2022	3	Suburb	2000
2022	4	City	1200
2022	5	Suburb	1800

Lösungsvorschlag:

a) Zwei mögliche Alternative Data-Mining-Prozesse zu CRISP-DM sind:

SEMMA: SEMMA steht für Sample, Explore, Modify, Model, Assess und wurde von SAS entwickelt. Dieser Prozess beginnt mit der Auswahl einer Stichprobe der Daten, gefolgt von der Exploration der Daten, der Modifikation von Variablen und der Modellbildung. Anschließend werden die Modelle bewertet und validiert.

KDD-Process (Knowledge Discovery in Databases): Der KDD-Prozess ist ein breiterer Rahmen, der über das reine Data Mining hinausgeht und auch andere Aspekte des Wissensentdeckungsprozesses umfasst, wie z.B. die Vorverarbeitung von Daten, die Interpretation von Ergebnissen und die Integration des Wissens in den Geschäftsprozess. Es umfasst sechs Phasen: Auswahl, Preprocessing, Transformation, Data Mining, Interpretation/Evaluation und Integration/Nutzung.

Weitere Nennungen sind möglich. Es reicht die reine Nennung der Prozessnamen.

b) Nach der Typologie von Rubin gibt es drei Arten von fehlenden Werten in Datensätzen:

MCAR (Missing Completely at Random):

Beschreibung: Bei dieser Art von fehlenden Werten fehlt der Beobachtungswert völlig zufällig, unabhängig von anderen Variablen oder Beobachtungen.

Beispiel aus dem Versicherungskontext: Ein Fragebogen zur Kundenzufriedenheit wird per Post verschickt, aber einige Kunden erhalten den Fragebogen nie, weil die Post zufällig verloren geht.

MAR (Missing at Random):

Beschreibung: Bei dieser Art von fehlenden Werten fehlt der Beobachtungswert zufällig, jedoch abhängig von anderen Variablen im Datensatz.

Beispiel aus dem Versicherungskontext: Kunden sollen bei einer Umfrage ihren Wohnort und ihre Einkommensdaten angeben, aber einige Kunden (aus „besseren" Wohngegenden) geben die Einkommensdaten nicht an.

MNAR (Missing Not at Random):

Beschreibung: Bei dieser Art von fehlenden Werten fehlt der Beobachtungswert nicht zufällig und ist abhängig von nicht beobachteten Variablen oder den fehlenden Werten selbst.

Beispiel aus dem Versicherungskontext: Kunden geben ihre Gesundheitsinformationen an, aber Kunden mit schwerwiegenden Vorerkrankungen könnten dazu neigen, diese Information bewusst zu verschweigen.

3 Punkte pro Typ, jeweils 1 Punkt für Abkürzung und Langname, einen für die Beschreibung und einen fürs Beispiel.

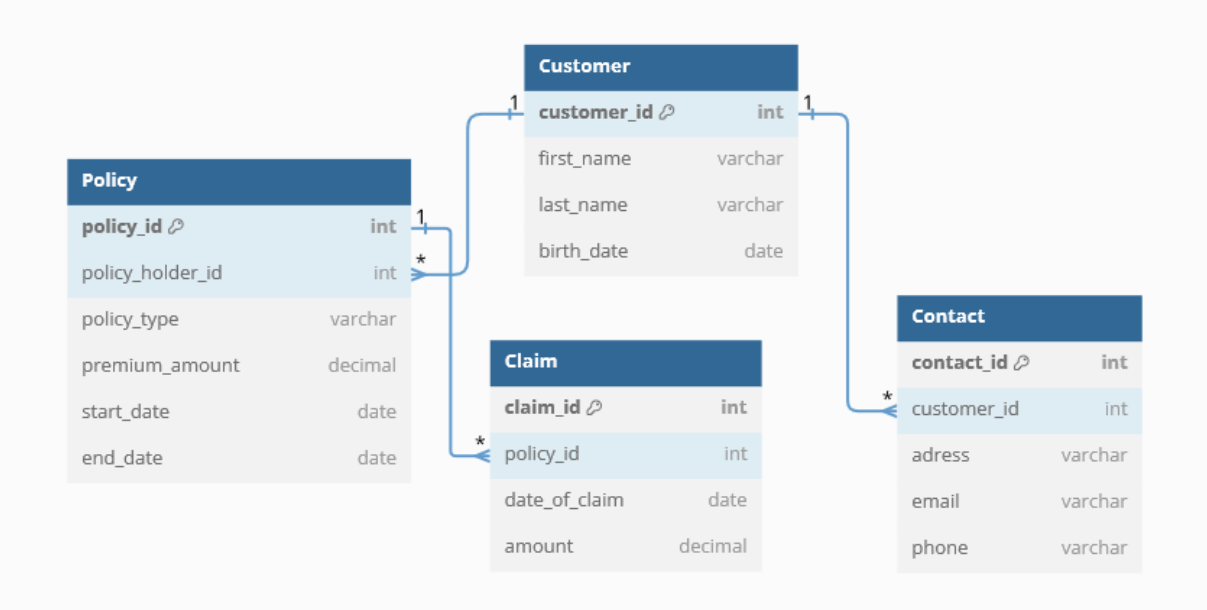
c) Datensatz nach der Anwendung von One-Hot-Encoding:

Year	Contract_ID	Claim_Location_City	Claim_Location_Rural	Claim_Location_Suburb	Claim_Amount
2022	1	1	0	0	1500
2022	2	0	1	0	800
2022	3	0	0	1	2000
2022	4	1	0	0	1200
2022	5	0	0	1	1800

Pro richtige Zeile ein Punkt.

Aufgabe 3 [2.1 Datenmanagement 2] (21 Punkte)

Gegeben ist das folgende Entity-Relation-Diagramm:



Und folgende Beispieldatensätze für die Tabellen `Policy` und `Customer`:

poli-cy_id	poli-cy_holde r_id	policy_type	premi-um_amount	start_date	end_date
1	1	Auto	500	2023-01-01	2024-01-01
2	2	Haus	1000	2023-02-01	2024-02-01
3	3	Leben	1500	2023-03-01	2024-03-01

customer_id	first_name	last_name	birth_date	contact_id
1	Max	Mustermann	1990-01-01	1
2	Anna	Schmidt	1985-05-15	2
3	Peter	Müller	1978-11-30	3

a) [5 Punkte] Identifizieren und beschreiben Sie die verschiedenen Entitäten und Beziehungen des Diagramms. Erläutern Sie, wie diese Elemente miteinander verbunden sind und welche Informationen sie repräsentieren.

b) [4 Punkte] Erläutern Sie die Begriffe *Primärschlüssel* und *Fremdschlüssel*, jeweils anhand eines Beispiels aus dem obigen Diagramm.

c) [3 Punkte] Schreiben Sie eine SQL-Abfrage, um alle Kunden mit ihren Namen und Geburtsdaten aus der Tabelle „`Customer`" abzurufen.

d) [3 Punkte] Schreiben Sie eine SQL-Abfrage, um einen neuen Schadensfall mit den Werten `policy_id = 1`, `date_of_claim = „2024-01-01"` und `amount = „12345.00"` in die Tabelle „Claim" einzufügen.

e) [6 Punkte] Geben Sie jeweils das Ergebnis der folgenden drei SQL-Befehle an:

i)

```
SELECT Policy.policy_id, Policy.policy_type, Customer.first_name,
Customer.last_name

FROM Policy
INNER JOIN Customer ON Policy.policy_holder_id = Customer.customer_id
WHERE Policy.start_date >= '2023-02-01' AND Policy.end_date <= '2024-
02-01';
```

ii)

```
SELECT Policy.policy_id, Policy.policy_type, Customer.first_name,
Customer.last_name
FROM Policy
LEFT JOIN Customer ON Policy.policy_holder_id = Customer.customer_id
WHERE Customer.customer_id IS NULL;
```

iii)

```
SELECT Policy.policy_id, Customer.first_name, Customer.last_name,
Policy.premium_amount
FROM Policy
LEFT JOIN Customer ON Policy.policy_holder_id = Customer.customer_id
WHERE Policy.premium_amount >= 1000
```

Lösungsvorschlag:

a) **Entitäten und Attribute**:

Versicherungspolice *(Policy)*: Enthält Informationen wie Policentyp, Prämienbetrag und Gültigkeitszeitraum.

Kunde *(Customer)*: Speichert Kundeninformationen wie Name und Geburtsdatum.

Kontaktdaten *(Contact)*: Speichert Adress- und Kontaktdaten.

Schadensfall *(Claim)*: Hält Daten zu Schadensfällen wie Datum und Schadensbetrag.

Beziehungen:

Zwischen "Versicherungspolice" und "Kunde": Ein Kunde kann mehrere Versicherungspolicen haben, aber eine Police gehört nur zu einem Kunden.

Zwischen "Versicherungspolice" und "Schadensfall": Eine Versicherungspolice kann mehrere Schadensfälle haben, aber ein Schadensfall ist nur mit einer Police verbunden.

Zwischen „Kunde" und „Kontaktdaten": Ein Kunde kann mehrere Kontaktdaten haben, aber ein gegebenes Set an Kontaktdaten gehört zu genau einem kunden.

Diese Entitäten und Beziehungen ermöglichen es der Versicherungsgesellschaft, Kunden, Versicherungspolicen und Schadensfälle effizient zu verwalten und miteinander zu verknüpfen. (1 Punkt je Entität und Beziehung)

b) **Primärschlüssel**: Ein Primärschlüssel identifiziert eindeutig jede Zeile in einer Tabelle. In unserem Diagramm ist zum Beispiel die Kundennummer ("customer_id") in der Tabelle "Kunde" ein Primärschlüssel.

Fremdschlüssel: Ein Fremdschlüssel ist eine Spalte in einer Tabelle, die auf den Primärschlüssel einer anderen Tabelle verweist. Zum Beispiel verweist die Spalte "policy_holder_id" in der Tabelle "Versicherungspolice" auf die Kundennummer in der Tabelle "Kunde". Sie zeigt an, welchem Kunden die Versicherungspolice gehört.

c) Der SQL-Befehl sollte lauten:

```
SELECT first_name, last_name, birth_date
```

```
FROM Customer;
```

d) Der SQL-Befehl sollte lauten:

```
INSERT INTO Claim (policy_id, date_of_claim, amount)

VALUES (1, '2024-01-01', 12345.00);
```

e) Ergebnis des SQL-Befehls mit INNER JOIN und zusätzlichen Bedingungen:

```
| policy_id | policy_type | first_name | last_name |
|-----------|-------------|------------|-----------|
| 2         | Haus        | Anna       | Schmidt   |
```

Ergebnis des SQL-Befehls mit LEFT JOIN und Bedingungen für fehlende Kundendaten:

```
Keine Ergebnisse gefunden.
```

Ergebnis des SQL-Befehls mit LEFT JOIN und Bedingungen für Prämie über 1000:

```
| policy_id | first_name | last_name | premium_amount |
|-----------|------------|-----------|----------------|
| 2         | Anna       | Schmidt   | 1000           |
| 3         | Peter      | Müller    | 1500           |
```

Aufgabe 4 [4.3 Korrelation & Kausale Inferenz] (30 Punkte)

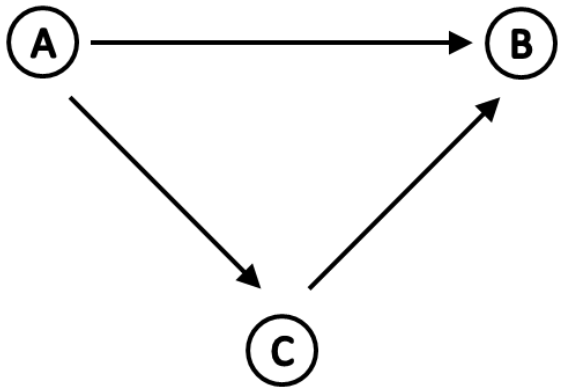
a) [6 Punkte] Geben Sie eine mathematische Definition für DAGs und erläutern Sie den Zusammenhang zur mathematischen Behandlung von Kausalität. Welche Rolle spielen das "D" und das "A" in „DAG" in Bezug auf die Eigenschaften einer Kausalbeziehung?

b) [9 Punkte] Zeichnen und benennen Sie die drei grundlegenden Strukturen („Grundformen"), die in DAGs vorkommen können. Nennen Sie jeweils ein einfaches Beispiel aus dem Versicherungsumfeld.

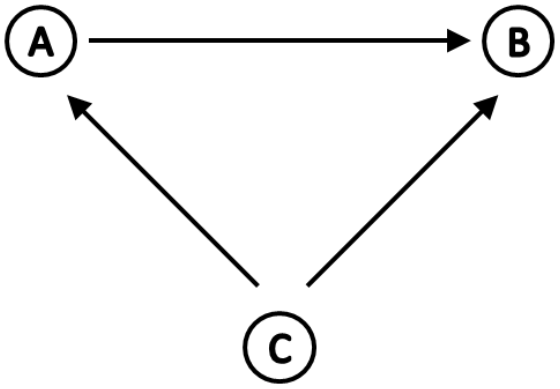
c) [7 Punkte] Wie ist der Zusammenhang zwischen DAGs und dem Konzept der stochastischen Unabhängigkeit und wie drückt sich das in mathematischen Formeln aus? Inwiefern zeigt dies, dass der herkömmliche statistische Kalkül nicht geeignet ist, um Kausalität auszudrücken, und wie lässt sich dieses Problem gemäß Pearl lösen?

d) [8 Punkte] Bei den folgenden vier Graphen steht die Kausalbeziehung zwischen *A* und *B* im Mittelpunkt. Welche Rolle spielt in diesem Zusammenhang jeweils *C*? Bestimmen Sie jeweils die gemeinsame Wahrscheinlichkeitsverteilung *P(A,B,C)*.

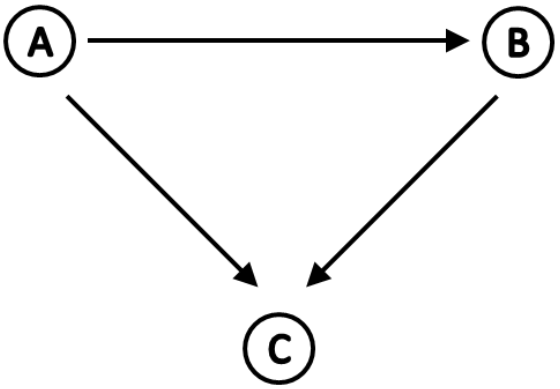
(d1):



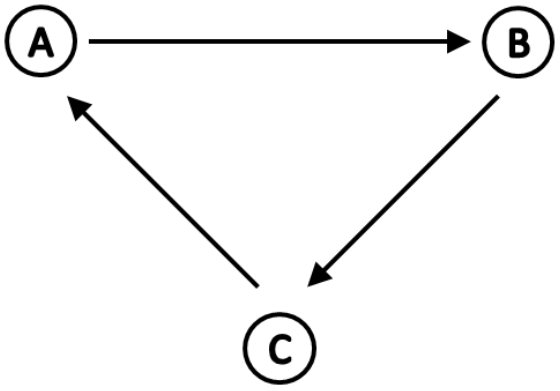
(d2):



(d3):



(d4):



Lösungsvorschlag:

a) Ein DAG („Directed Acyclic Graph" = „gerichteter azyklischer Graph") ist ein Begriff aus der Graphentheorie. Wie jeder Graph besteht ein DAG aus Knoten und Kanten.

Ein Graph ist ein Paar $G = (V, E)$, wobei V die Menge der Knoten und E die Menge der Kanten ist. Dabei ist jede Kante e in E einem ungeordneten Paar (u,v) von Knoten in V zugeordnet („e verbindet u und v").

Bei einem DAG haben die Kanten eine Richtung („directed") und es gibt keine Zyklen („acyclic"), also keinen Pfad (keine Folge von Kanten) entlang der Richtung der Kanten, der zum Ausgangsknoten zurückführt.

In der mathematischen Theorie der Kausalität betrachtet man die Knoten als Zufallsvariable und die Kanten als Ursache-Wirkungs-Beziehung; eine Kante zwischen zwei Zufallsvariablen bedeutet das Vorliegen einer direkten Ursache.

In dieser Interpretation sind das „D" („directed") und das „A" wesentlich, um die Eigenschaften der Kausalbeziehung abzubilden:

- Die Kausalbeziehung ist asymmetrisch, d.h. wenn A Ursache von B ist, kann B nicht Ursache von A sein. Daher muss der Graph gerichtet sein („D").
- Die Kausalbeziehung ist transitiv, d.h. wenn A Ursache von B ist und B Ursache von C, so ist A Ursache von C. Daher muss – in Verbindung mit der Asymmetrie – der Graph azyklisch sein („A").

b) Die drei grundlegenden Strukturen („Grundformen"), die in DAGs vorkommen können, sind Chain, Fork und Collider. Sie zeigen jeweils Beziehungen zwischen drei Zufallsvariablen X, Y und Z.

1. Chain:

Diese lineare Struktur zeigt eine aufeinanderfolgende Abhängigkeit zwischen den drei Variablen.

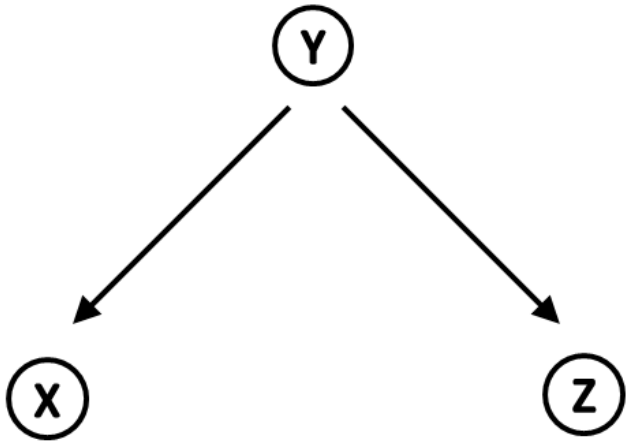
Chain

Beispiel aus dem Versicherungsumfeld:

Feuer → Rauch → Rauchmelder schlägt an

2. Fork:

Diese Gabelungsstruktur zeigt eine gemeinsame Ursache *Y* für die beiden Variablen *X* und *Z*.



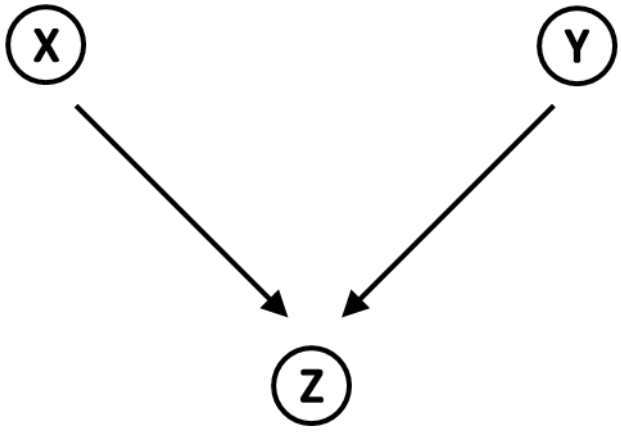
Fork

Beispiel aus dem Versicherungsumfeld:

Sterbewahrscheinlichkeit ← Alter → Fahrverhalten

3. Collider:

Diese Zusammenflussstruktur zeigt eine Situation, in der zwei Variablen *X* und *Y* als Ursache für eine Variable *Z* fungieren, was zu einer scheinbaren Korrelation zwischen ihnen führen kann.



Collider

Beispiel aus dem Versicherungsumfeld:

Jährliche Fahrkilometer → Unfall ← PS

c) Für die drei grundlegenden Strukturen aus b) gilt folgendes:

Chain: Gegeben Y sind X und Z bedingt unabhängig ($X \perp\!\!\!\perp Z|Y$), d.h.

$$P(X,Z|Y) = P(X|Y) \cdot P(Z|Y).$$

Fork: Gegeben Y sind X und Z bedingt unabhängig ($X \perp\!\!\!\perp Z|Y$), d.h.

$$P(X,Z|Y) = P(X|Y) \cdot P(Z|Y).$$

Collider: Gegeben Z sind X und Y bedingt abhängig ($X \not\perp\!\!\!\perp Y|Z$), d.h.

$$P(X,Y|Z) \neq P(X|Z) \cdot P(Y|Z).$$

Die Chain $X \rightarrow Y \rightarrow Z$ und die Fork $X \leftarrow Y \rightarrow Z$ haben entsprechend der genannten bedingten Unabhängigkeit die gleiche Wahrscheinlichkeitsverteilung, aber verschiedene kausale Strukturen. Damit ist die Sprache der traditionellen Wahrscheinlichkeitsverteilung nicht ausreichend, um diese kausalen Unterschiede abzubilden.

Dies gelingt aber mit dem von Pearl eingeführten do-Operator, der Interventionen abbildet. $P(Y|X=x_1)$, die traditionelle bedingte Wahrscheinlichkeit, gibt nur die beobachtete Wahrscheinlichkeit für Y wieder, wenn X zufällig x_1 ist. Dagegen ist $P(Y|do(X=x_1))$ die Wahrscheinlichkeit für Y, die bei der Intervention gilt, bei der X „auf x_1 gesetzt" wird. Es gilt i.a.

$P(Y|X=x_1) \neq P(Y|do(X=x_1))$.

Dies ist der Unterschied zwischen Sehen / Beobachten und Tun.

d) In **(d1)** erfolgt die Wirkung von *A* auf *B* nicht (nur) direkt, sondern teilweise über *C*. *(Anm.: C wird hier auch als Mediator bezeichnet.)* Zur Bestimmung des direkten Effekts von *A* auf *B* könnte man durch geeignete Verfahren den totalen Effekt von *A* auf *B* (z.B. durch Regression) sowie den Mediationseffekt von *C* (z.B. durch eine Pfadanalyse) getrennt bestimmen und die Differenz hieraus bilden.

$$P(A,B,C) = P(B|A,C)\,P(A,C) = P(B|A,C)\,P(C|A)\,P(A)$$

In **(d2)** ist *C* ein Confounder (Störgröße) für die Wirkung von *A* auf *B*. Er beeinflusst sowohl *A* wie auch *B*. Ein Confounder kann zu falschen Schlussfolgerungen führen, wenn er nicht angemessen kontrolliert oder berücksichtigt wird.

$$P(A,B,C) = P(B|A,C)\,P(A,C) = P(B|A,C)\,P(A|C)\,P(C)$$

In **(d3)** ist *C* ein Collider. Gegeben *C* werden *A* und *B* (auch wenn sie unabhängig sind) bedingt abhängig. Hierdurch können kontraintuitive Korrelationen entstehen.

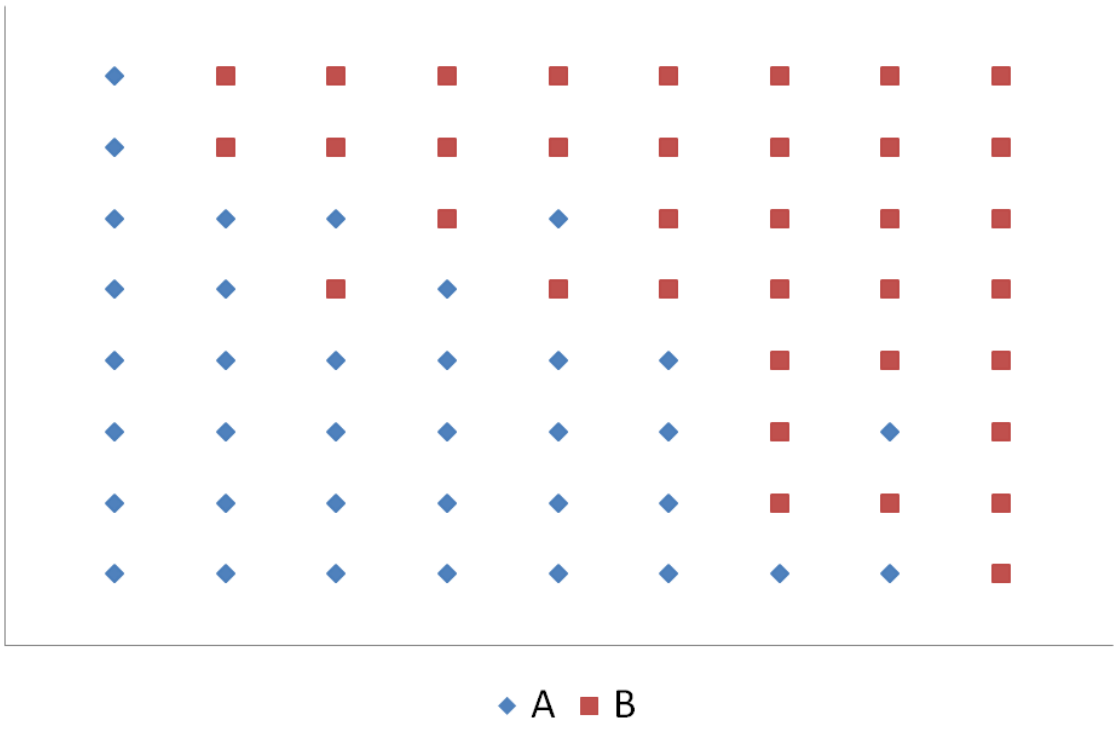
$$P(A,B,C) = P(C|A,B)\,P(A,B) = P(C|A,B)\,P(B|A)\,P(A)$$

In **(d4)** liegt kein DAG vor. Der Graph ist zyklisch und somit nicht für die Abbildung eines Kausalzusammenhangs geeignet.

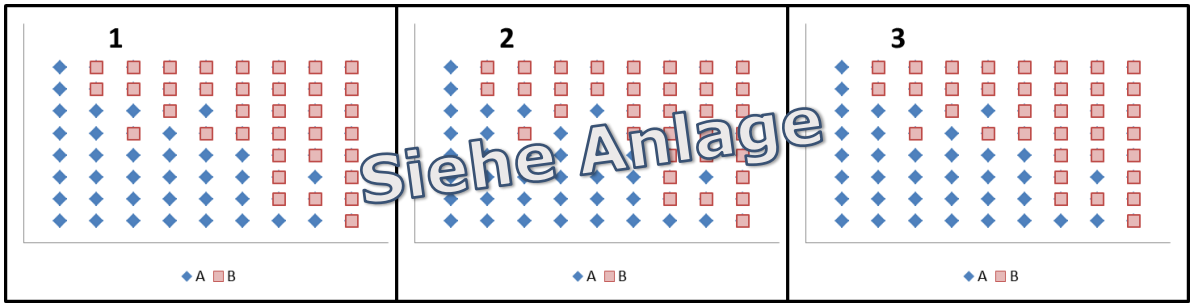
Insbesondere lässt sich $P(A,B,C)$ nicht in ähnlicher Weise faktorisieren.

Aufgabe 5 [3.2 Analytics 2] (15 Punkte)

a) [6 Punkte] Was versteht man unter Over- und Underfitting? Erläutern Sie dies allgemein verbal sowie durch Einzeichnung von glatten Kurven Ihrer Wahl in der folgenden Grafik. Diese zeigt Ausprägungen A und B eines binären Merkmals in Abhängigkeit von zwei unabhängigen Merkmalen.



Hinweis: Benutzen Sie zur Beantwortung von Aufgabe a) bitte die der Klausur beigefügten Grafiken.



Achten Sie darauf, die von Ihnen bearbeitete Seite zusätzlich abzugeben!

b) [4 Punkte] Wie lässt sich Overfitting erkennen? Nennen und erläutern Sie zwei verschiedene Möglichkeiten.

c) [5 Punkte] Wie kann es zu Overfitting kommen? Leiten Sie hieraus zwei grundlegend verschiedene Ansätze zu seiner Vermeidung ab und erläutern Sie diese anhand eines einfachen Beispiels.

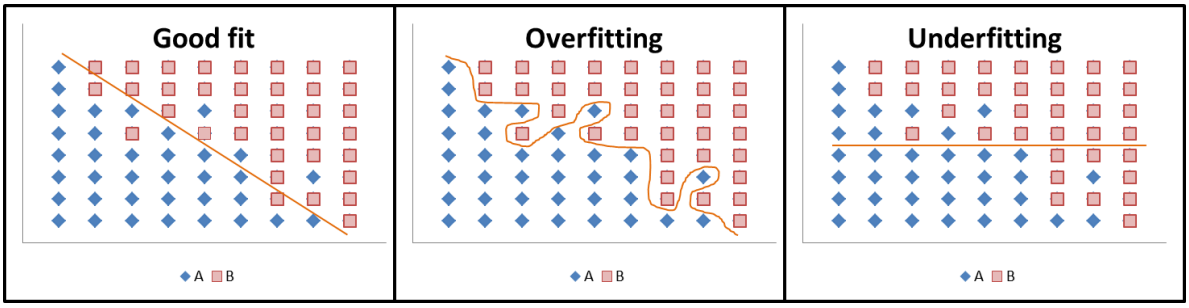
Lösungsvorschlag:

a) Von Over- und Underfitting spricht man im Zusammenhang von maschinellem Lernen bei der Beschreibung der Güte von Modellen. In beiden Fällen kann das Modell keine guten Vorhersagen für neue Daten erzeugen.

Overfitting (Überanpassung) liegt vor, wenn ein Modell zu stark an die Trainingsdaten angepasst ist. Ein überangepasstes Modell bildet ein Rauschen oder zufällige Schwankungen in den Trainingsdaten zu stark ab, um das tatsächlich zu findende Muster korrekt wiederzugeben.

Underfitting (Unteranpassung) liegt vor, wenn ein Modell zu wenig die in den Daten enthaltenen Muster wiedergibt.

Die folgenden Grafiken zeigen Extrembeispiele für Over- und Underfitting sowie zum Vergleich ein angemessen gefittetes Modell („Good fit").



b) Overfitting kann auf verschiedene Weisen erkannt werden (*die Nennung und Erläuterung von zwei Möglichkeiten genügt für das Erreichen der vollen Punktzahl*):

Unterschiedliche Modellgüte bei Trainings- und Validierungsdaten:

Ein deutlicher Hinweis auf Overfitting liegt vor, wenn das Modell auf den Trainingsdaten eine sehr hohe Güte zeigt, aber auf neuen Daten (Validierungsdaten) eine deutlich schlechtere Güte erbringt. Dies deutet darauf hin, dass das Modell zu stark an die Trainingsdaten angepasst ist und nicht in der Lage ist, die zugrunde liegenden Muster zu generalisieren.

Komplexe Modelle im Vergleich zu einfachen Modellen:

Wenn ein komplexes Modell deutlich besser auf den Trainingsdaten abschneidet als ein einfacheres Modell, aber auf Validierungsdaten nicht besser oder sogar schlechter ist, deutet dies auf Overfitting

hin. In solchen Fällen kann es sinnvoll sein, die Komplexität des Modells zu reduzieren oder Techniken wie Regularisierung anzuwenden, um Overfitting zu verhindern.

Analyse von Lernkurven:

Durch die Visualisierung von Lernkurven, die den Trainingsfehler und den Validierungsfehler im Laufe des Trainings darstellen, lässt sich Overfitting erkennen. Wenn das Modell übertrainiert ist, wird der Trainingsfehler wahrscheinlich weiter abnehmen, während der Validierungsfehler irgendwann zu steigen beginnt, nachdem er sein Minimum erreicht hat.

c) Overfitting kann insbesondere aus folgenden Gründen auftreten:

1. Zu hohe Modellkomplexität:

Wenn ein Modell zu viele Parameter und/oder eine zu hohe Komplexität besitzt, besteht die Gefahr einer zu starken Anpassung an die Trainingsdaten und damit ein Lernen des Rauschens in den Daten anstelle von tatsächlichen Mustern.

2. Unausgewogene Trainingsdaten:

Wenn die Trainingsdaten nicht repräsentativ für die Daten sind, für die das Modell zum Einsatz kommen soll, kann Overfitting auftreten. Das Modell passt sich möglicherweise zu stark an die spezifischen Eigenschaften der Trainingsdaten an und kann daher nicht gut auf neue Daten verallgemeinern.

Entsprechend ergeben sich insbesondere folgende Ansätze zur Vermeidung von Overfitting, die auch miteinander kombiniert werden können:

1. Kontrolle der Modellkomplexität:

Ziel der Steuerung der Modellkomplexität sollte ein ausgewogenes Verhältnis zwischen Modellgenauigkeit und Generalisierungsfähigkeit sein.

Zum Beispiel lässt sich bei Baumverfahren die Komplexität des Modells durch frühes Stoppen des Algorithmus bzw. durch Beschneiden der Äste (Pruning) steuern.

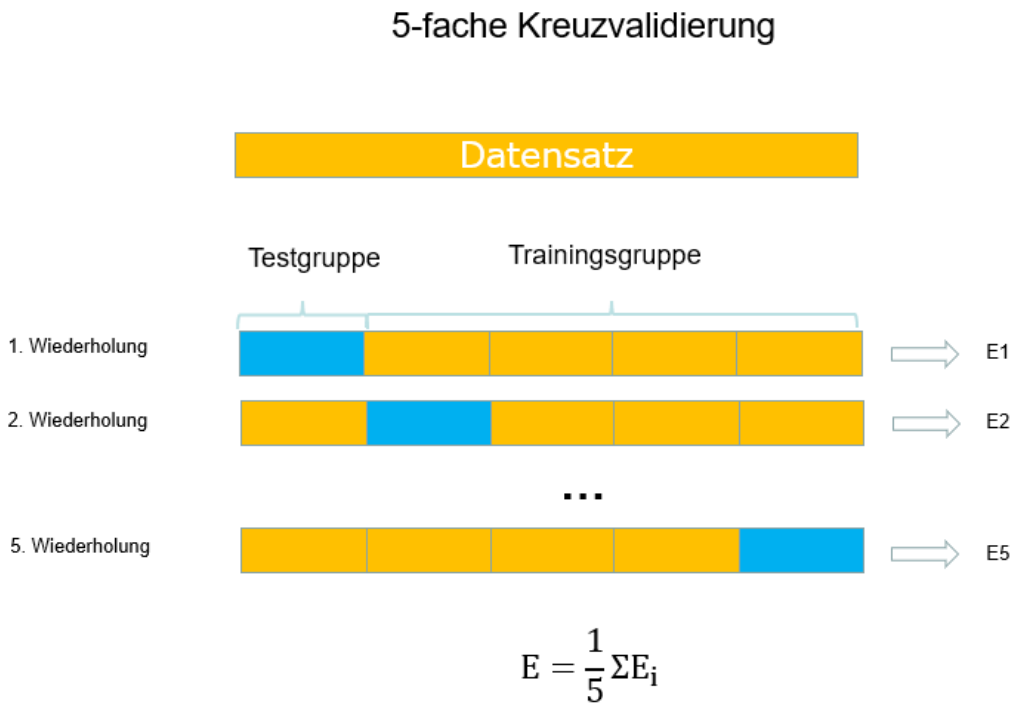
2. Verwendung von Kreuzvalidierung:

Techniken wie k-fache Kreuzvalidierung führen zu Modellen, die weniger abhängig von einzelnen Besonderheiten in den Trainingsdaten sind.

Die folgende Grafik verdeutlicht das Vorgehen bei einer 5-fachen Kreuzvalidierung. Dabei wird der gesamte Datensatz zufällig in 5 (ungefähr) gleich große Teile zerlegt und jeweils das Training auf nur 4/5 des Datensatzes durchgeführt, während das restliche Fünftel zur Validierung (in der folgenden Grafik als Testgruppe bezeichnet) herangezogen wird. In jeder der 5 Wiederholungen lässt sich ein Schätzer E_i für den Testfehler bestimmen. Hieraus ergibt sich dann der Gesamt-Testfehler zu

$$E = \frac{1}{k} \sum_{i=1}^{k} E_i \ .$$

Durch dieses Vorgehen wird vermieden, dass bei unglücklicher Auswahl der Trainingsdaten das Modell zu stark an die Besonderheiten der Trainingsdaten angepasst wird.



Anm.: Diese Grafik dient nur der Illustration und ist kein Bestandteil der geforderten Lösung.

Aufgabe 6 [2.1 Datenmanagement 2] (15 Punkte)

a) [3 Punkte] Grenzen Sie relationale Datenbanken zu NoSQL-Datenbanken ab, indem Sie drei verschiedene Vorteile von NoSQL-Datenbanken nennen.

b) [4 Punkte] Erläutern Sie das CAP-Theorem, indem Sie die Kernaussage wiedergeben sowie auf jeden der drei Buchstaben eingehen.

c) [8 Punkte] Gliedern Sie NoSQL-Implementierungen anhand von vier Beispielen. Beschreiben Sie die jeweiligen Implementierungen und geben Sie jeweils ein konkretes System an.

Lösungsvorschlag:

a) Beispielhaft können die folgenden drei Punkte genannt werden:

   a. (Bessere Skalierbarkeit) NoSQL-Datenbanken können höher skalieren als vergleichbare relationale Datenbanken. Das kann u.a. dadurch realisiert werden, dass keine hohen Ansprüche an ein Datenschemata gestellt werden und die Daten damit schnell gespeichert werden können.

   b. (Höhere Flexibilität) Verschiedene Datenbankabfragen werden durch relationale Datenbanken nicht oder nur mit sehr viel Aufwand unterstützt.

   c. (Anpassungsfähig) Durch niedrigere Ansprüche an ein Datenbankschema sind diese nicht so restriktiv gegenüber dem Datenmodell. Eine relationale Datenstruktur kann das Datenmodell sehr stark einschränken.

b) Das CAP-Theorem besagt, dass es in einem verteilten System unmöglich ist, gleichzeitig die drei Eigenschaften Consistency (Konsistenz), Availability (Verfügbarkeit) und Partition Tolerance (Ausfalltoleranz) zu garantieren.

   a. Konsistenz: Die Antworten von unterschiedlichen Systemteilen sollten konsistent sein.

   b. Verfügbarkeit: Alle Antworten an das System werden stets beantwortet (auch wenn die Kommunikation innerhalb des Systems gestört ist).

   c. Ausfalltoleranz: Das System arbeitet auch weiter, wenn es partitioniert wird (d.h. wenn Knoten nicht mehr miteinander kommunizieren können).

c) Gliederung von NoSQL-Datenbanken (Auswahl):

   a. Key-Value Store: Hierunter versteht man Verallgemeinerungen von Dictionary/Hashmap-Objekten, wie Sie aus vielen Programmiersprachen bekannt sind. Beispiele: memcached, Redis

   b. Graphendatenbank: Datenbank, die Graphen benutzt, um stark vernetzte Informationen darzustellen und abzuspeichern. Beispiel: Neo4j

c. Dokumentenorientierte Datenbank: Erweiterung der Key-Value-Stores. Grundeinheit zur Speicherung ist ein Dokument, wobei dieses typischerweise durch ein JSON-Objekt dargestellt wird. Beispiele: Elasticsearch, MongoDB.

d. Spaltenorientierte Datenbanken: Ein Datenbankmanagementsystem, welches seine Inhalte spaltenweise (und nicht zeilenweise) physisch abspeichert. Beispiele: Apache Cassandra, Google Big Table.

Aufgabe 7 [4.2 Deep Learning 2] (20 Punkte)

a) [3 Punkte] Erklären Sie den Begriff Konvolution. Gehen Sie dabei auch auf die Begriffe Padding und Stride ein.

b) [8 Punkte] Gegeben sei die folgende zweidimensionale Matrix

10	10	10	10	10	10
10	10	10	10	10	10
0	0	0	0	10	10
0	0	0	0	10	10
0	0	0	0	10	10

zusammen mit dem Filter:

-1	-2	-1
0	0	0
1	2	1

Berechnen Sie die Konvolution (verwenden Sie einen Stride von 1 und kein Padding). Wozu dient der angegebene Filter?

c) [5 Punkte] Beschreiben Sie die Funktionsweise von Pooling. Warum wird es verwendet? Zu nennen sind, neben Max Pooling, zwei weitere Arten. Im Anschluss ist ein Max Pooling mit einem 2x2 Filter (f=2) auf das Ergebnis aus Teil b) anzuwenden.

d) [4 Punkte] Geben Sie zwei Regularisierungsverfahren für neuronale Netzwerke an und beschreiben Sie diese.

Lösungsvorschlag:

a) Bei der Konvolution (auch: Faltung) einer Matrix wird diese stückweise mit einer kleineren Matrix multipliziert. Dabei wird prinzipiell eine kleinere Matrix (z.B. 3x3) schrittweise über ein „Bild" geschoben. Dabei bezeichnet der Stride die Schrittweite: Bei einem Stride von 1 wird die Verschiebung um „einen Pixel" nach rechts bzw. um „einen Pixel" nach unten vorgenommen. Über Padding werden „virtuelle Randpixel" zur Berechnung der Konvolution hinzugefügt. Dabei wird angegeben, wie viele „Pixel" tief dieser Rand sein soll. Typischerweise werden als Inhalte Nullen verwendet, es sind jedoch auch andere Werte denkbar. Damit kann man den beiden folgenden Punkten entgegenwirken:

   a. Die Ergebnismatrix ist kleiner als das ursprüngliche Bild

   b. Pixel am Rand gehen weniger in die Ergebnismatrix ein

b) Als Ergebnis erhalt man die folgende 3x4 Matrix:

-40	-40	-30	-10
-40	-40	-30	-10
0	0	0	0

An diesem Ergebnis ist erkennbar, dass es sich um einen Detektor für horizontale Kanten handelt.

c) Beim Pooling geht es lediglich darum, die Größe der Eingabe zu reduzieren, um damit die Anzahl der Parameter zu verringern. Damit wird die Generalisierung erhöht um größere „Toleranz" zu erwirken. Analog der Konvolution wird ein Filter über die Eingabe geschoben und eine einfache Operation (z.B. Ermittlung eines Maximalwertes) ausgeführt. Alternativ kann noch der Durchschnitt (Average Pooling) oder die L2-Norm (L2 Norm Pooling) verwendet werden. Für das Beispiel aus Teil b) ergibt sich:

-40	-30	-10
0	0	0

d) Drei gebräuchliche Regularisierungsverfahren für neuronale Netzwerke sind die L1, L2 und Dropout.

Bei der L1 bzw. der L2-Regularisierung wird die Verlustfunktion um einen zusätzlichen Term erweitert. Hierbei geht die L1 bzw. L2-Norm der Gewichtsmatrizen in diesen Strafterm ein. Dieser Strafterm dient dazu, die jeweiligen Gewichte klein zu halten (bzw. auf 0 zu schrumpfen). Damit verringert sich die Komplexität des Modells.

Beim Dropout werden beim Trainieren des Modells zufällige Neuronen ausgeschaltet (d.h. mit 0 multipliziert). Bei der Anwendung des Modells wird jedes Neuron mit der entsprechenden Inklusionswahrscheinlichkeit multipliziert.

Aufgabe 8 [4.1 Überwachtes Lernen 2] (35 Punkte)

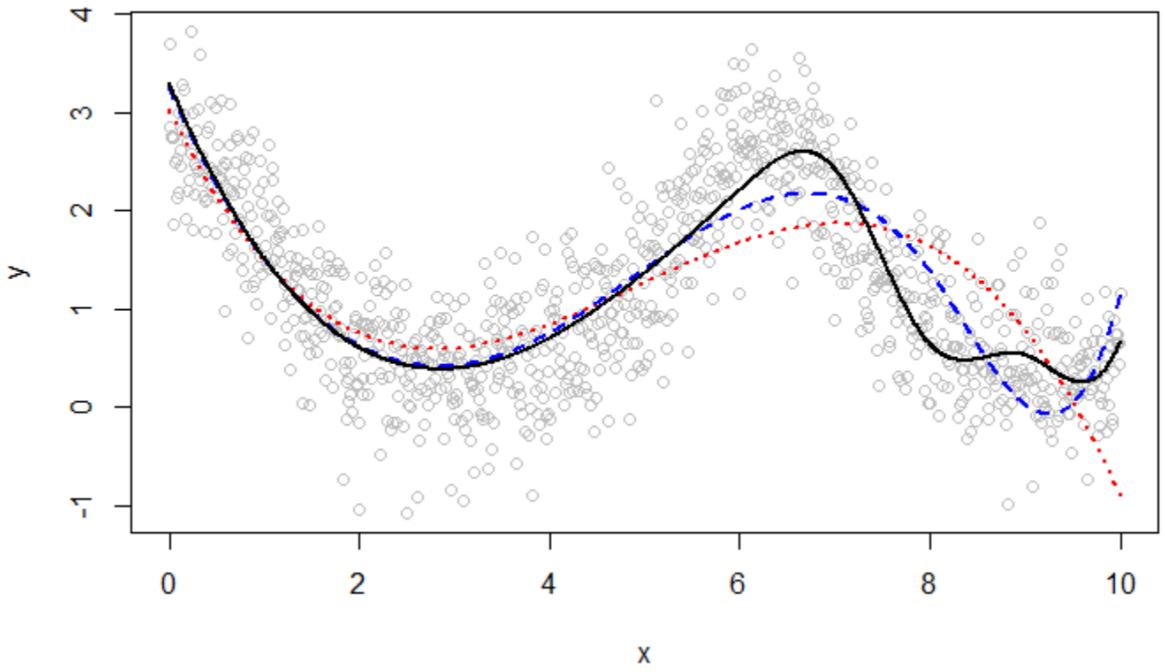
a) [10 Punkte] Bei Ihren Analysen in einem Data Warehouse haben Sie den im Bild (s.u.) dargestellten Datensatz vor sich. Für eine Modellierung verwenden Sie zwei verschiedene Methoden. Zunächst eine polynomiale Modellierung:

```
fit_poly <- lm(y ~ poly(x, 3), data=df)
pred_poly <- predict(fit_poly, newdata=list(x=x.grid), se=TRUE)
```

Zusätzlich verwenden Sie eine Modellierung mit Splines, wobei Sie verschiedene Knotenanzahlen verwenden (einmal drei und einmal sechs Knoten). Die Modellierung erfolgt nach dem folgenden Schema:

```
fitsp1 <-  lm(y~bs(x, knots=c(...)), data=df)
predsp1 <- predict(fitsp1, newdata=list(x=x.grid), se=TRUE)
```

Die Datenpunkte und die Ergebnisse der Modellierung sind in der folgenden Abbildung dargestellt:



Leider ist die Zuordnung, welche Kurve zu welchem Modell gehört, nicht mehr vorhanden. Orden Sie den Kurven (schwarz durchgezogen,

blau gestrichelt und rot gepunktet) mit einer Begründung einem entsprechenden Modell zu. Welche Splines werden zur Modellierung verwendet? Gehen Sie auch darauf ein, wo Sie (begründet!) die Lage der Knoten vermuten!

b) [7 Punkte] Sie stellen sich die Frage, wie viele Knoten bei der Modellierung in Teil (b) berücksichtigt werden sollten. Sie haben mehrere Knotenmengen (bezeichnet mit K_1, K_2, …, K_n) zur Verfügung. Wie können Sie sich entscheiden? Ist die alleinige Betrachtung der Residuenquadratsumme eine gute Idee (Begründung)? Wie kann man alternativ vorgehen?

c) [8 Punkte] Bei einer Modellierung betrachten Sie die folgende Funktion:

$$s(x) = \begin{cases} (x+1) + (x+1)^3, -1 \leq x < 0 \\ 4 + (x-1) + (x-1)^3, 0 \leq x \leq 1 \end{cases}$$

Welche Eigenschaften eines natürlichen kubischen Splines bzgl. den Knoten $x_0 = a = -1$, $x_1 = 0$ und $x_2 = b = 1$ liegen vor bzw. liegen nicht vor? Fertigen Sie eine Skizze an.

d) [10 Punkte] Gegeben sind die beiden folgenden Kurven $\hat{h}_1$ und $\hat{h}_2$:

$$\hat{h}_1 = arg\min_g \left(\sum_{i=1}^n \left(y_i - h(x_i)\right)^2 + \lambda \int \left[h^{(3)}(x)\right]^2 dx \right),$$

$$\hat{h}_2 = arg\min_g \left(\sum_{i=1}^n \left(y_i - h(x_i)\right)^2 + \lambda \int \left[h^{(4)}(x)\right]^2 dx \right)$$

Dabei bezeichnet $h^{(m)}$ die m-te Ableitung von h, und λ eine Regularisierungsparameter. Diskutieren und begründen Sie die folgenden Punkte:

a. Für $\lambda \to \infty$, liegt der kleinere Trainingswert für RSS bei $\hat{h}_1$ oder $\hat{h}_2$?

b. Für $\lambda \to \infty$, liegt der kleinere Testwert für RSS bei $\hat{h}_1$ oder $\hat{h}_2$?

c. Für $\lambda = 0$, hat $\hat{h}_1$ oder $\hat{h}_2$ den geringeren Trainings- und Test RSS?

Lösungsvorschlag:

a) Bei dem polynomialen Fit handelt es sich um ein Polynom vom Grad drei. In der Graphik ist nur die in rot (gepunktet) eingezeichnete Kurve zu einem Polynom vom Grad drei ähnlich. Damit handelt es sich bei der schwarzen (durchgezogen) und blauen (gestrichelten) Kurve um Splineschätzungen. R verwendet hier standardmäßig kubische Polynome. Beide Kurven haben im Intervall von 0 bis 6 einen ähnlichen Verlauf. Unterschiede gibt es im Intervall von 6 bis 10. Die schwarze Kurve hat hier einen deutlich unruhigeren Verlauf, was darauf hindeutet, dass sich in diesem Bereich mehrere Knoten befinden, die diesen Verlauf verantworten.

Anmerkung: Bei diesen simulierten Daten wurden für die blaue Kurve nur die Knotenpunkte 0, 4 und 8 verwendet. Für die schwarze Kurve hingegen 6, 7, 8, 9 und 10.

b) Die alleinige Betrachtung von RSS ist keine gute Idee, da man hierbei nur von einem gemessenen Wert ausgeht. Besser wäre folgendes Vorgehen: Einteilung des Datensatzes in Trainings- und Validierungsteil. Training des Modells auf dem Trainingsteil, Anwendung des Modells auf dem Validierungsteil. Dieses Verfahren wiederholen und RSS als Durchschnittswert berechnen. Das kann für die verschiedenen Knotenmengen K_1, …, K_n durchgeführt werden. Die Menge K mit dem kleinsten durchschnittlichsten RSS wird schließlich ausgewählt („Kreuzvalidierung").

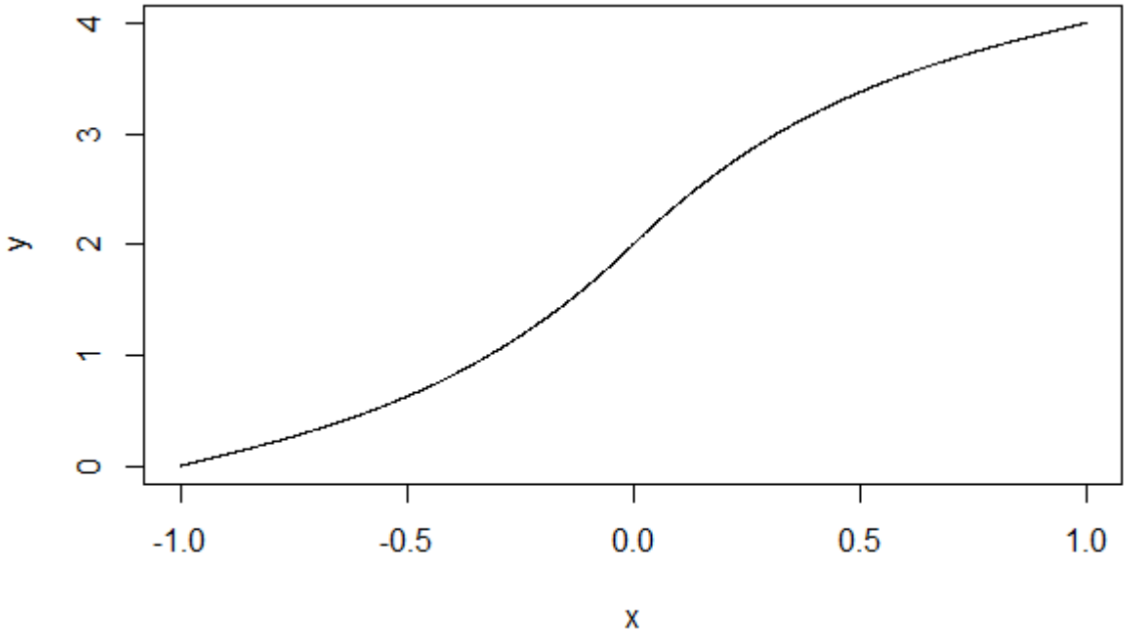
c) Bezeichne mit $f_1(x) = (x+1) + (x+1)^3$ und $f_2(x) = 4 + (x-1) + (x-1)^3$. Daraus ergibt sich $f_1'(x) = 1 + 3(x+1)^2, f_1''(x) = 6(x+1)$ und $f_2'(x) = 1 + 3(x-1)^2, f_2''(x) = 6(x-1)$.

Damit folgt:

- $f_1(0) = f_2(0) = 2$
- $f_1'(0) = 4 = f_2'(0)$
- $f_1''(0) = 6 \neq f_2''(0) = -6$
- $f_1''(-1) = 0 = f_2''(1)$

Auf den jeweiligen Intervallen liegt eine Funktion vom Grad 3 vor. Jedoch ist eine Glattheitsbedingung verletzt.

Skizze der Funktion:



d) Die Punkte können wie folgt beantwortet werden:

a. Von den beiden betrachteten Funktionen ist die Funktion $\hat{h}_2$ flexibler, da eine höhere Ableitung im Strafterm verwendet wird. Sie wird also versuchen noch glattere Funktionen zu bevorzugen. Daher ist für $\hat{h}_2$ ein geringerer RSS-Wert beim Training zu erwarten.

b. Eine pauschale Aussage ist schwierig zu treffen. Durch die höhere Flexibilität der Funktion $\hat{h}_2$ kann argumentiert werden, dass die Gefahr von Overfitting größer ist, und damit der kleinere Testfehler bei $\hat{h}_1$ liegt.

c. Für den Fall $\lambda = 0$ ist der Strafterm nicht vorhanden, d.h. hier liegt eine Interpolation der Trainingsdaten vor. In diesem Fall werden $\hat{h}_1$ und $\hat{h}_2$ gleich sein. Damit liegt der Wert von RSS im Fall des Trainings bei 0. Andererseits kann ein hoher Test-RSS Wert erwartet werden.

DAV

DEUTSCHE
AKTUARVEREINIGUNG e.V.

Written examination in special knowledge

Actuarial Data Science Advanced

According to Prüfungsordnung 5
of Deutsche Aktuarvereinigung e. V.

On the 18th of October 2024

Notes:

- A calculator is permitted as an aid.
- The total score is 180 points. The exam is passed if at least 90 points are achieved.
- Please check the exam you have been given for completeness. The exam consists of 33 pages.
- All answers must be justified and the solution to arithmetic problems must be clear.

Members of the examination commission:

Axel Kiermaier, Dr. René Külheim, Dr. Jonas Offtermatt,
Dr. Felix Spangenberg

Exercise 1 [1.2 Data protection 2, 3.3 Visualization 1, 3.4 Innovative products 1] (28 points)

Your company is about to make an innovation in product development. Data science methods are to be used in product development for the first time.

a) [6 points] You are to explain the advantages of data science in product development to your colleagues in a presentation. To do this, name 4 advantages that result from the use of data science methods in product development.

b) [6 points] Following your presentation, you are approached by your company's data protection officer. He has concerns about Article 25 of the GDPR. Explain to him how the principles of data protection through privacy by design and privacy by default can help to develop innovative products that respect data protection and increase customer trust. In particular, explain what is meant by privacy by design and privacy by default.

c) [8 points] After a few weeks of project work, your colleagues have fitted the first model. They present you with the following results:

Data point	1	2	3	4	5	6	7	8
y_test	84	80	95	85	85	94	100	92
y_predicted	75	71	90	70	74	88	86	84

Draw a scatter plot y_test against y_predicted and a residual plot based on the table. Please also pay attention to the basic requirements for a visualization (e.g. use of a ruler, axis labelling, title, ...).

d) [4 points] Interpret the results of the visualizations.

e) [5 points] Drawing visualizations by hand is quite time-consuming. Apart from at school and in actuarial exams, nobody does this anymore. Which Python commands could you use to create the ScatterPlot? (Don't forget the necessary imports).

Proposed Solution:

a) Possible advantages are:

Improved decision making: By using data science, large amounts of data can be analyzed to make informed decisions throughout the product development process. This makes it possible to identify trends, better understand customer needs and optimize product performance.

Increased efficiency: Data science enables the automation of processes in product development, such as the prediction of production times, the optimization of supply chains or the early detection of potential problems. This can save time and resources and speed up product development.

Personalized product design: By analysing customer data and behaviour, products can be individually adapted to customers' needs and preferences. Data science enables the development of customized products that offer greater added value and increase customer satisfaction.

Competitive advantage: Companies that integrate data science into their product development can gain a competitive advantage by bringing innovative products to market faster while improving the quality and effectiveness of their products. By using data, they can also identify trends at an early stage and adapt better to market changes.

Further benefits can be mentioned. 1.5 points per advantage.

b) Privacy by design and privacy by default are principles that ensure that data protection is integrated into the development of products from the outset.

Privacy by design refers to the integration of data protection features into the architecture and design of products and services. This means that data protection is taken into account from the outset, for example by implementing privacy-friendly default settings, applying data minimization and anonymization and ensuring the security of data transmission.

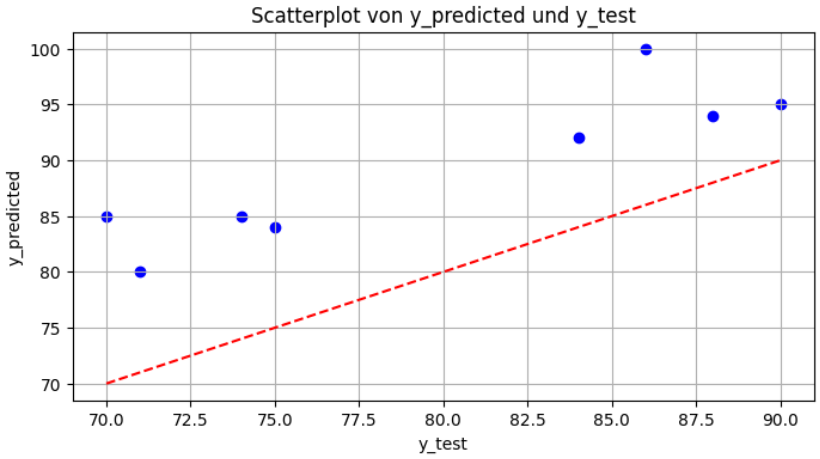
Privacy by default means that data protection is set as standard without the user having to make any manual settings. This means that the strictest data protection options are automatically activated and the user must actively make other settings if they want to change the data protection.

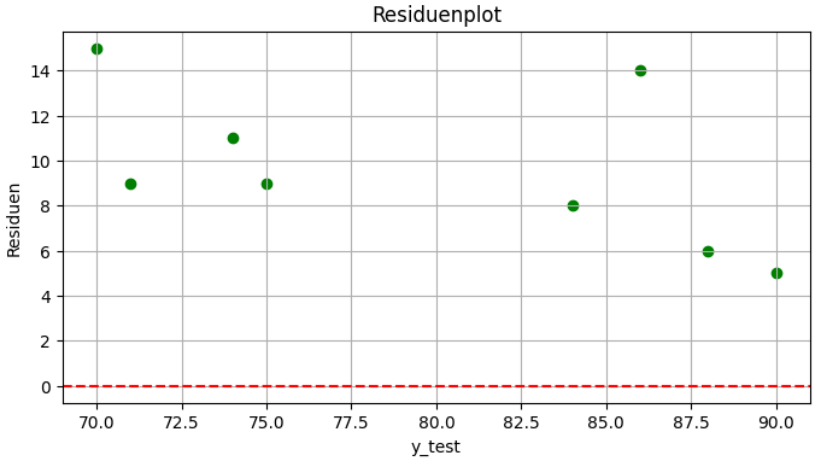
Both principles help to increase consumer confidence in the security of their personal data and minimize the risk of data breaches. By integrating

data protection into the development process from the outset and setting it as a standard, companies can develop innovative products that respect data protection while complying with legal requirements, in particular the provisions of the GDPR.

2 points per explanation of a principle and 2 points for the general explanation of why innovative products are designed with it.

c) The visualization should look like :





As always, there are deductions for missing labels, unclean drawings, etc.

d) Since two separate clusters of points lie far from the normal line in the scatter plot, this indicates that there is a strong non-linear relationship between the variables. This means that the data points do not scatter along a straight line, but rather have a curved or skewed shape.

The fact that the residuals show a clear distance from the zero line indicates that the model cannot explain all variations in the dependent varia-

ble. The residuals are the difference between the observed values and the predicted values of the model. If the residuals have a large distance from the zero line, this means that the model makes large errors in predicting the dependent variable. A large distance from the zero line indicates that the model has systematic errors and an inadequate fit to the data.

Overall, the scatterplot and residual plot suggest that the current model may not be adequate to describe the relationship between the variables. More complex models or non-linear transformations of the data may need to be considered to better capture the relationship and improve prediction accuracy.

e) The python code should look like:

```
import matplotlib.pyplot as plt
# Data
y_predicted = [84, 80, 95, 85, 85, 94, 100, 92]
y_test = [75, 71, 90, 70, 74, 88, 86, 84]

# Scatterplot
plt.figure(figsize=(8, 4))
plt.scatter(y_test, y_predicted, color='blue')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)],
color='red', linestyle='--')
plt.xlabel('y_test')
plt.ylabel('y_predicted')
plt.title('Scatterplot von y_predicted und y_test')
plt.grid(True)
plt.show()
```

Exercise 2 [3.1 Data Mining 2, 4.4 Data preparation for model creation] (16 points)

a) [2 points] Name two alternative data mining process models to CRISP-DM.

b) [9 points] Describe the three types of missing values in data records according to Rubin's typology with their respective abbreviations, long names and give an example from the insurance context for each type.

c) [5 points] Perform one-hot encoding / dummy encoding on the Claim_Location column on the following data set.

Year	Contract_ID	Claim_Location	Claim_Amount
2022	1	City	1500
2022	2	Rural	800
2022	3	Suburb	2000
2022	4	City	1200
2022	5	Suburb	1800

Proposed Solution:

a) Two possible alternative data mining processes to CRISP-DM are:

SEMMA: SEMMA stands for Sample, Explore, Modify, Model, Assess and was developed by SAS. This process begins with the selection of a sample of data, followed by exploration of the data, modification of variables and model building. The models are then assessed and validated.

KDD process (Knowledge Discovery in Databases): The KDD process is a broader framework that goes beyond pure data mining and includes other aspects of the knowledge discovery process, such as pre-processing data, interpreting results and integrating the knowledge into the business process. It comprises six phases: Selection, Preprocessing, Transformation, Data Mining, Interpretation/Evaluation and Integration/Utilization.

Further entries are possible. Simply mentioning the process names is sufficient.

b) According to Rubin's typology, there are three types of missing values in data sets:

MCAR (Missing Completely at Random):

Description: In this type of missing value, the observation value is missing completely at random, independent of other variables or observations.

Example from the insurance context: A questionnaire on customer satisfaction is sent by post, but some customers never receive the questionnaire because the post is lost by chance.

MAR (Missing at Random):

Description: In this type of missing value, the observation value is missing at random, but dependent on other variables in the data set.

Example from the insurance context: Customers are asked to provide their place of residence and income data in a survey, but some customers (from "better" residential areas) do not provide the income data.

MNAR (Missing Not at Random):

Description: In this type of missing value, the observed value is not missing at random and is dependent on unobserved variables or the missing values themselves.

Example from the insurance context: Customers provide their health information, but customers with serious pre-existing conditions may tend to deliberately omit this information.

3 points per type, 1 point each for abbreviation and long name, one for the description and one for the example.

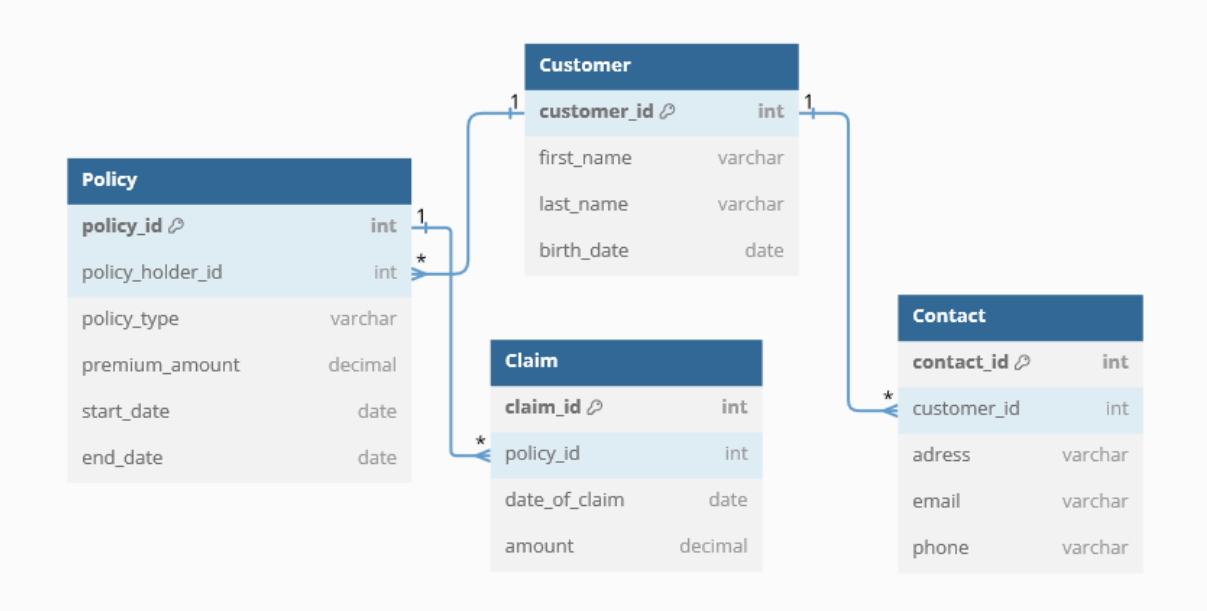
c) Data-Set after One-Hot-Encoding:

Year	Contract_ID	Claim_Location_City	Claim_Location_Rural	Claim_Location_Suburb	Claim_Amount
2022	1	1	0	0	1500
2022	2	0	1	0	800
2022	3	0	0	1	2000
2022	4	1	0	0	1200
2022	5	0	0	1	1800

One point per correct row.

Exercise 3 [2.1 Data management 2] (21 points)

The following entity-relationship diagram is given:



And the following sample data records for the `Policy` and `Customer` tables:

poli-cy_id	poli-cy_holde r_id	policy_type	premi-um_amount	start_date	end_date
1	1	Auto	500	2023-01-01	2024-01-01
2	2	Haus	1000	2023-02-01	2024-02-01
3	3	Leben	1500	2023-03-01	2024-03-01

customer_id	first_name	last_name	birth_date	contact_id
1	Max	Mustermann	1990-01-01	1
2	Anna	Schmidt	1985-05-15	2
3	Peter	Müller	1978-11-30	3

a) [5 points] Identify and describe the different entities and relationships of the diagram. Explain how these elements are connected to each other and what information they represent.

b) [4 points] Explain the terms primary key and foreign key, each using an example from the diagram above.

c) [3 points] Write an SQL query to retrieve all customers with their names and dates of birth from the "`Customer`" table.

d) [3 points] Write an SQL query to insert a new claim with the `values policy_id = 1, date_of_claim = "2024-01-01"` and `amount = "12345.00"` into the "`Claim`" table.

e) [6 points] State the result of each of the following three SQL commands:

i)

```
SELECT Policy.policy_id, Policy.policy_type, Customer.first_name,
Customer.last_name

FROM Policy
INNER JOIN Customer ON Policy.policy_holder_id = Customer.customer_id
WHERE Policy.start_date >= '2023-02-01' AND Policy.end_date <= '2024-
02-01';
```

ii)

```
SELECT Policy.policy_id, Policy.policy_type, Customer.first_name,
Customer.last_name
FROM Policy
LEFT JOIN Customer ON Policy.policy_holder_id = Customer.customer_id
WHERE Customer.customer_id IS NULL;
```

iii)

```
SELECT Policy.policy_id, Customer.first_name, Customer.last_name,
Policy.premium_amount
FROM Policy
LEFT JOIN Customer ON Policy.policy_holder_id = Customer.customer_id
WHERE Policy.premium_amount >= 1000
```

Proposed Solution:

a) Entities and attributes:

Insurance policy (Policy): Contains information such as policy type, premium amount and validity period.

Customer (Customer): Stores customer information such as name and date of birth.

Contact details: Stores address and contact details.

Claim: Holds data on claims such as date and claim amount.

Relationships:

Between "insurance policy" and "customer": A customer can have several insurance policies, but a policy only belongs to one customer.

Between "Insurance policy" and "Claim": An insurance policy can have several claims, but a claim is only associated with one policy.

Between "customer" and "contact data": A customer can have multiple contact data, but a given set of contact data belongs to exactly one customer.

These entities and relationships enable the insurance company to efficiently manage and link customers, insurance policies and claims. (1 point per entity and relationship)

b) **Primary key:** A primary key uniquely identifies each row in a table. In our diagram, for example, the customer number ("customer_id") in the "Customer" table is a primary key.

Foreign key: A foreign key is a column in a table that refers to the primary key of another table. For example, the "policy_holder_id" column in the "Insurance policy" table refers to the customer number in the "Customer" table. It shows which customer the insurance policy belongs to.

c) The SQL command should be:

```
SELECT first_name, last_name, birth_date

FROM Customer;
```

d) The SQL command should be:

```
INSERT INTO Claim (policy_id, date_of_claim, amount)

VALUES (1, '2024-01-01', 12345.00);
```

e) Result of the SQL command with INNER JOIN and additional conditions:

```
| policy_id | policy_type | first_name | last_name |
|-----------|-------------|------------|-----------|
| 2         | Haus        | Anna       | Schmidt   |
```

Result of the SQL command with LEFT JOIN and conditions for missing customer data:

```
No records found.
```

Result of the SQL command with LEFT JOIN and conditions for premium over 1000:

```
| policy_id | first_name | last_name | premium_amount |
|-----------|------------|-----------|----------------|
| 2         | Anna       | Schmidt   | 1000           |
| 3         | Peter      | Müller    | 1500           |
```

Exercise 4 [4.3 Correlation & Causal Inference] (30 points)

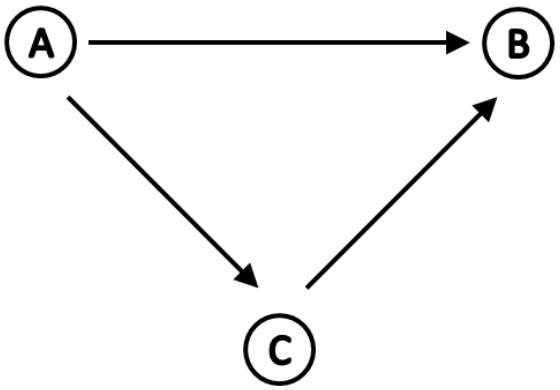
a) [6 points] Give a mathematical definition for DAGs and explain the connection to the mathematical treatment of causality. What role do the "D" and the "A" in "DAG" play in relation to the properties of a causal relationship?

b) [9 points] Draw and name the three basic structures ("basic forms") that can occur in DAGs. Give a simple example of each from the insurance environment.

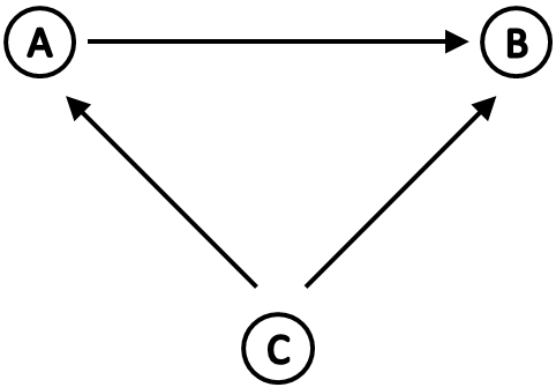
c) [7 points] What is the relationship between DAGs and the concept of stochastic independence and how is this expressed in mathematical formulas? How does this show that the conventional statistical calculus is not suitable for expressing causality, and how can this problem be solved according to Pearl?

d) [8 points] The following four graphs focus on the causal relationship between *A* and *B*. What role does *C* play in this context in each case? Determine the joint probability distribution *P(A,B,C)* in each case.

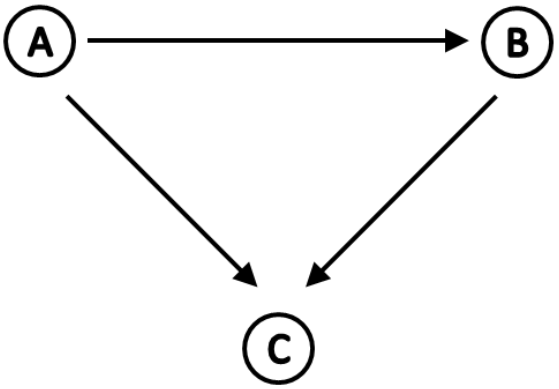
(d1):



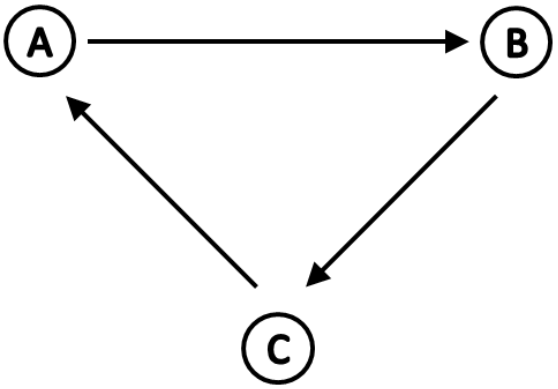
(d2):



(d3):



(d4):



Proposed Solution:

a) A DAG ("directed acyclic graph") is a term from graph theory. Like any graph, a DAG consists of nodes and edges.

A graph is a pair $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each edge e in E is assigned to an unordered pair (u,v) of nodes in V ("e connects u and v").

In a DAG, the edges have a direction ("directed") and there are no cycles ("acyclic"), i.e. no path (no sequence of edges) along the direction of the edges that leads back to the starting node.

In the mathematical theory of causality, the nodes are regarded as random variables and the edges as cause-effect relationships; an edge between two random variables means the existence of a direct cause.

In this interpretation, the "D" ("directed") and the "A" are essential to depict the properties of the causal relationship:

- The causal relationship is asymmetrical, i.e. if A is the cause of B, B cannot be the cause of A. Therefore, the graph must be directed ("D")
- The causal relationship is transitive, i.e. if A is the cause of B and B is the cause of C, then A is the cause of C. Therefore - in conjunction with the asymmetry - the graph must be acyclic ("A").

b) The three basic structures ("basic forms") that can occur in DAGs are chain, fork and collider. They each show relationships between three random variables X, Y and Z.

1. Chain:

This linear structure shows a successive dependency between the three variables.

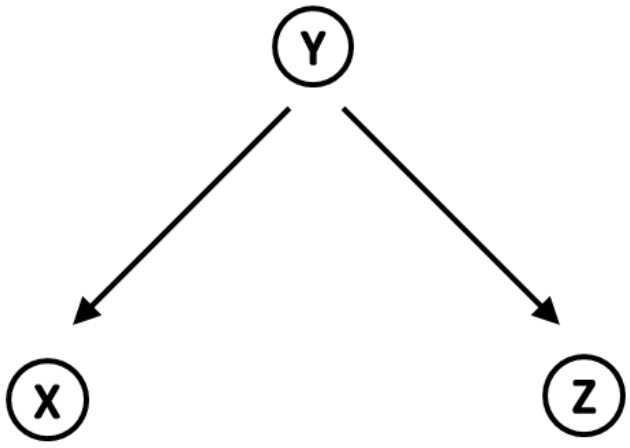
Chain

Example from the insurance sector:

Fire → Smoke → Smoke detector goes off

2. Fork:

This bifurcation structure shows a common cause *Y* for the two variables *X* and *Z*.



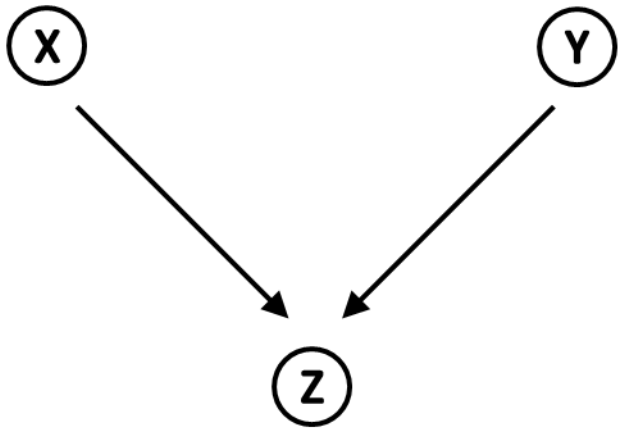
Fork

Example from the insurance sector:

Probability of death ← Age → Driving Behaviour

3. Collider:

This confluence structure shows a situation in which two variables *X* and *Y* act as a cause for a variable *Z*, which can lead to an apparent correlation between them.



Collider

Example from the insurance sector:

Annual kilometers driven → Accident ← Horsepower

c) The following applies to the three basic structures from b):

Chain: Given Y, X and Z are conditionally independent ($X \perp\!\!\!\perp Z|Y$), i.e.

$$P(X,Z|Y) = P(X|Y) \cdot P(Z|Y).$$

Fork: Given Y, X and Z are conditionally independent ($X \perp\!\!\!\perp Z|Y$), i.e.

$$P(X,Z|Y) = P(X|Y) \cdot P(Z|Y).$$

Collider: Given Z, X and Y are conditionally dependent ($X \not\perp\!\!\!\perp Y|Z$), i.e.

$$P(X,Y|Z) \neq P(X|Z) \cdot P(Y|Z).$$

The chain $X \rightarrow Y \rightarrow Z$ and the fork $X \leftarrow Y \rightarrow Z$ have the same probability distribution according to the conditional independence mentioned above, but different causal structures. This means that the language of the traditional probability distribution is not sufficient to depict these causal differences.

However, this is achieved with the do operator introduced by Pearl, which depicts interventions. $P(Y|X=x_1)$, the traditional conditional probability, only reflects the observed probability for Y if X happens to be x_1. In contrast, $P(Y|do(X=x_1))$ is the probability for Y that applies to the intervention in which X is "set to x_1". The following generally applies

$P(Y|X=x_1) \neq P(Y|do(X=x_1))$.

This is the difference between seeing / observing and doing.

d) In (d1) the effect of *A* on *B* is not (only) direct, but partly via *C*. (Note: *C* is also referred to here as a mediator.) To determine the direct effect of *A* on *B*, the total effect of *A* on *B* (e.g. by regression) and the mediation effect of *C* (e.g. by path analysis) could be determined separately using suitable methods and the difference calculated from this.

$$P(A,B,C) = P(B|A,C)\,P(A,C) = P(B|A,C)\,P(C|A)\,P(A)$$

In (d2), *C* is a confounder for the effect of *A* on *B*. It influences both *A* and *B*. A confounder can lead to incorrect conclusions if it is not adequately controlled or taken into account.

$$P(A,B,C) = P(B|A,C)\,P(A,C) = P(B|A,C)\,P(A|C)\,P(C)$$

In (d3), *C* is a collider. Given *C*, *A* and *B* (even if they are independent) become conditionally dependent. This can lead to counterintuitive correlations.

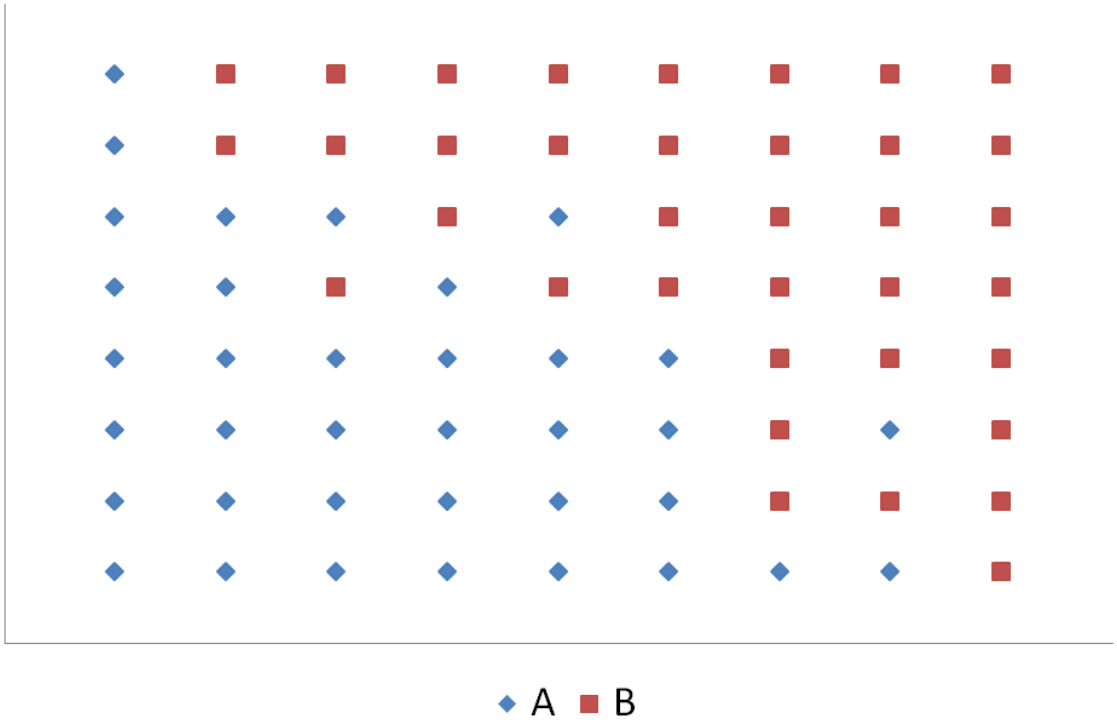
$$P(A,B,C) = P(C|A,B)\,P(A,B) = P(C|A,B)\,P(B|A)\,P(A)$$

There is no DAG in (d4). The graph is cyclic and therefore not suitable for mapping a causal relationship.

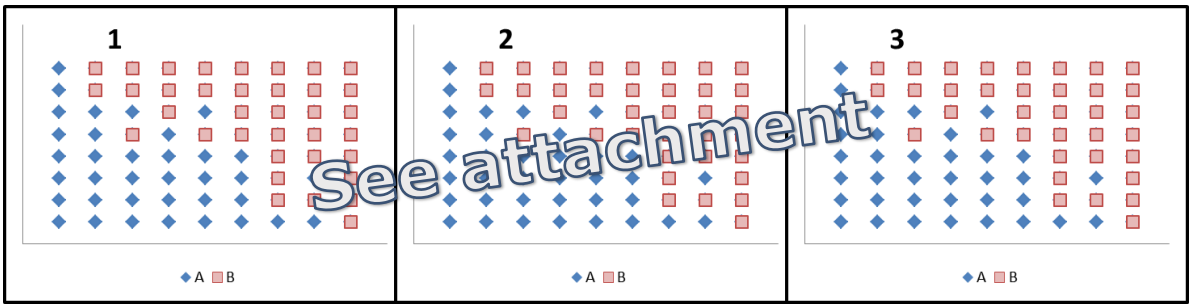
In particular, $P(A,B,C)$ cannot be factorized in a similar way.

Exercise 5 [3.2 Analytics 2] (15 points)

a) [6 points] What is meant by overfitting and underfitting? Explain this in general verbally and by drawing smooth curves of your choice in the following diagram. This shows values A and B of a binary characteristic as a function of two independent characteristics.



Note: To answer task a), please use the diagrams attached to the exam.



Make sure you also hand in the page you have edited!

b) [4 points] How can overfitting be recognized? Name and explain two different possibilities.

c) [5 points] How can overfitting occur? Derive two fundamentally different approaches to avoid it and explain them using a simple example.

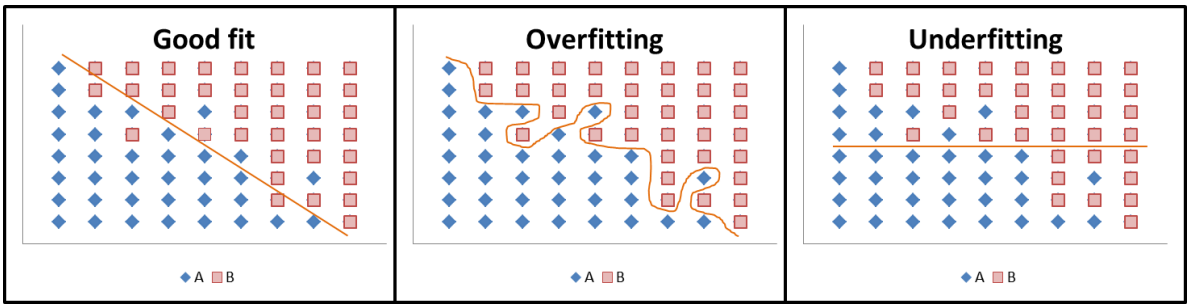
Proposed Solution:

a) Over- and underfitting are used in the context of machine learning to describe the quality of models. In both cases, the model cannot produce good predictions for new data.

Overfitting occurs when a model is too strongly adapted to the training data. An overfitting model reproduces noise or random fluctuations in the training data too strongly to correctly reproduce the actual pattern to be found.

Underfitting occurs when a model does not reproduce enough of the patterns contained in the data.

The following graphs show extreme examples of overfitting and underfitting and, for comparison, an appropriately fitted model ("good fit").



b) Overfitting can be recognized in various ways *(naming and explaining two possibilities is sufficient to achieve full marks)*:

Different model quality for training and validation data:

There is a clear indication of overfitting if the model shows a very high quality on the training data but a significantly poorer quality on new data (validation data). This indicates that the model is too strongly adapted to the training data and is not able to generalize the underlying patterns.

Complex models compared to simple models:

If a complex model performs significantly better on the training data than a simpler model, but is no better or even worse on validation data, this indicates over-fitting. In such cases, it may be useful to reduce the complexity of the model or apply techniques such as regularization to prevent overfitting.

Analysis of learning curves:

Overfitting can be detected by visualizing learning curves that show the training error and validation error over the course of training. If the model is overtrained, the training error will probably continue to decrease, while the validation error will start to increase at some point after it has reached its minimum.

c) Overfitting can occur in particular for the following reasons:

1. Model complexity too high:

If a model has too many parameters and/or too much complexity, there is a risk of over-fitting to the training data and thus learning the noise in the data instead of actual patterns.

2. Unbalanced training data:

If the training data is not representative of the data for which the model is to be used, overfitting may occur. The model may adapt too strongly to the specific characteristics of the training data and therefore cannot generalize well to new data.

Accordingly, the following approaches in particular can be used to avoid overfitting, which can also be combined with each other:

1. Control of model complexity:

The aim of controlling model complexity should be to achieve a balance between model accuracy and generalization capability.

For example, in tree methods, the complexity of the model can be controlled by stopping the algorithm early or by pruning the branches.

2. Use of cross-validation:

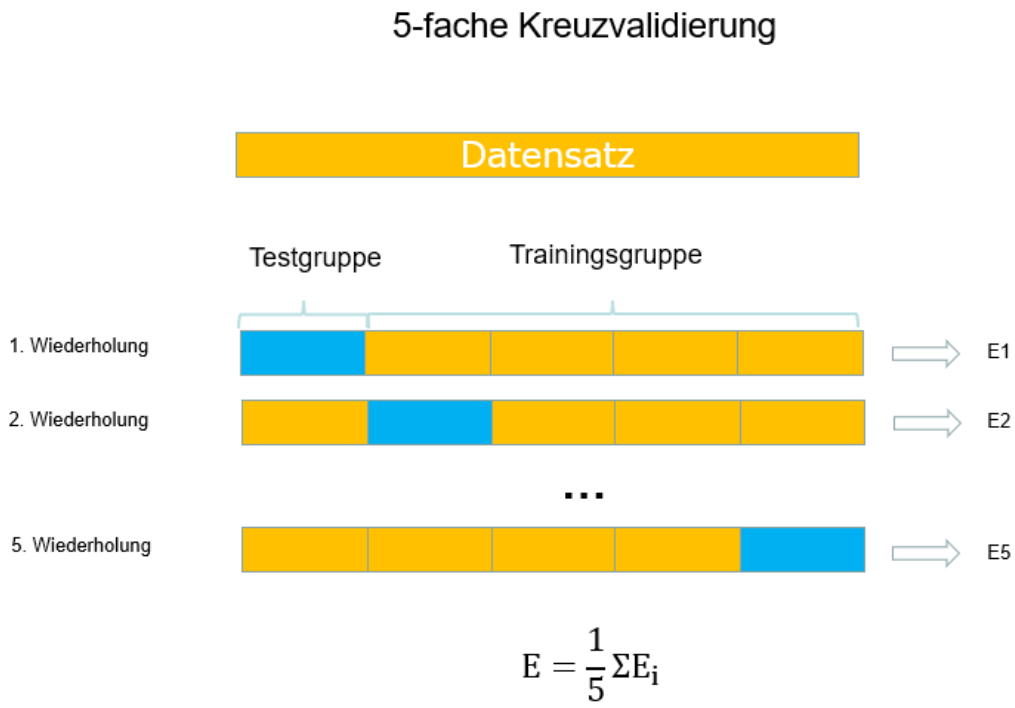
Techniques such as k-fold cross-validation lead to models that are less dependent on individual features in the training data.

The following diagram illustrates the procedure for 5-fold cross-validation. The entire data set is randomly divided into 5 (approximately) equal parts and training is carried out on only 4/5 of the data set, while the remaining fifth is used for validation (referred to as

the test group in the following diagram). An estimator E_i for the test error can be determined in each of the 5 repetitions. This then results in the total test error of

$E = \frac{1}{5} \sum_{i=1}^{5} E_i$.

This procedure prevents the model from being adapted too strongly to the special features of the training data if the training data are not selected correctly.



Note: This graphic is for illustration purposes only and is not part of the required solution.

Exercise 6 [2.1 Data management 2] (15 points)

a) [3 points] Differentiate relational databases from NoSQL databases by naming three different advantages of NoSQL databases.

b) [4 points] Explain the CAP theorem by stating the core statement and explaining each of the three letters.

c) [8 points] Outline NoSQL implementations using four examples. Describe the respective implementations and specify a concrete system in each case.

Proposed Solution:

a) The following three points can be mentioned as examples:

   a. (Better scalability) NoSQL databases can scale higher than comparable relational databases. This can be realized, among other things, by the fact that no high demands are placed on a data schema and the data can therefore be stored quickly.

   b. (Greater flexibility) Various database queries are not supported by relational databases, or only with a great deal of effort.

   c. (Adaptable) Due to lower demands on a database schema, these are not as restrictive in relation to the data model. A relational data structure can severely restrict the data model.

b) The CAP theorem states that in a distributed system it is impossible to guarantee the three properties Consistency, Availability and Partition Tolerance at the same time.

   a. Consistency: The answers from different parts of the system should be consistent.

   b. Availability: All requests to the system are always answered (even if communication within the system is disrupted).

   c. Failure tolerance: The system continues to work even if it is partitioned (i.e. if nodes can no longer communicate with each other).

c) Structure of NoSQL databases (selection):

   a. Key-value store: This refers to generalizations of dictionary/hashmap objects, as they are known from many programming languages. Examples: memcached, Redis

   b. Graph database: Database that uses graphs to display and store highly networked information. Example: Neo4j

   c. Document-oriented database: Extension of the key-value stores. The basic storage unit is a document, which is typically represented by a JSON object. Examples: Elasticsearch, MongoDB.

d. Column-oriented databases: A database management system that physically stores its contents in columns (and not in rows). Examples: Apache Cassandra, Google Big Table.

Exercise 7 [4.2 Deep Learning 2] (20 points)

a) [3 points] Explain the term convolution. Also explain the terms padding and stride.

b) [8 points] Given the following two-dimensional matrix

10	10	10	10	10	10
10	10	10	10	10	10
0	0	0	0	10	10
0	0	0	0	10	10
0	0	0	0	10	10

together with the filter:

-1	-2	-1
0	0	0
1	2	1

Calculate the convolution (use a stride of 1 and no padding). What is the purpose of the specified filter?

c) [5 points] Describe how pooling works. Why is it used? In addition to max pooling, state two other types. Then apply max pooling with a 2x2 filter (f=2) to the result from part b).

d) [4 points] Specify two regularization methods for neural networks and describe them.

Proposed Solution:

a) During the convolution of a matrix, it is multiplied piece by piece with a smaller matrix. In principle, a smaller matrix (e.g. 3x3) is moved step by step over an "image". The stride denotes the increment: with a stride of 1, the shift is made "one pixel" to the right or "one pixel" downwards. Padding is used to add "virtual edge pixels" for calculating the convolution. This specifies how many "pixels" deep this border should be. Typically, zeros are used as content, but other values are also conceivable. This can be used to counteract the following two points:

   a. The result matrix is smaller than the original image

   b. Pixels at the edge are included less in the result matrix

b) The result is the following matrix:

-40	-40	-30	-10
-40	-40	-30	-10
0	0	0	0

This result shows that this is a detector for horizontal edge.

c) Pooling is simply a matter of reducing the size of the input in order to reduce the number of parameters. This increases the generalization in order to achieve greater "tolerance". Analogous to convolution, a filter is moved over the input and a simple operation (e.g. determination of a maximum value) is carried out. Alternatively, the average (Average Pooling) or the L2 norm (L2 Norm Pooling) can be used. For the example from part b) this results in:

-40	-30	-10
0	0	0

d) Three common regularization methods for neural networks are L1, L2 and dropout.

In L1 or L2 regularization, the loss function is extended by an additional term. The L1 or L2 norm of the weight matrices is included in this penalty term. This penalty term is used to keep the respective weights small (or to shrink them to 0). This reduces the complexity of the model.

With dropout, random neurons are switched off (i.e. multiplied by 0) when training the model. When the model is applied, each neuron is multiplied by the corresponding inclusion probability.

Exercise 8 [4.1 Supervised Learning 2] (35 points)

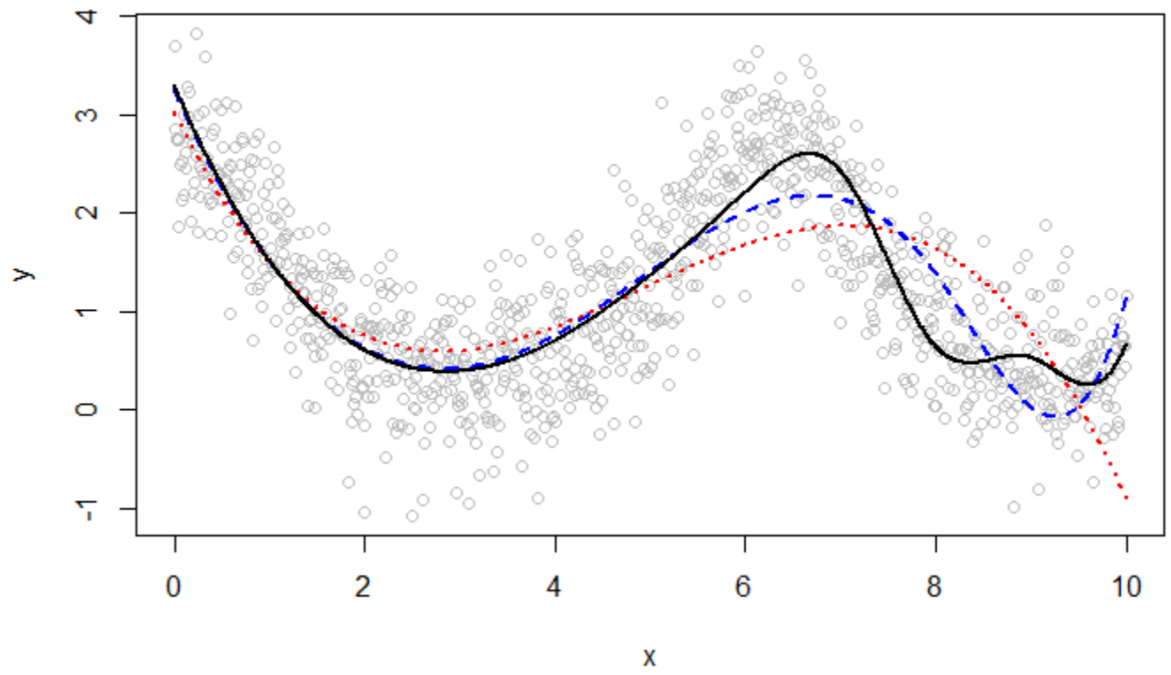
a) [10 points] For your analyses in a data warehouse, you have the data set shown in the image (see below) in front of you. You use two different methods for modeling. Firstly, polynomial modeling:

```
fit_poly <- lm(y ~ poly(x, 3), data=df)
pred_poly <- predict(fit_poly, newdata=list(x=x.grid), se=TRUE)
```

In addition, you use modeling with splines, whereby you use different numbers of nodes (once three and once six nodes). The modeling is carried out according to the following scheme:

```
fitsp1 <-  lm(y~bs(x, knots=c(...)), data=df)
predsp1 <- predict(fitsp1, newdata=list(x=x.grid), se=TRUE)
```

The data points and the results of the modeling are shown in the following figure:



Unfortunately, the assignment of which curve belongs to which model is no longer available. Assign the curves (solid black, dashed blue and dotted red) to a corresponding model with an explanation. Which

splines are used for modeling? Also explain where you (justifiably!) assume the position of the nodes!

b) [7 points] You ask yourself how many nodes should be considered in the modeling in part (a). You have several node sets (labeled K_1, K_2, …, K_n) available. How can you decide? Is considering only the residual sum of squares a good idea (justification)? How can you proceed alternatively?

c) [8 points] When modeling, you consider the following function:

$$s(x) = \begin{cases} (x+1) + (x+1)^3, -1 \leq x < 0 \\ 4 + (x-1) + (x-1)^3, 0 \leq x \leq 1 \end{cases}$$

Which properties of a natural cubic spline with regard to the nodes x_0 = a = -1, x_1 = 0 und x_2 = b = 1 are present or not present? Make a sketch.

d) [10 points] The following curves $\hat{h}_1$ und $\hat{h}_2$ are given:

$$\hat{h}_1 = arg\min_h \left(\sum_{i=1}^n \left(y_i - h(x_i)\right)^2 + \lambda \int \left[h^{(3)}(x)\right]^2 dx \right),$$

$$\hat{h}_2 = arg\min_h \left(\sum_{i=1}^n \left(y_i - h(x_i)\right)^2 + \lambda \int \left[h^{(4)}(x)\right]^2 dx \right)$$

$h^{(m)}$ denotes the m-th differentation of h, and λ is a regularization parameter. Discuss and justify the following points:

a. If $\lambda \to \infty$, is the RSS training value smaller for $\hat{h}_1$ or $\hat{h}_2$?

b. If $\lambda \to \infty$, is the RSS test value smaller for $\hat{h}_1$ or $\hat{h}_2$?

c. If $\lambda = 0$, does $\hat{h}_1$ or $\hat{h}_2$ gains the smaller training- and test RSS?

Proposed Solution:

a) The polynomial fit is a polynomial of degree three. In the graph, only the curve drawn in red (dotted) is similar to a polynomial of degree three. This means that the black (solid) and blue (dashed) curves are spline estimates. R uses cubic polynomials here by default. Both curves have a similar course in the interval from 0 to 6. There are differences in the interval from 6 to 10, where the black curve has a much rougher course, which indicates that there are several nodes in this area that are responsible for this course.

Note: In these simulated data, only the node points 0, 4 and 8 were used for the blue curve. However, 5, 6, 7, 8, 9 and 10 were used for the black curve.

b) The sole consideration of RSS is not a good idea, as it is only based on a measured value. The following procedure would be better: Divide the data set into training and validation parts. Train the model on the training part, apply the model to the validation part. Repeat this procedure and calculate RSS as an average value. This can be done for the different node sets K_1, ..., K_n. The set K with the smallest average RSS is finally selected ("cross-validation").

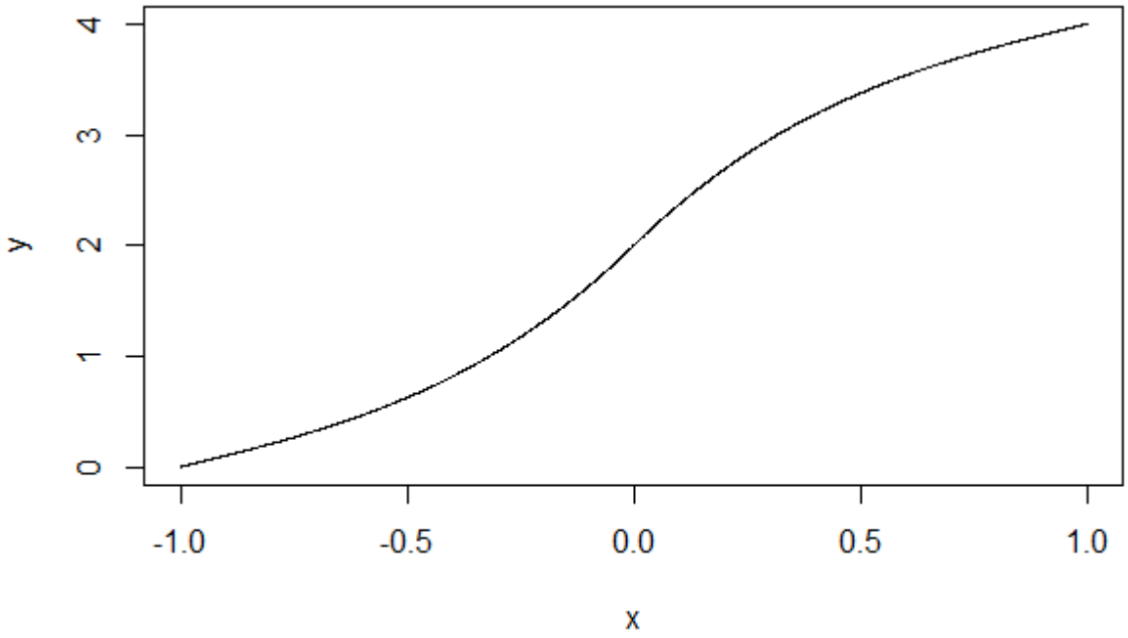
c) $f_1(x) = (x+1) + (x+1)^3$ and $f_2(x) = 4 + (x-1) + (x-1)^3$. With this $f_1'(x) = 1 + 3(x+1)^2, f_1''(x) = 6(x+1)$ and $f_2'(x) = 1 + 3(x-1)^2, f_2''(x) = 6(x-1)$.

Hence:

- $f_1(0) = f_2(0) = 2$
- $f_1'(0) = 4 = f_2'(0)$
- $f_1''(0) = 6 \neq f_2''(0) = -6$
- $f_1''(-1) = 0 = f_2''(1)$

A function of degree 3 exists on the respective intervals. However, a smoothness condition is violated.

Sketch of the function:



d) The points can be answered as follows:

a. Of the two functions considered, the function $\hat{h}_2$ s more flexible, as a higher derivative is used in the staggered term. It will therefore try to favor even smoother functions. Therefore, a lower RSS value can be expected for $\hat{h}_2$ during training.

b. It is difficult to make a general statement. Due to the greater flexibility of the function$\hat{h}_2$ it can be argued that the risk of overfitting is greater, and therefore the smaller test error lies with $\hat{h}_1$.

c. In the case of $\lambda = 0$ the penalty term does not exist, i.e. the training data is interpolated. In this case $\hat{h}_1$ and $\hat{h}_2$ will be the same. This means that the value of RSS in the case of training is 0. On the other hand, a high test RSS value can be expected.