DAV

DEUTSCHE
AKTUARVEREINIGUNG e.V.

Schriftliche Prüfung im Prüfungsfach

# Actuarial Data Science Basic

gemäß der Prüfungsordnung 2.0
der Deutschen Aktuarvereinigung e. V.
zum Erwerb der Zusatzqualifikation CADS

am 29.05.2026

*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.
- Alle Antworten sind zu begründen und bei Rechenaufgaben muss der Lösungsweg ersichtlich sein.
- Bitte vermeiden Sie bei der Lösungserstellung die nicht zusammenhängende Streuung der Lösungen zu den einzelnen Aufgabenteilen.
- Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet.

*Mitglieder der Prüfungskommission:*

Axel Kiermaier, Dr. René Külheim, Prof. Jonas Offtermatt

**Aufgabe 1** [1.1 Grundlagen, 1.2 Digitalisierung, 1.3 Gesellschaftliches Umfeld & Ethik, 1.4 Datenschutz] (28 Punkte)

Ihr Versicherungsunternehmen, ein mittelgroßer deutscher Kompositversicherer, plant die **Einführung von Künstlicher Intelligenz** in mehreren Geschäftsprozessen. Ziel ist es, durch Automatisierung und datengetriebene Entscheidungen effizienter zu arbeiten und neue Produkte schneller auf den Markt zu bringen. Der Vorstand hat eine interne Arbeitsgruppe gegründet, in der auch Sie als Aktuarin bzw. Aktuar mitarbeiten.

Die Geschäftsführung erwartet nun von Ihnen eine fundierte Einschätzung zur **Einführung von KI in der Versicherungswirtschaft** – sowohl hinsichtlich **technologischer Potenziale**, **regulatorischer Anforderungen** als auch **praktischer Umsetzbarkeit**.

a) [5 Punkte] Erläutern Sie allgemeinverständlich (für den Vorstand) die Idee hinter Künstlicher Intelligenz. Geben Sie hierbei auch für die Begriffe *maschinelles Lernen*, *Deep Learning* und *generative KI* eine **kurze** Erklärung.

b) [6 Punkte] Diskutieren Sie, welche Anforderungen die Digitalisierung und der Einsatz von KI an die Anwendungslandschaft und Datenverarbeitung eines Versicherungsunternehmens stellen. Gehen Sie dabei auf die Aspekte Datenqualität, Datensicherheit, Verarbeitungsgeschwindigkeit und IT-Architektur ein.

c) [9 Punkte] Im Rahmen eines Workshops wurden drei konkrete KI-Anwendungen vorgeschlagen:

    1. Das Unternehmen nutzt KI, um automatisch verdächtige Schadenmeldungen zu erkennen und eingehender prüfen zu lassen. Die betroffenen Kunden erfahren nichts von der KI-Prüfung.

    2. Eine neue Vital-App erfasst Gesundheitsdaten von Kunden in Echtzeit und erstellt daraus individuelle Tarifempfehlungen. Durch den Download der App stimmen die Kunden automatisch der Datenverarbeitung zu.

    3. Der Vertriebsinnendienst bekommt eine KI-gestützte Vorhersage darüber, welche Bestandskunden mit hoher Wahrscheinlichkeit kündigen werden, um gezielt Rückgewinnungsaktionen durchzuführen. Dabei werden auch Alter, Wohnort und bisherige Schadenhistorie einbezogen.

Bewerten Sie für jede Anwendung, ob sie aus Sicht der DSGVO verboten, erlaubt oder ein Grenzfall ist. Begründen Sie Ihre Einschätzung jeweils mit 1–2 Sätzen.

d) [8 Punkte] Der Vorstand ist überrascht zu hören, dass KI und Datenschutz zusammenhängen. Erklären Sie ihm **vier** der zentralen Grundprinzipien aus Artikel 5 der DSGVO. (Nennen Sie jeweils das Prinzip und beschreiben Sie es mit einem kurzen Satz.)

**Lösungsvorschlag:**

a) Künstliche Intelligenz (KI) ermöglicht es Computern, Aufgaben zu lösen, die sonst menschliches Denken erfordern – etwa Lernen, Entscheiden oder Erkennen.

**Maschinelles Lernen (ML)** ist ein Teilbereich der KI, bei dem Systeme aus Daten Muster erkennen, statt nach festen Regeln zu arbeiten.
**Deep Learning**, ein Teilbereich des ML, nutzt künstliche neuronale Netze, um besonders komplexe Muster in großen Datenmengen zu erkennen, z. B. bei Bildern oder Sprache.
**Generative KI** kann aus vorhandenen Daten selbst neue Inhalte erzeugen, etwa Texte, Bilder oder Vorschläge.
*(2 Punkte für Erklärung und jeweils 1 Punkt pro erklärtem Begriff.)*

b) **Datenqualität**: KI benötigt saubere, vollständige und strukturierte Daten. Schlechte Datenqualität führt zu fehlerhaften Entscheidungen („Garbage in, garbage out").

**Verarbeitungsgeschwindigkeit**: Oft müssen große Datenmengen effizient verarbeitet werden – dies erfordert leistungsfähige Systeme und moderne Datenbanken.
**IT-Architektur**: Klassische monolithische Systeme stoßen an Grenzen. Moderne Architekturen (z. B. Microservices, Cloud) sind flexibler und unterstützen KI-Anwendungen besser.
**Datensicherheit**: KI benötigt oft personenbezogene Daten – entsprechende Sicherheitsmaßnahmen sind unerlässlich (z. B. Verschlüsselung, Zugriffskontrollen).
*(1,5 Punkte pro Kriterium)*

c) KI erkennt verdächtige Schadenmeldungen – Kunden werden nicht informiert

→ Grenzfall
Begründung: Es liegt eine automatisierte Verarbeitung mit potenziell nachteiligen Folgen vor (Art. 22 DSGVO). Ohne Information und Widerspruchsmöglichkeit fehlt es an Transparenz und Rechtmäßigkeit.

Gesundheits-App erfasst Vitaldaten – Zustimmung per AGB beim Download
→ Verboten
Begründung: Gesundheitsdaten sind besonders schützenswert (Art. 9 DSGVO). Eine wirksame Einwilligung muss ausdrücklich und informiert erfolgen – eine Zustimmung über AGB reicht nicht.

KI prognostiziert Kündigungswahrscheinlichkeit inkl. Alter, Wohnort, Schadenhistorie

→ Erlaubt, unter Bedingungen
Begründung: Die Datenverarbeitung ist zulässig, wenn ein berechtigtes Interesse besteht (Art. 6 Abs. 1 lit. f DSGVO), und Betroffene informiert werden (Art. 13 f.). Profiling ist erlaubt, solange keine automatisierte Entscheidung mit rechtlicher Wirkung erfolgt.
*(3 Punkte pro Einschätzung, die Einschätzung kann auch anders ausfallen, wichtig ist die korrekte Begründung.)*

d) Die Datenschutz-Grundverordnung (DSGVO) nennt in Artikel 5 sieben grundlegende Prinzipien, an denen sich jede Verarbeitung personenbezogener Daten orientieren muss. Vier davon sollten ausgewählt und beschrieben werden. Zur Vollständigkeit sind hier alle sieben Prinzipien mit kurzen Erklärungen aufgeführt:

1. **Rechtmäßigkeit, Verarbeitung nach Treu und Glauben, Transparenz**
   → Daten müssen auf rechtmäßige Weise, nach Treu und Glauben und für die betroffene Person nachvollziehbar verarbeitet werden.
2. **Zweckbindung**
   → Daten dürfen nur für festgelegte, eindeutige und legitime Zwecke erhoben und nicht in einer mit diesen Zwecken unvereinbaren Weise weiterverarbeitet werden.
3. **Datenminimierung**
   → Es dürfen nur so viele personenbezogene Daten verarbeitet werden, wie für den jeweiligen Zweck tatsächlich notwendig sind.
4. **Richtigkeit**
   → Es muss sichergestellt sein, dass personenbezogene Daten sachlich richtig und auf dem neuesten Stand sind; unrichtige Daten müssen unverzüglich berichtigt oder gelöscht werden.
5. **Speicherbegrenzung**
   → Daten dürfen nur so lange gespeichert werden, wie es für die Zwecke der Verarbeitung erforderlich ist. Danach müssen sie gelöscht oder anonymisiert werden.
6. **Integrität und Vertraulichkeit** (auch: Sicherheit der Verarbeitung)
   → Die Daten müssen durch geeignete technische und organisatorische Maßnahmen vor unbefugtem Zugriff, Verlust oder Zerstörung geschützt werden.
7. **Rechenschaftspflicht**
   → Das Unternehmen muss jederzeit nachweisen können, dass es die oben genannten Prinzipien einhält („Accountability").

*(2 Punkte pro Prinzip mit Erläuterung, max. 8 Punkte)*

**Aufgabe 2** [3.1 Data Mining, 4.1 Überwachtes Lernen, 5.3 Programmiersprachen für Data Science, 6.1 Use Case] (32 Punkte)

Ihr Unternehmen plant, einen KI-basierten Entscheidungsbaum zur Einschätzung der Abschlusswahrscheinlichkeit von Neukunden zu entwickeln. Dafür wurde eine kleine Trainingsstichprobe von Hand erfasst:

| ID | Age | Occupation | Previous_Policy | Income | Purchase |
|---|---|---|---|---|---|
| **1** | 25 | Employee | Yes | 3,000 | Yes |
| **2** | 45 | Self-Employed | No | 6,500 | No |
| **3** | 30 | Student | No | 1,200 | Yes |
| **4** | 50 | Employee | Yes | 4,800 | Yes |
| **5** | 23 | Student | No | 1,000 | No |
| **6** | 35 | Self-Employed | Yes | 5,800 | Yes |
| **7** | 29 | Employee | No | 3,200 | No |
| **8** | 40 | Self-Employed | No | 5,000 | No |
| **9** | 33 | Student | Yes | 2,100 | Yes |
| **10** | 55 | Employee | Yes | 4,500 | Yes |

a) [5 Punkte] Erstellen Sie eine geeignete Visualisierung, um den Zusammenhang zwischen einer beliebigen Eingangsvariablen (z. B. Beruf=*Occupation*, Vorvertrag=*Previous_Policy*, Alter=*Age*, Einkommen=*Income*) und der Zielvariable Abschluss (=*Purchase*) darzustellen. (*Bitte achten Sie auf die grundlegenden Anforderungen an eine Visualisierung: Verwendung eines Lineals, Achsenbeschriftung, etc. Hingekritzelte Skizzen werden mit 0 Punkten bewertet.)*

b) [6 Punkte] Bereiten Sie die folgenden Variablen für den Einsatz in einem Entscheidungsbaum-Modell vor: `Occupation`, `Previous_Policy`, `Purchase`. Verwenden Sie hierfür eine passende numerische Kodierung für jede der Variablen.

c) [5 Punkte] Erstellen Sie einen einfachen Entscheidungsbaum (minimal 2 Ebenen/Wurzeln, maximal 3), der aus den Daten ableitet, ob ein Abschluss erfolgt oder nicht. Verwenden Sie als erstes Splitkriterium: „`Previ-`

ous_Policy**"**. Begründen Sie Ihre Entscheidung für weitere Splits mit der Häufigkeit der Zielvariablen.

d) [8 Punkte] Gegeben sei folgender Entscheidungsbaum:

```
Income > 4800 ?
├── Yes  → predict Purchase = No
└── No   → split by Previous_Policy?
       ├── Previous_Policy = Yes  → predict Purchase = Yes
       └── Previous_Policy = No   → predict Purchase = No
```

Wenden Sie diesen Baum auf die 10 Trainingsfälle an und erstellen Sie die entstehende Konfusionsmatrix. Berechnen Sie zusätzlich die Accuracy (Anteil korrekt klassifizierter Fälle) und nennen Sie die IDs der falsch klassifizierten Fälle. Erläutern Sie kurz (jeweils ein Satz), woran diese Fehlklassifikation liegen könnte.

e) [4 Punkte] Im Projekt wurde folgender R-Code verwendet:

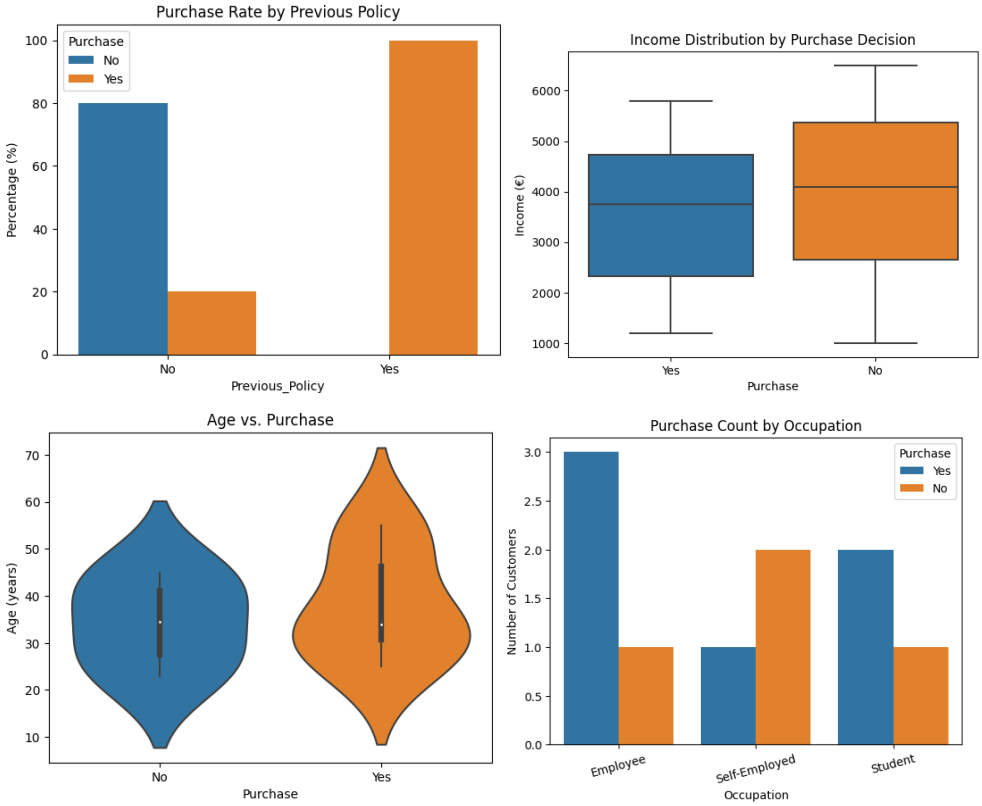
```
library(rpart)

model <- rpart(Purchase ~ Age + Income + Occupation + Previ-
ous_Policy,
               data = df,
               method = "class",
               control = rpart.control(minsplit = 20, cp = 0.01))
```

Welche Art von Modell wird hier trainiert? Was bedeuten die Parameter `method, minsplit, cp`?

f) [4 Punkte] Was versteht man unter *Bagging* (Bootstrap Aggregating) im Zusammenhang mit Entscheidungsbäumen? Erläutern Sie kurz, wie Bagging funktioniert und welchen Vorteil es gegenüber einem einzelnen Entscheidungsbaum bietet.

**Lösungsvorschlag:**

a) Mögliche Visualisierungen sind beispielsweise:




*(Die Diagrammart ist nicht vorgegeben, Abzüge gibt es für unsauber ausgeführte Zeichnungen / Skizzen.)*

b) Da es sich um kategoriale Variablen handelt, wird jeder Kategorie eine Zahl zugeordnet, z.B.:

| **Variable** | **Mögliche Ausprägungen** | **Kodierung** |
|---|---|---|
| Occupation | Employee, Student, Self-Employed | Employee = 0, Student = 1, Self-Employed = 2 |
| Previous_Policy | Yes / No | Yes = 1, No = 0 |
| Purchase | Yes / No | Yes = 1, No = 0 |

c) Erstellen des Entscheidungsbaums:

Erster Split nach „`Previous_Policy`”:

```
|Previous_Policy = Yes| ID: 1, 4, 6, 9, 10| Alle → Purchase = Yes|
|Previous_Policy = No | ID: 2, 3, 5, 7, 8 | Nur ID 3 → Yes |
```

„`Previous_Policy`” ist also ein ganz gutes Splitkriterium, nur eine ID wird falsch einsortiert.

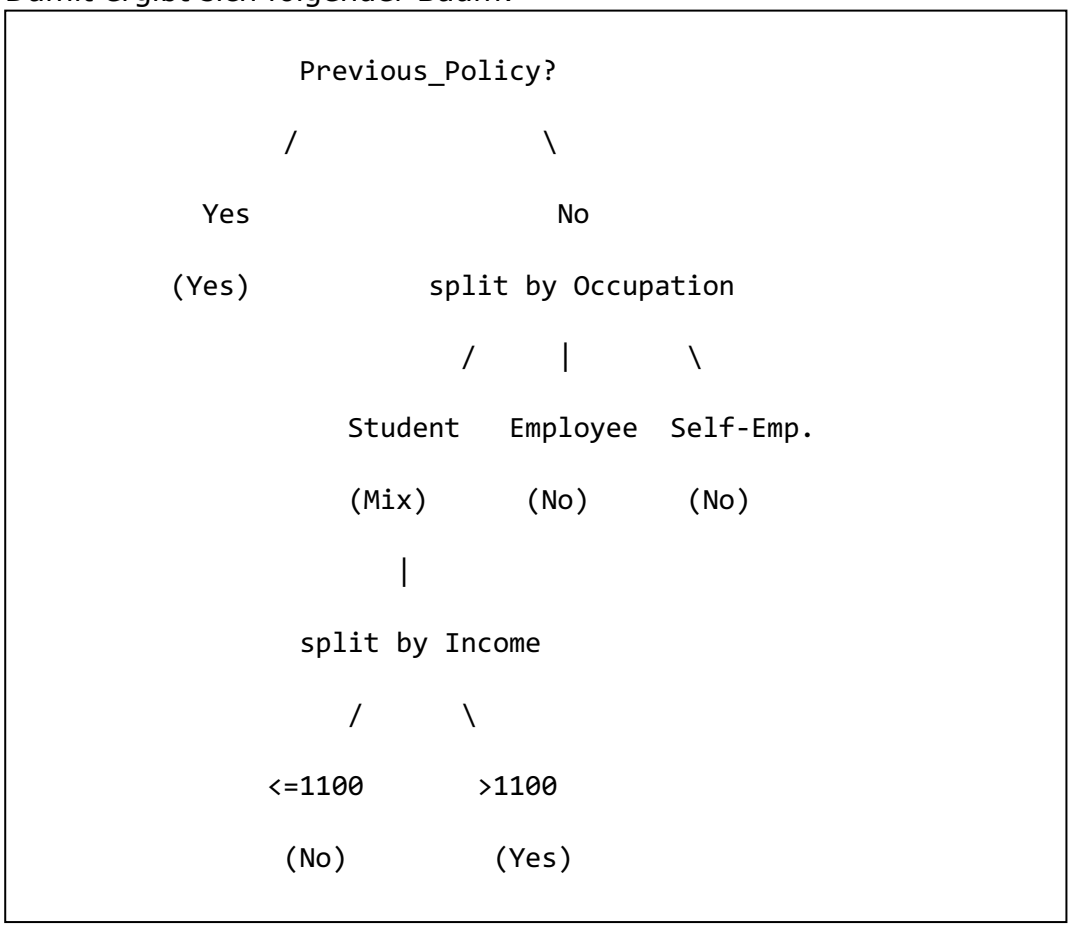
Für ein weiteres Splitkriterium hat man nun die Wahl zwischen `Age`, `Income`, `Occupation`. Die Schaubilder in Aufgabenteil a) zeigen, dass alle drei keine guten Entscheidungen liefern, wir wählen deshalb `Occupation`.

```
|Occupation = Student                    | ID: 3, 5    | Yes, No |
|Occupation = Self-Employed Or Employee | ID: 2, 7, 8 | Alle No |
```

Wie nach dem ersten Split, haben wir weiterhin eine falsch einsortierte ID. Als dritte Ebene verwenden wir Income:

```
|Income <= 1100  | ID: 5    | No  |
|Income > 1100   | ID: 3    | Yes |
```

Damit ergibt sich folgender Baum:




d) Wendet man den Baum auf die Trainingsfälle an, so ergibt sich folgendes Ergebnis:

| ID | Income | Previous_Policy | Actual Purchase | Predicted Purchase | Correct |
|---|---|---|---|---|---|
| **1** | 3000 | Yes | Yes | Yes | Correct |
| **2** | 6500 | No | No | No | Correct |
| **3** | 1200 | No | Yes | No | Incorrect |
| **4** | 4800 | Yes | Yes | Yes | Correct |

| | | | | | |
|---|---|---|---|---|---|
| **5** | 1000 | No | No | No | Correct |
| **6** | 5800 | Yes | Yes | No | Incorrect |
| **7** | 3200 | No | No | No | Correct |
| **8** | 5000 | No | No | No | Correct |
| **9** | 2100 | Yes | Yes | Yes | Correct |
| **10** | 4500 | Yes | Yes | Yes | Correct |

Die Konfusionsmatrix ergibt sich damit zu:

| **actual \ predicted** | **pred Yes** | **pred No** |
|---|---|---|
| **actual Yes** | 4 | 2 |
| **actual No** | 0 | 4 |

Und die Accuracy zu:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{4 + 4}{4 + 4 + 0 + 2} = \frac{8}{10} = 80\%$$

IDs 3 und 6 sind falsch klassifiziert.

ID 3: Income = 1200 (≤4800) und Previous_Policy = No → Baum sagt No, tatsächlich Yes. Hier hat der Kunde trotz fehlendem Vorvertrag und niedrigem Einkommen gekauft — ein Ausreißer, den die Regel nicht erfasst.

ID 6: Income = 5800 (>4800) → Baum sagt No, tatsächlich Yes. Dieser Kunde hat hohes Einkommen, aber kaufte trotzdem; der einfache Income-Cut führte hier zu einer Fehlvorhersage.

*(4 Punkte für korrekte Konfusionsmatrix, 2 Punkte für korrekte Accuracy, 1 Punkt pro erläuterter falsch klassifizierter ID)*

e) Art des Modells:

Ein Entscheidungsbaum-Klassifikationsmodell (rpart), das die Zielvariable `Purchase` basierend auf den Prädiktoren `Age, Income, Occupation` und `Previous_Policy` vorhersagt.

Wichtigste Parameter:

- method = "class": Modell für Klassifikationsaufgaben (kategoriale Zielvariable).
- control = rpart.control(minsplit = 20, cp = 0.01)
  - minsplit = 20: Ein Knoten wird nur geteilt, wenn mindestens 20 Beobachtungen darin sind.

- cp = 0.01: Komplexitätsparameter zur Vermeidung von Überanpassung und unnötiger Rechenzeit; steuert die Baumgröße durch Pruning.

*(1 Punkt für Art des Modells, jeweils ein Punkt pro Parameter)*

f) Bagging (Bootstrap Aggregating) ist ein Verfahren, bei dem **viele Entscheidungsbäume** auf **unterschiedlichen, zufällig mit Zurücklegen gezogenen Stichproben (Bootstrap-Samples)** aus den Trainingsdaten trainiert werden. Bei Random Forests wird zusätzlich bei jedem Split nur eine zufällige Teilmenge der Merkmale betrachtet, damit die Bäume möglichst unterschiedlich sind. Die finale Vorhersage entsteht durch **Mehrheitsentscheidung (Klassifikation) oder Durchschnitt (Regression)** aller Bäume. Durch diese Kombination wird die **Varianz des Modells reduziert** und **die Genauigkeit verbessert**, da Überanpassung (Overfitting) einzelner Bäume vermindert wird.

*(Für die volle Punktzahl sind die fett markierten Stichpunkte zu nennen.)*

**Aufgabe 3** [4.2 Unüberwachtes Lernen 1] (34 Punkte)

Sie präsentieren Ihrem Vorstand einige Ergebnisse Ihrer neuen Clusterstudie und beantworten einige seiner Verständnisfragen.

a) [12 Punkte] Nennen Sie drei möglichst verschiedene Clusterverfahren und erläutern Sie in jeweils zwei Sätzen den Ansatz der drei Verfahren (Ihr Vorstand hat nicht viel Zeit!). Dabei geht es um eine klare Darstellung des Grundprinzips, nicht der konkreten Definitionen und Algorithmen. Nennen Sie für jedes der drei Verfahren jeweils zwei wesentliche Vor- und zwei Nachteile.

b) [5 Punkte] Skizzieren Sie die einzelnen Schritte des k-Means-Algorithmus im Pseudo-Code. Nutzen Sie dabei nicht – wie sonst üblich – die quadratische euklidische Distanz bzw. die L2-Metrik, sondern die Manhattan-Distanz bzw. L1-Metrik.

   *Hinweise:*

   1. Die Manhattan-Distanz zwischen zwei Punkten ist als die Summe der absoluten Differenzen ihrer Einzelkoordinaten definiert.

   2. Bei der Manhattan-Distanz ist der Median der optimale Repräsentant eines Clusters (nicht der Mittelwert).

c) [5 Punkte] Wie sollte im k-Means-Algorithmus generell mit den Themen

   - Ausreißer und

   - unterschiedlich skalierte Merkmale

   umgegangen werden und warum? Wie wirkt sich die Wahl der Metrik (L2 vs. L1) auf diese beiden Themen aus? Begründen Sie kurz Ihre jeweilige Antwort.

d) [12 Punkte] Gegeben ist der folgende einfache Datensatz mit den zwei standardisierten Merkmalen Feature_1 und Feature_2:

| Item | Feature_1 | Feature_2 |
|---|---|---|
| 1 | -0,77 | -0,05 |
| 2 | -0,42 | -0,85 |
| 3 | 1,76 | 1,86 |
| 4 | -0,24 | -0,25 |
| 5 | 1,29 | 1,03 |
| 6 | -0,54 | -0,57 |
| 7 | -1,07 | -1,17 |

Führen Sie die erste Iteration einer k-Means-Clusterung basierend auf der Manhattan-Distanz / L1-Metrik durch.

Verwenden Sie dabei die Items 1, 2 und 3 als Startcentroiden. Welche Position haben nach der ersten Iteration die aktualisierten Centroiden? Ihre Berechnungsschritte sollten zumindest beispielhaft nachvollziehbar sein.

Bei einer k-Means-Clusterung derselben Daten basierend auf der euklidischen Distanz / L2-Metrik ergeben sich nach der ersten Iteration die folgenden Clusterzugehörigkeiten:

| Item | Cluster |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 1 |
| 5 | 3 |
| 6 | 2 |
| 7 | 2 |

Bestimmen Sie die aktualisierten Centroiden bei Verwendung der L2-Metrik und interpretieren Sie das Ergebnis.

**Lösungsvorschlag:**

a) Drei verschiedene Typen von Clusterverfahren:

- Centroidbasierte Verfahren (z.B. k-Means-Clustering)
- Konnektivitätsbasierte Verfahren (z.B. Hierarchisches agglomeratives Clustering)
- Dichtebasierte Verfahren (z.B. DBSCAN)

*(Hier genügt jeweils die Nennung eines Beispiels; es müssen aber für die volle Punktzahl alle drei Typen von Clusterverfahren repräsentiert sein. Im Folgenden werden z. T. mehr als zwei Vor- bzw. Nachteile genannt; für die volle Punktzahl sind jeweils zwei Vor- und zwei Nachteile ausreichend.)*

**k-Means-Clustering:**

*Grundprinzip:*

Ausgangspunkt sind eine Auswahl von k Beobachtungen (Objekte) als Start-Clusterzentren (sog. Centroiden), denen sämtliche Objekte so zugeordnet werden, dass die Objekte innerhalb der Cluster möglichst ähnlich sind. Im Anschluss wird für jedes Cluster das neue Zentrum als Mittelpunkt/Schwerpunkt der enthaltenen Objekte bestimmt und diesen neuen Clusterzentren werden erneut sämtliche Objekte zugeordnet.

*Vorteile:*

Leicht zu verstehen, zu erklären und zu implementieren.

Effizient bei hochdimensionalen Daten, findet schnell die Zentren der Gruppen.

Schnelle Anpassung an neue Beobachtungen.

*Nachteile:*

Anzahl der Cluster ist vorzugeben (k ist Hyperparameter).

Cluster stark abhängig von initialer Wahl der Clusterzentren.

Führt je nach Abstandsmaß zu bestimmt geformten Clustern (z.B. bei euklidischer Norm zu kreis-/kugelförmigen Clustern); generell

sind es im Sinne der gewählten Norm konvexe Formen – andere Formen werden nicht erkannt.

Schwache Performance bei Regionen unterschiedlicher Dichte der Beobachtungen.

k-Means ist empfindlich gegenüber Ausreißern.

**Hierarchisches Clustering (agglomerativ):**

*Grundprinzip:*

Ausgehend von den einzelnen Punkten (Beobachtungen) wird anhand von Ähnlichkeiten eine Baumstruktur entwickelt. Je nachdem, auf welcher Ebene man den „Schnitt" macht, erhält man mehr oder weniger Cluster.

*Vorteile:*

- Keine Vorab-Festlegung der Clusteranzahl nötig.
- Reproduzierbarkeit der Ergebnisse (z.B. unabhängig von einer initialen Wahl von Centroiden).
- Gute Erklärbarkeit aufgrund der Baumstruktur der Ergebnisse.

*Nachteile:*

- Hoher Zeit- und Speicherbedarf gegenüber k-Means, auch bei mittelgroßen Datensätzen.
- Hohe Empfindlichkeit gegenüber Ausreißern.
- Interpretation des Dendrogramms kann subjektiv sein.

**DBSCAN:**

*Grundprinzip:*

Alle Punkte (Objekte, Beobachtungen) werden eingeteilt in Kernobjekte, dichte-erreichbare Objekte und Rauschobjekte, wobei diese Einteilung auf ε-Nachbarschaften und Mindestanzahlen von Punkten in der ε-Nachbarschaft beruht. Alle durch dieselben Kernobjekte verbundenen Objekte bilden ein Cluster und die Rauschobjekte stellen Ausreißer dar.

*Vorteile:*

- Keine Vorab-Festlegung der Clusteranzahl nötig.
- Nur wenige Parameter erforderlich.
- Kommt gut mit Ausreißern zurecht.
- Unregelmäßig geformte Cluster können erkannt werden.

*Nachteile:*

- Nichtdeterministische Zuordnung von Nicht-Kernobjekten, also solchen, die nicht dicht sind.
- Laufzeitprobleme bei hochdimensionalen Daten.
- Optimale Festlegung von Nachbarschaftsradius und Mindestnachbarschaftspunktzahl kann herausfordernd sein, insbesondere wenn Gebiete mit sehr unterschiedlichen Dichten vorliegen.

b) Ein möglicher Pseudocode für k-Means mit Manhattan-Distanz sieht so aus:

Eingabe:

Beobachtungen $x_1, \ldots, x_n$

Anzahl der Cluster k

1. Wähle k Clustercentroiden $\mu_1, \ldots, \mu_k$ zufällig aus $\{x_1, \ldots, x_n\}$.
2. Wiederhole bis es keine oder nur minimale Änderungen gibt:

   a) Zuweisungsschritt:

   Für jeden Punkt $x_i$:

   Weise $x_i$ dem Cluster $C_j$ zu, dessen Centroid $\mu_j$ den kleinsten Manhattan-Abstand zu $x_i$ hat; dieser berechnet sich als

   $$d(x_i, \mu_j) = \Sigma\ |x_{im} - \mu_{jm}| \text{ über alle Merkmale } m$$

   b) Aktualisierungsschritt:

```
        Für jeden Cluster C_j:

            Definiere den neuen Centroiden μ_j
            durch den Median jedes Merkmals der Punkte in C_j.

    Rückgabe:

        Cluster C_1, …, C_k

        Centroide μ_1, …, μ_k
```

c) Generell bestimmt im Zuweisungsschritt von k-Means die Metrik, wie stark große Abweichungen das Clustering beeinflussen. Bei der L2-Metrik (euklidische Distanz) gehen große Abweichungen quadratisch (also stärker) ein, bei der L1-Metrik (Manhattan-Distanz) dagegen nur linear.

**Ausreißer:**

Ausreißer sollten generell vor der eigentlichen Clusteranalyse (temporär) aus dem Datensatz entfernt werden, da Ausreißer die Lage der Clusterzentren verzerren bzw. anderenfalls jeder Ausreißer ein eigenes Cluster beansprucht und somit das notwendige k in die Höhe getrieben wird.

Mit der L1-Metrik (Manhattan-Distanz) wirken sich große Abweichungen und damit Ausreißer weniger stark auf die Clusterzentren aus als mit der L2-Metrik. Außerdem sind Mediane robuster als Mittelwerte (Aktualisierungsschritt). Insgesamt verzerren damit Ausreißer bei L1 weniger das Ergebnis von k-Means als bei L2.

**Unterschiedlich skalierte Merkmale:**

Sowohl bei L1 als auch L2 führen unterschiedlich skalierte Merkmale zu verzerrten Distanzen; daher ist grundsätzlich in beiden Fällen eine Normierung oder Standardisierung der Daten erforderlich.

Die Notwendigkeit zur Standardisierung ist bei der L2-Metrik größer, da sie große Abweichungen quadratisch betont und damit stark von unterschiedlich skalierten Merkmalen verzerrt wird, während die L1-Metrik Abweichungen linear berücksichtigt und dadurch etwas robuster gegenüber Maßstabsunterschieden ist.

d) Zunächst wird für jedes Item dessen **Distanz zu den Items 1, 2 und 3** berechnet. Die kleinste dieser drei Distanzen zeigt dann an, zu welchem Item / Cluster das jeweilige Item zuzuordnen ist.

| Item | F_1 | F_2 | Dist. zu 1 | Dist. zu 2 | Dist. zu 3 | Cluster |
|---|---|---|---|---|---|---|
| 1 | -0,77 | -0,05 | 0,00 | 1,15 | 4,44 | 1 |
| 2 | -0,42 | -0,85 | 1,15 | 0,00 | 4,89 | 2 |
| 3 | 1,76 | 1,86 | 4,44 | 4,89 | 0,00 | 3 |
| 4 | -0,24 | -0,25 | 0,73 | 0,78 | 4,11 | 1 |
| 5 | 1,29 | 1,03 | 3,14 | 3,59 | 1,30 | 3 |
| 6 | -0,54 | -0,57 | 0,75 | 0,40 | 4,73 | 2 |
| 7 | -1,07 | -1,17 | 1,42 | 0,97 | 5,86 | 2 |

Beispiel: Berechnung der Distanz von Item 5 zu Item 1:

$$|1{,}29 - (-0{,}77)| + |1{,}03 - (-0{,}05)| = 2{,}06 + 1{,}08 = 3{,}14.$$

Bemerkung: Für die Items 1 bis 3 muss die gegenseitige Distanz nicht ermittelt werden, um die Clusterzugehörigkeit zu bestimmen, da per definitionem Item 1 zu Cluster 1 etc. gehört. Daher sind die Werte in der obigen Tabelle ausgegraut.

Die **aktualisierten Centroiden** ergeben sich über den Median der Feature-Werte im jeweiligen Cluster wie folgt (auf zwei Nachkommastellen gerundet):

| Cluster | Centroid |
|---|---|
| 1 | (-0,51 ; -0,15) |
| 2 | (-0,54 ; -0,85) |
| 3 | (1,53 ; 1,45) |

Beispiel: Der Wert von Feature_1 für den neuen Centroiden von Cluster 1 ergibt sich als Median der Werte -0,77 und -0,24; also ((-0,77) + (-0,24)) / 2 = -0,505. Der Wert von Feature_1 für den neuen Centroiden von Cluster 2 ergibt sich als Median der Werte -0,42 und -0,54 und -1,07; also -0,54.

Bei Verwendung der **L2-Metrik** ergeben sich die aktualisierten Centroiden über den Mittelwert der Feature-Werte im jeweiligen Cluster wie folgt:

| Cluster | Centroid |
|---|---|
| 1 | (-0,51 ; -0,15) |
| 2 | (-0,68 ; -0,86) |
| 3 | (1,53 ; 1,45) |

Beispiel: Der Wert von Feature_1 für den neuen Centroiden von Cluster 1 ergibt sich als Mittelwert der Werte -0,77 und -0,24; also ((-0,77) + (-0,24)) / 2 = -0,505.

**Beobachtung / Interpretation:**

Die Clusterzugehörigkeiten nach der ersten Iteration sind bei L1- und L2-Metrik identisch.

Die Centroiden der Cluster 1 und 3 sind identisch, da hier – insbesondere wegen des sehr kleinen Stichprobenumfangs und der geraden Anzahl der in diesen Clustern enthaltenen Items – Mittelwert und Median zusammenfallen.

Der Centroid von Cluster 2 unterscheidet sich etwas, da der bei L2 verwendete Mittelwert bei Feature_1 etwas stärker zum Ausreißer (Item 7) hingezogen wird als der bei L1 verwendete der Median. Da es sich aber nicht um einen extremen Ausreißer handelt und die Daten insgesamt relativ nahe beieinander liegen, ist der Effekt relativ gering.

**Aufgabe 4** [1.1 Grundlagen, 3.1 Data Mining 1, 2.1 Datenmanagement 1] (26 Punkte)

a) [7 Punkte] Erläutern Sie die Unterschiede von strukturierten, unstrukturierten und semistrukturierten Daten (wobei Sie jeweils ein einfaches Beispiel nennen) sowie die Bedeutung dieser Unterscheidung im Kontext von Data Science. Zählen Sie außerdem mindestens drei möglichst verschiedene Beispiele von Datenquellen im aktuariellen Umfeld auf.

b) [7 Punkte] Gegeben ist folgender Beispiel-Datensatz mit den Merkmalen Angebotsnummer, Name, Versicherungsbeginn, Eintrittsalter, Geschlecht, Versicherungssumme und Vorversicherung (j/n):

| QuotationNumber | Name | InceptionDate | EntryAge | Sex | SumAssured | PreviousCover |
|---|---|---|---|---|---|---|
| 001200345 | Peter Meier | 01.12.2024 | 23 | M | 150000 | TRUE |
| 001200856 | Petra Huber | 01.01.2025 | 63 | F | 35000 | FALSE |

Geben Sie für jedes der sieben Merkmale die Merkmalsart (qualitativ etc.), das Skalenniveau (Ordinalskala etc.) und einen sinnvollen Datentyp (Character etc.) an.

c) [6 Punkte] Zum Verarbeiten des obigen Beispieldatensatzes unter maximaler Nutzung der gegebenen Informationen:

Wie könnte dieser Beispiel-Datensatz als csv-Datei aussehen? Wie als JSON-Datei?

d) [6 Punkte] Mit welcher Anweisung können Sie die csv-Datei test.csv mit R einlesen? Nennen Sie fünf Elemente, die zusätzlich zum Dateinamen zu spezifizieren sind – auch wenn sie häufig durch Defaultwerte belegt werden. Gesucht sind nicht die Parameterbezeichner für den Funktionsaufruf, aber eine knappe inhaltliche Beschreibung.

**Lösungsvorschlag:**

a) **Strukturierte Daten:**

… haben ein vorgegebenes Format (eine normalisierte Form), in das alle Informationen eingeordnet sind.

Beispiel: relationale Datenbank mit definierter Zeilen- und Spaltenposition für alle Daten.

**Unstrukturierte Daten:**

… liegen nicht in einer klar nachvollziehbaren Struktur vor.

Beispiele: Texte, Bilder, Audio- und Videodateien.

**Semistrukturierte Daten:**

… sind nicht vollständig strukturiert und nicht vollständig unstrukturiert.

Beispiel: Der Nachrichtenkopf einer E-Mail hat eine klare Struktur (Absender, Empfänger, etc.). Der eigentliche Inhalt ist in der Regel ein strukturloser Text.

**Datenquellen im aktuariellen Umfeld:**

Hierzu gehören z.B.

- Informationen aus den Bestandsführungssystemen, DWH-Systemen, …
- Informationen aus Wearables, Daten aus der Nutzung von Apps, Homepage, …
- Kundendaten (neben den klassischen Bestandsdaten auch aus der Korrespondenz mit dem Kunden), z.B. Infos über Familienstand und Familienmitglieder, …
- Wetterdaten
- Unfalldaten, Daten zu Einbrüchen, …
- Daten aus Unfallmeldesystemen beim Auto

*(Die Nennung von drei verschiedenen Quellen genügt hier.)*

**Bedeutung der Unterscheidung:**

Strukturierte Daten lassen sich relativ einfach z.B. mit SQL analysieren. Viele heute in den Unternehmen anfallenden bzw. verwendeten Daten sind unstrukturiert (siehe obige Aufstellung der Datenquellen). Je weniger strukturiert die Daten vorliegen, umso aufwändiger ist deren Verarbeitung und Analyse im Rahmen von Data-Science-Aufgaben.

b) Die Zuordnungen für die einzelnen Merkmale zeigt die folgende Tabelle:

| | Merkmalsart | Skalenniveau | Datentyp |
|---|---|---|---|
| QuotationNumber | qualitativ | Nominalskala | Character |
| Name | qualitativ | Nominalskala | Character |
| InceptionDate | quantitativ | Intervallskala | Date |
| EntryAge | quantitativ | Verhältnisskala | Integer |
| Sex | qualitativ | Nominalskala | Character |
| SumAssured | quantitativ | Verhältnisskala | Integer/Float |
| PreviousCover | qualitativ | Nominalskala | Logical |

c) Der Beispiel-Datensatz im csv-Format:

```
QuotationNumber;Name;InceptionDate;EntryAge;Sex;SumAssured;PreviousCover
001200345;Peter Meier;01.12.2024;23;M;150000;TRUE
001200856;Petra Huber;01.01.2025;63;F;35000;FALSE
```

Und das Gleiche im JSON-Format:

```
[
  {
    "QuotationNumber": "001200345",
    "Name": "Peter Meier",
    "InceptionDate": "01.12.2024",
    "EntryAge": 23,
    "Sex": "M",
    "SumAssured": 150000,
    "PreviousCover": true
  },
  {
    "QuotationNumber": "001200856",
    "Name": "Petra Huber",
    "InceptionDate": "01.01.2025",
    "EntryAge": 63,
    "Sex": "F",
    "SumAssured": 35000,
    "PreviousCover": false
  }
]
```

*Hinweis:* Klammern, Kommas und Anführungszeichen in der JSON-Darstellung sind wesentlich für die Korrektheit der Lösung. Fehlen diese, gibt es einen entsprechenden Punktabzug. Die Einzüge hingegen sind technisch nicht zwingend erforderlich.

d) Die Anweisung zum Einlesen von test.csv in R lautet:

```
read.csv("test.csv")
```

Zusätzlich zum Dateinamen kann man folgende Informationen spezifizieren:

• Trennungszeichen (in der Lösung zu Teilaufgabe c ";", aber auch TAB oder im Englischen "," sind gebräuchlich)

• Überschrift (y/n)

• Zeilenumbruch – in unixorientierten Betriebssystemen normalerweise "\n" und unter Windows "\n\r"

• quote character

• escape character

• Dezimaltrennzeichen und ggf. Tausender-Trennzeichen für Zahlen

• Enkodierung

*Hinweis:* Die Nennung von fünf dieser Informationen genügt für die volle Punktzahl.

**Aufgabe 5** [3.1 Analytics 1] (15 Punkte)

a) [5 Punkte] Erklären Sie, was unter dem Bias-Varianz-Kompromiss (bias-variance tradeoff) zu verstehen ist. Die (lokale) Formel hierfür ist unter der Annahme $y = f(x) + \epsilon$ (N(0,1) verteiltes Rauschen) explizit anzugeben und die einzelnen Bestandteile zu erläutern.

b) [4 Punkte] Visualisieren Sie die möglichen Kombinationen aus hohe/r / niedrige/r Bias/Varianz in Bullaugen-Diagrammen (siehe Anlage). Erläutern Sie die entstandenen Diagramme.

c) [6 Punkte] Fertigen Sie eine Skizze an, in der auf der y-Achse der (Modell) Fehler abgetragen wird, und auf der x-Achse die Modellkomplexität. Erläutern Sie, wie die typischen Kurven von totalem Fehler, Varianz und Bias verlaufen. Wo würden Sie eine optimale Modellkomplexität feststellen?

**Lösungsvorschlag:**

a) Bei Problemstellungen des maschinellen Lernens beschreibt der Bias-Varianz-Kompromiss das Problem, gleichzeitig zwei Fehlerquellen zu minimieren, die bewirken, dass überwachte Lernalgorithmen nicht über ihre Trainingsdaten hinaus verallgemeinern können. Beide Bestandteile finden sich in der folgenden Formel (lokale Betrachtung für ein $x$):

$$E(y - f(x))^2 = Var\left(\hat{f}(x)\right) + Bias\left(\hat{f}(x)\right)^2 + Var(\varepsilon)$$

Dabei ist $y = f(x) + \varepsilon$. Zudem bezeichnet $\hat{f}(x)$ eine Funktion, die eine wahre Funktion $f(x)$ möglichst gut approximiert (diverse Verfahren zur Ermittlung von $\hat{f}$ sind im Kurs vorgestellt worden).

Der Bias misst dabei, wie stark die Vorhersagen des Modells systematisch vom tatsächlichen Zielwert abweichen.

Die Varianz misst, wie stark die Modellvorhersagen schwanken, wenn man mit leicht unterschiedlichen Trainingsdaten trainiert.

b) Bei dem Bullaugen-Diagramm ist die Interpretation wie folgt: Der Punkt im Zentrum entspricht einem Modell, welches die Daten perfekt vorhersagt. Je weiter sich von diesem Punkt entfernt wird, desto schlechter werden die Vorhersagewerte. Jeder im Folgenden dargestellte schwarze Punkt entspricht der Realisation eines Modells (d.h. der Prozess der Modellbildung wurde für unterschiedliche Trainingsdatensätze durchgeführt). Dann lassen sich folgende Situationen unterscheiden:

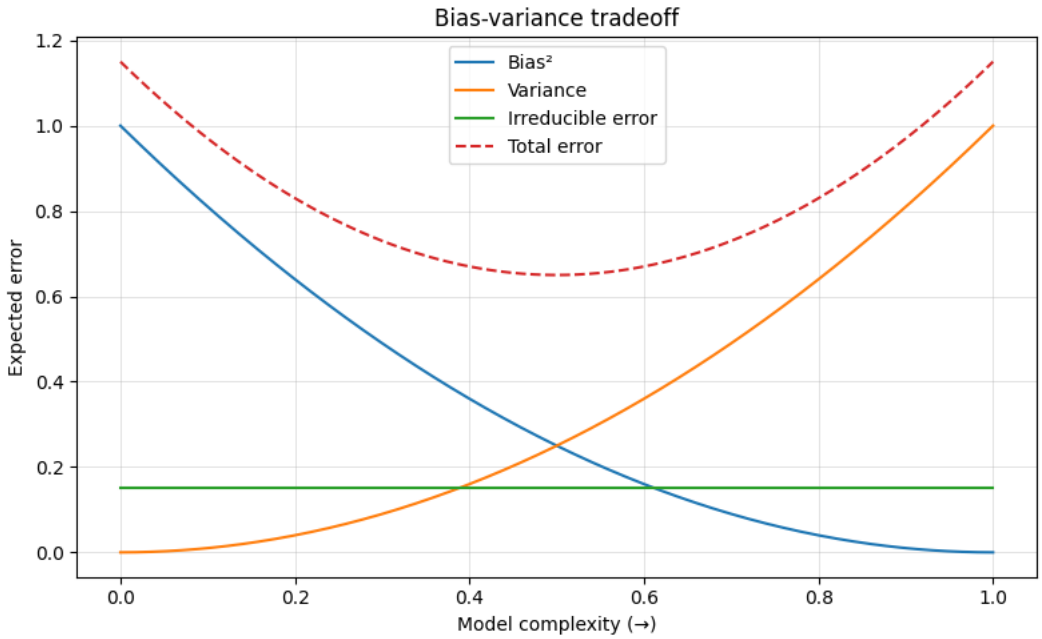
Niedriger Bias, niedrige Varianz:

Niedriger Bias, hohe Varianz:

Hoher Bias, niedrige Varianz:

Hoher Bias, hohe Varianz:

c) Die angesprochenen Elemente finden sich in folgender Graphik:




Bei geringer Modellkomplexität ist der Bias hoch und die Varianz gering (z.B. bei der linearen Regression). Das Verhältnis dreht sich mit zunehmender Modellkomplexität um (d.h. die Modelle enthalten mehr Parameter, z.B. werden zu einem linearen Modell nach und nach weitere Parameter aufgenommen). Der Gesamtfehler (gestrichelte Linie, als Summe der restlichen Bestandteile) hat dabei eine typische „U-Form". Eine optimale Modellkomplexität liegt an der Stelle, wo der Gesamtfehler minimal ist: Das Modell ist komplex genug, um den systematischen Fehler (Bias) klein zu halten, aber einfach genug, um nicht durch hohe Varianz eine Überanpassung zu erzeugen.

**Aufgabe 6** [4.3 Deep Learning 1] (27 Punkte)

a) [5 Punkte] Die Softmax-Aktivierungsfunktion ist gegeben durch:

$$s: \mathbb{R}^n \rightarrow \mathbb{R}^n, s_i = \frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}}, \forall\, 1 \leq i \leq n$$

Zu berechnen ist der Output dieser Funktion für den Vektor

$$z = (z_1, z_2, z_3)^t = (0,1,-1)^t$$

Welche zwei grundlegenden Eigenschaften besitzt die Softmax-Funktion $s$?

*Hinweis:* Die Ergebnisse sind auf zwei Stellen nach dem Komma zu runden.

b) [12 Punkte] Für ein Training eines Modells mit Aktivierungsfunktion aus Teil a) wird zum Update der Gewichte Backpropagation verwendet. Hierfür ist, unter anderem, die Berechnung von

$$\frac{\partial \mathcal{L}}{\partial z_j}, \forall\, 1 \leq j \leq n,$$

erforderlich, wobei $\mathcal{L}$ die Loss-Funktion bezeichnet. Allerdings wird zunächst die Jacobimatrix der Softmax-Funktion benötigt. Diese ist, unter zwingender Einhaltung der folgenden Schritte, zu berechnen:

i. Verwenden Sie den Ausdruck $ln(s_i)$, um eine Formel für $\frac{\partial s_i}{\partial z_j}$ zu ermitteln.

ii. Berechnen Sie, unter Nutzung der Formel für die Softmax-Funktion, den Ausdruck $ln(s_i)$ und vereinfachen Sie diesen so weit wie möglich.

iii. Nutzen Sie die Ergebnisse aus i. und ii. für die Berechnung von $\frac{\partial s_i}{\partial z_j}$ und vereinfachen Sie diesen Ausdruck so weit wie möglich.

c) [2 Punkte] Geben Sie mit dem Ergebnis aus b) die Jacobi-Matrix für die Softmax-Funktion s, d.h. $J_s$ für den Fall $n = 3$ in allgemeiner Form an.

*Hinweis:* Die Ergebnisse sind auf zwei Stellen nach dem Komma zu runden.

d) [8 Punkte] Die Berechnung von

$$\frac{\partial \mathcal{L}}{\partial z_j}, \forall\, 1 \leq j \leq n,$$

soll abschließend mit dem Cross-Entropy Loss, d.h.

$$\mathcal{L}(y, s) = -\sum_{l=1}^{c} y_l \, ln(s_l),$$

durchgeführt werden. Dabei ist $s_i$ die oben angegebene Softmax-Funktion und $y$ ein One-Hot Vektor (im Fall von $n = 3$ z.B. $y = (y_1, y_2, y_3)^t = (0,0,1)^t$). $c$ bezeichnet die Anzahl der Klassen, die zu berücksichtigen sind. Vereinfachen Sie das Ergebnis so weit wie möglich. Nutzen Sie für die Berechnung das Ergebnis aus Teil c).

**Lösungsvorschlag:**

a) Mit dem angegebenen Vektor $z = (z_1, z_2, z_3)^t = (0{,}1, -1)^t$ ergibt sich aus der angegebenen Formel für die Softmax-Aktivierungsfunktion $s$:

$$s = (0.24, 0.67, 0.09)$$

Grundlegende Eigenschaften der Softmax-Funktion:

i. Die Funktion erzeugt Wahrscheinlichkeiten (Einträge summieren sich zu 1).

ii. Die Rangordnung der Elemente bleibt erhalten, d.h.

$$-1 < 0 < 1 \ \rightarrow 0.09 < 0.24 < 0.67$$

b) Die einzelnen Schritte ergeben folgende Ergebnisse:

i. Zunächst berechnet man: $\frac{\partial ln(s_i)}{\partial z_j} = \frac{\partial ln(s_i)}{\partial s_i} \frac{\partial s_i}{\partial z_j} = \frac{1}{s_i} \frac{\partial s_i}{\partial z_j}$.

Daraus ergibt sich: $\frac{\partial s_i}{\partial z_j} = s_i \frac{\partial ln(s_i)}{\partial z_j}$

ii. $ln(s_i) = ln(e^{z_i}) - ln(\sum_{k=1}^{n} e^{z_k}) = z_i - ln(\sum_{k=1}^{n} e^{z_k})$

iii. Aus Punkt ii. ergibt sich zunächst:

$$\frac{\partial ln(s_i)}{\partial z_j} = \frac{\partial}{\partial z_j}\left(z_i - \ln\left(\sum_{k=1}^{n} e^{z_k}\right)\right) = 1_{\{i=j\}} - s_j,$$

wobei $1_{\{i=j\}}$ die Indikatorfunktion bezeichnet.

Mit Punkt i. ergibt sich daher folgender vereinfachter Ausdruck:

$$\frac{\partial s_i}{\partial z_j} = s_i \left(1_{\{i=j\}} - s_j\right)$$

c) Mit den Ergebnissen aus Teil c) ergibt sich:

$$J_{s_3} = \begin{pmatrix} s_1(1-s_1) & -s_1 s_2 & -s_1 s_3 \\ -s_2 s_1 & s_2(1-s_2) & -s_2 s_3 \\ -s_3 s_1 & -s_3 s_2 & s_3(1-s_3) \end{pmatrix}$$

Hinweis: Mit den Ergebnissen aus Teil b) erhält man:

$$\begin{pmatrix} 0.18 & -0.16 & -0.02 \\ -0.16 & 0.22 & -0.06 \\ -0.02 & -0.06 & 0.08 \end{pmatrix}$$

d) Zunächst ergibt sich:

$$\frac{\partial \mathcal{L}}{\partial z_j} = -\frac{\partial}{\partial z_j} \sum_{l=1}^{c} y_l ln(s_l) = -\sum_{l=1}^{c} \frac{y_l}{s_l} \frac{\partial s_l}{\partial z_j}$$

Mit den Ergebnissen aus Teil b) ergibt sich:

$$\frac{\partial \mathcal{L}}{\partial z_j} = -\sum_{l=1}^{c} \frac{y_l}{s_l} s_l \left(1_{\{l=j\}} - s_j\right) = -\sum_{l=1}^{c} y_l \left(1_{\{l=j\}} - s_j\right)$$

Weitere Umformungen ergeben dann:

$$\sum_{l=1}^{c} y_l s_j - \sum_{l=1}^{c} y_l 1_{\{l=j\}} = \sum_{l=1}^{c} y_l s_j - y_j = s_j - y_j,$$

wobei im letzten Schritt die Tatsache genutzt wurde, dass $\sum_{l=1}^{c} y_l = 1$.

Insgesamt ergibt sich damit:

$$\frac{\partial \mathcal{L}}{\partial z} = s - y$$

**Aufgabe 7** [5.3 Programmiersprachen für Data Science] (18 Punkte)

a) [5 Punkte] Nennen Sie zwei Grundsätze, die für das Verständnis der Programmiersprache R von fundamentaler Bedeutung sind. Nennen Sie im Anschluss drei Eigenschaften der dynamischen Typisierung, die in R verwendet wird.

b) [13 Punkte] Sie arbeiten als Data Scientist in einem Versicherungsunternehmen. Ihre Aufgabe ist es, die Loss Ratio (Schaden-/Prämienquote) pro Versicherungszweig („line") und Jahr zu berechnen.

Die Daten werden hierfür aus zwei csv-Dateien ermittelt:

`premium.csv` enthält die Spalten `policy_id, line, year, earned_premium`

`claims.csv` enthält die Spalten `policy_id, year, paid, outstanding`

Kontrollausgabe `premium.csv`:

| policy_id <chr> | line <chr> | year <dbl> | earned_premium <dbl> |
|---|---|---|---|
| P1 | Motor | 2022 | 1000 |
| P2 | Motor | 2022 | 2000 |
| P3 | Motor | 2022 | 3000 |
| P4 | Motor | 2023 | 4000 |
| P5 | Home | 2022 | 1500 |

5 rows

Kontrollausgabe `claims.csv`:

| policy_id <chr> | year <dbl> | paid <dbl> | outstanding <dbl> |
|---|---|---|---|
| P1 | 2022 | 500 | 200 |
| P2 | 2022 | 1000 | 300 |
| P3 | 2022 | 2000 | 500 |
| P4 | 2023 | 1000 | 200 |
| P5 | 2022 | 400 | 100 |

5 rows

Hierfür hat ein Kollege folgenden R-Code vorbereitet:

```
prem   <- read.csv("premium.csv")

claims <- read.csv("claims.csv")

incurred <- transform(claims, incurred = paid + outstanding)

by_line_year <- aggregate(incurred ~ year + line, data = merge(incurred, prem), sum)

prem_agg <- aggregate(earned_premium ~ year + line, data = prem, mean)

loss_ratio <- merge(by_line_year, prem_agg)

loss_ratio$lr <- loss_ratio$incurred %/% loss_ratio$earned_premium

loss_ratio <- loss_ratio[order(loss_ratio$year, loss_ratio$line), ]

print(loss_ratio[, c("year","line","lr")])
```

i. Beschreiben Sie zunächst in 2-3 Sätzen die Ablauflogik für die Berechnung des Loss-Ratios.

ii. Leider berechnet der R-Code nicht die erwarteten Werte. Identifizieren Sie zwei Fehler im R-Code. Erklären Sie, warum diese Fehler zu falschen Loss-Ratio-Werten führen. Aktuell liefert die Ausgabe des Programms mit dem oben angegebenen Code:

| | year <dbl> | line <chr> | lr <dbl> |
|---|---|---|---|
| 1 | 2022 | Home | 0 |
| 2 | 2022 | Motor | 2 |
| 3 | 2023 | Motor | 0 |

3 rows

Erwartet werden die folgenden Werte:

| | year <dbl> | line <chr> | lr <dbl> |
|---|---|---|---|
| 1 | 2022 | Home | 0.3333333 |
| 2 | 2022 | Motor | 0.7500000 |
| 3 | 2023 | Motor | 0.3000000 |

3 rows

iii. Geben Sie jeweils eine korrigierte R-Codezeile an, die den Fehler behebt.

**Lösungsvorschlag:**

a) Die zwei Grundsätze (nach John Chambers) sind:

i. Everything that exists is an object („Alles, was existiert, ist ein Objekt").

ii. Everything that happens is a function call („Alles, was geschieht, ist ein Funktionsaufruf").

Eigenschaften der dynamischen Typisierung:

i. Ein Objekt wird automatisch beim Interpretieren durch den zu belegenden Wert im Speicher angelegt,
ii. anhand des zu belegenden Werts wird automatisch ein eindeutiger Objekttyp bestimmt,
iii. bei Zuweisung eines Namens werden für diesen die Referenz auf den Speicherort implizit angelegt und der Objekttyp des Objekts übernommen,
iv. das Objekt existiert am allokierten Speicherort, solange es referenziert wird (d.h. mindestens einen Namen hat); andernfalls wird es automatisch gelöscht und der Speicher freigegeben.

b)

i. Das Programm soll für jede Versicherungssparte und jedes Jahr den Loss Ratio, also das Verhältnis von Schadenaufwand zu verdienten Prämien, berechnen. Dazu werden zunächst Schadendaten und Prämien über gemeinsame Schlüssel (policy_id, year) zusammengeführt, anschließend die Summen pro Jahr und Sparte gebildet und daraus das Verhältnis berechnet (mit lr = incurred / earned_premium). Das Ergebnis zeigt, wie profitabel jede Sparte in einem bestimmten Jahr war.

ii. Der erste Fehler befindet sich in Zeile 5: Anstatt die Prämienwerte zu summieren, wird der mean verwendet. Der zweite Fehler befindet sich in Zeile 7: Hier wird die ganzzahlige Division verwendet. Beide Fehler tragen zu den falschen Loss-Ratio-Werten bei.

iii. Die in Punkt ii. identifizierten Fehler können wie folgt korrigiert werden:

```
prem_agg <- aggregate(earned_premium ~ year + line, data=prem, sum)

loss_ratio$lr <- loss_ratio$incurred / loss_ratio$earned_premium
```

DAV

DEUTSCHE
AKTUARVEREINIGUNG e.V.

Written examination in

# Actuarial Data Science Basic

in accordance with examination regulations 2.0
of the German Actuarial Association
for the EAA Certificate in Actuarial Data Science

on 29 May 2026

*Notes:*

- A calculator is permitted as an aid.
- The total number of points is 180. The examination is passed if at least 90 points are achieved.
- Please check that your exam paper is complete. The exam consists of 34 pages.
- All answers must be justified and, in the case of calculation tasks, the solution must be clearly shown.
- When preparing the solutions, please avoid scattering them haphazardly across the individual parts of the assignment.
- To improve readability, we have opted not to use the gender-neutral designation (m/f/d) alongside the masculine and feminine forms.

*Members of the examination board:*

Axel Kiermaier, Dr. René Külheim, Prof. Jonas Offtermatt

**Task 1** [1.1 Fundamentals, 1.2 Digitalisation, 1.3 Social Environment & Ethics, 1.4 Data Protection] (28 points)

Your insurance company, a medium-sized German composite insurer, is planning **to introduce artificial intelligence** into several business processes. The aim is to work more efficiently through automation and data-driven decisions and to bring new products to market more quickly. The board of directors has set up an internal working group, in which you are also involved as an actuary.

The management now expects you to provide a well-founded assessment of **the introduction of AI in the insurance industry** – in terms of **technological potential**, **regulatory requirements** and **practical feasibility**.

a) [5 points] Explain the idea behind artificial intelligence in a way that is easy for the Executive Board to understand. Please also provide a **brief** explanation of the terms *machine learning*, *deep learning* and *generative AI*.

b) [6 points] Discuss the demands that digitalisation and the use of AI make on the application landscape and data processing of an insurance company. Address the aspects of data quality, data security, processing speed and IT architecture.

c) [9 points] Three specific AI applications were proposed during a workshop:

   1. The company uses AI to automatically identify suspicious claims and have them investigated in more detail. The customers concerned are not informed about the AI check.

   2. A new health app collects customer health data in real time and uses it to generate individual tariff recommendations. By downloading the app, customers automatically consent to data processing.

   3. The internal sales team receives an AI-supported prediction of which existing customers are likely to cancel their policies so that targeted recovery campaigns can be carried out. Age, place of residence and previous claims history are also taken into account.

   For each application, assess whether it is prohibited, permitted or a borderline case from the perspective of the GDPR. Justify your assessment in 1–2 sentences.

d) [8 points] The board of directors is surprised to hear that AI and data protection are related. Explain **four** of the key principles from Article 5 of the GDPR to them. (Name each principle and describe it in a short sentence.)

**Suggested solution:**

a) Artificial intelligence (AI) enables computers to perform tasks that would otherwise require human thinking, such as learning, deciding or recognising.

**Machine learning (ML)** is a subfield of AI in which systems recognise patterns from data instead of working according to fixed rules.
**Deep learning**, a subfield of ML, uses artificial neural networks to recognise particularly complex patterns in large amounts of data, e.g. in images or language.
**Generative AI** can generate new content from existing data, such as text, images or suggestions.
*(2 points for explanation and 1 point for each term explained.)*

b) **Data quality**: AI requires clean, complete and structured data. Poor data quality leads to incorrect decisions ("garbage in, garbage out").

**Processing speed**: Large amounts of data often need to be processed efficiently – this requires powerful systems and modern databases.
**IT architecture**: Traditional monolithic systems are reaching their limits. Modern architectures (e.g. microservices, cloud) are more flexible and better support AI applications.
**Data security**: AI often requires personal data – appropriate security measures are essential (e.g. encryption, access controls).
*(1.5 points per criterion)*

c) AI detects suspicious claims – customers are not informed

→ Borderline case
Reason: Automated processing with potentially adverse consequences is involved (Art. 22 GDPR). Without information and the possibility to object, there is a lack of transparency and lawfulness.

Health app records vital data – consent via terms and conditions when downloading
→ Prohibited
Reason: Health data is particularly sensitive (Art. 9 GDPR). Effective consent must be explicit and informed – consent via terms and conditions is not sufficient.

AI predicts probability of termination, including age, place of residence, claims history
→ Permitted, under certain conditions
Reason: Data processing is permissible if there is a legitimate interest (Art. 6(1)(f) GDPR) and data subjects are informed (Art. 13(f)). Profiling is permitted as long as no automated decision with legal effect is made.

*(3 points per assessment; the assessment may also be different, but the correct justification is important.)*

d) Article 5 of the General Data Protection Regulation (GDPR) lists seven fundamental principles that must be followed in the processing of personal data. Four of these should be selected and described. For the sake of completeness, all seven principles are listed here with brief explanations:

1. **Lawfulness, fairness and transparency**
   → Data must be processed lawfully, fairly and in a transparent manner in relation to the data subject.
2. **Purpose limitation**
   → Data may only be collected for specified, explicit and legitimate purposes and not further processed in a manner incompatible with those purposes.
3. **Data minimisation**
   → Only as much personal data as is actually necessary for the respective purpose may be processed.
4. **Accuracy**
   → It must be ensured that personal data is factually correct and up to date; inaccurate data must be corrected or deleted without delay.
5. **Storage limitation**
   → Data may only be stored for as long as is necessary for the purposes of processing. After that, it must be deleted or anonymised.
6. **Integrity and confidentiality** (also: security of processing)
   → Data must be protected against unauthorised access, loss or destruction by appropriate technical and organisational measures.
7. **Accountability**
   → The company must be able to demonstrate at all times that it complies with the above principles ("accountability").

*(2 points per principle with explanation, max. 8 points)*

**Task 2** [3.1 Data mining, 4.1 Supervised learning, 5.3 Programming languages for data science, 6.1 Use case] (32 points)

Your company is planning to develop an AI-based decision tree to assess the likelihood of new customers closing deals. A small training sample was collected manually for this purpose:

| ID | Age | Occupation | Previous_Policy | Income | Purchase |
|---|---|---|---|---|---|
| **1** | 25 | Employee | Yes | 3,000 | Yes |
| **2** | 45 | Self-employed | No | 6,500 | No |
| **3** | 30 | Student | No | 1,200 | Yes |
| **4** | 50 | Employee | Yes | 4,800 | Yes |
| **5** | 23 | Student | No | 1,000 | No |
| **6** | 35 | Self-employed | Yes | 5,800 | Yes |
| **7** | 29 | Employee | No | 3,200 | No |
| **8** | 40 | Self-employed | No | 5,000 | No |
| **9** | 33 | Student | Yes | 2,100 | Yes |
| **10** | 55 | Employee | Yes | 4,500 | Yes |

a) [5 points] Create a suitable visualisation to illustrate the relationship between any input variable (e.g. *occupation*, *previous policy*, *age*, *income*) and the target variable *purchase*. (*Please pay attention to the basic requirements for a visualisation: use of a ruler, axis labelling, etc. Sketches scribbled on paper will be graded with 0 points.)*
b) [6 points] Prepare the following variables for use in a decision tree model: `Occupation`, `Previous_Policy`, `Purchase`. Use appropriate numerical coding for each of the variables.
c) [5 points] Create a simple decision tree (minimum 2 levels/roots, maximum 3) that derives from the data whether a purchase is made or not. Use the following as the first split criterion: `"Previous_Policy"`. Justify your decision for further splits with the frequency of the target variables.

d) [8 points] Given the following decision tree:

```
Income > 4800 ?
├── Yes  → predict Purchase = No
└── No   → split by Previous_Policy?
       ├── Previous_Policy = Yes  → predict Purchase = Yes
       └── Previous_Policy = No   → predict Purchase = No
```

Apply this tree to the 10 training cases and create the resulting confusion matrix. Additionally, calculate the accuracy (proportion of correctly classified cases) and list the IDs of the incorrectly classified cases. Briefly explain (one sentence each) what could be the reason for these misclassifications.

e) [4 points] The following R code was used in the project:

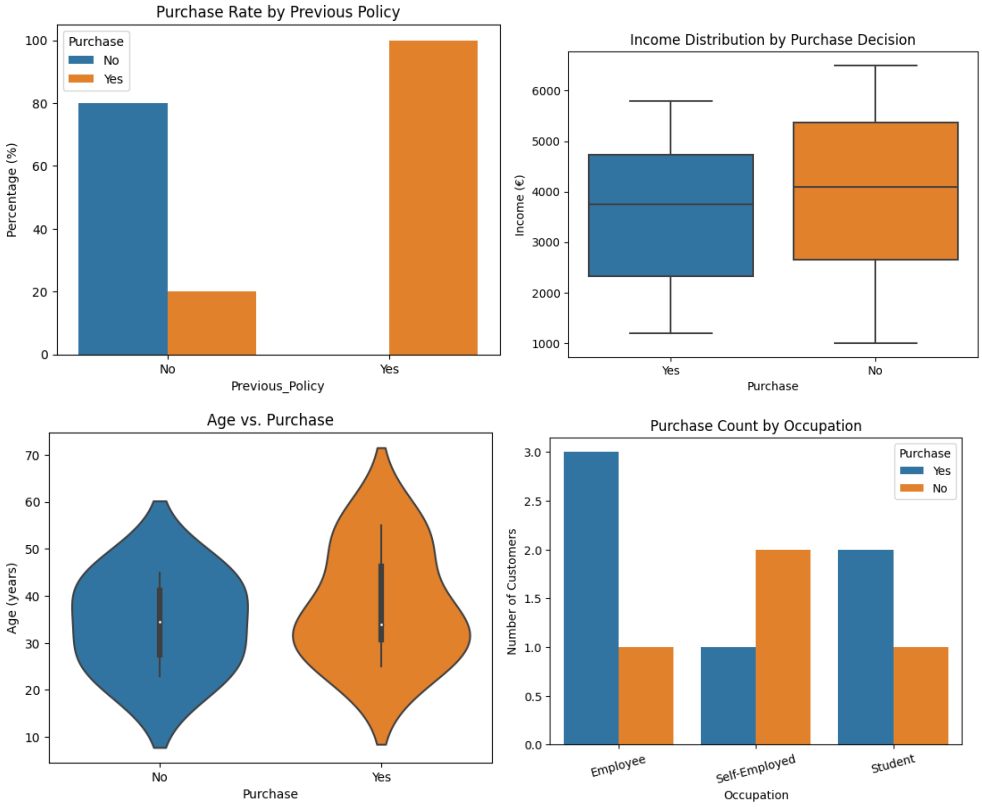
```
library(rpart)

model <- rpart(Purchase ~ Age + Income + Occupation +
Previous_Policy,
               data = df,
               method = "class",
               control = rpart.control(minsplit = 20, cp = 0.01))
```

What type of model is being trained here? What do the parameters method, minsplit and cp mean?

f) [4 points] What is *bagging* (bootstrap aggregating) in the context of decision trees? Briefly explain how bagging works and what advantage it offers over a single decision tree.

**Suggested solution:**

a) Possible visualisations include, for example:




*(The type of diagram is not specified; points will be deducted for unclean drawings/sketches.)*

b) Since these are categorical variables, each category is assigned a number, e.g.:

| **Variable** | **Possible values** | **Coding** |
|---|---|---|
| Occupation | Employee, Student, Self-employed | Employee = 0, Student = 1, Self-Employed = 2 |
| Previous_Policy | Yes / No | Yes = 1, No = 0 |
| Purchase | Yes / No | Yes = 1, No = 0 |

c) Creating the decision tree:

First split according to `"Previous_Policy"`:

```
|Previous_Policy = Yes| ID: 1, 4, 6, 9, 10| All → Purchase = Yes|
|Previous_Policy = No | ID: 2, 3, 5, 7, 8 | Only ID 3 → Yes |
```

"Previous_Policy" is therefore a good split criterion, with only one ID being sorted incorrectly.

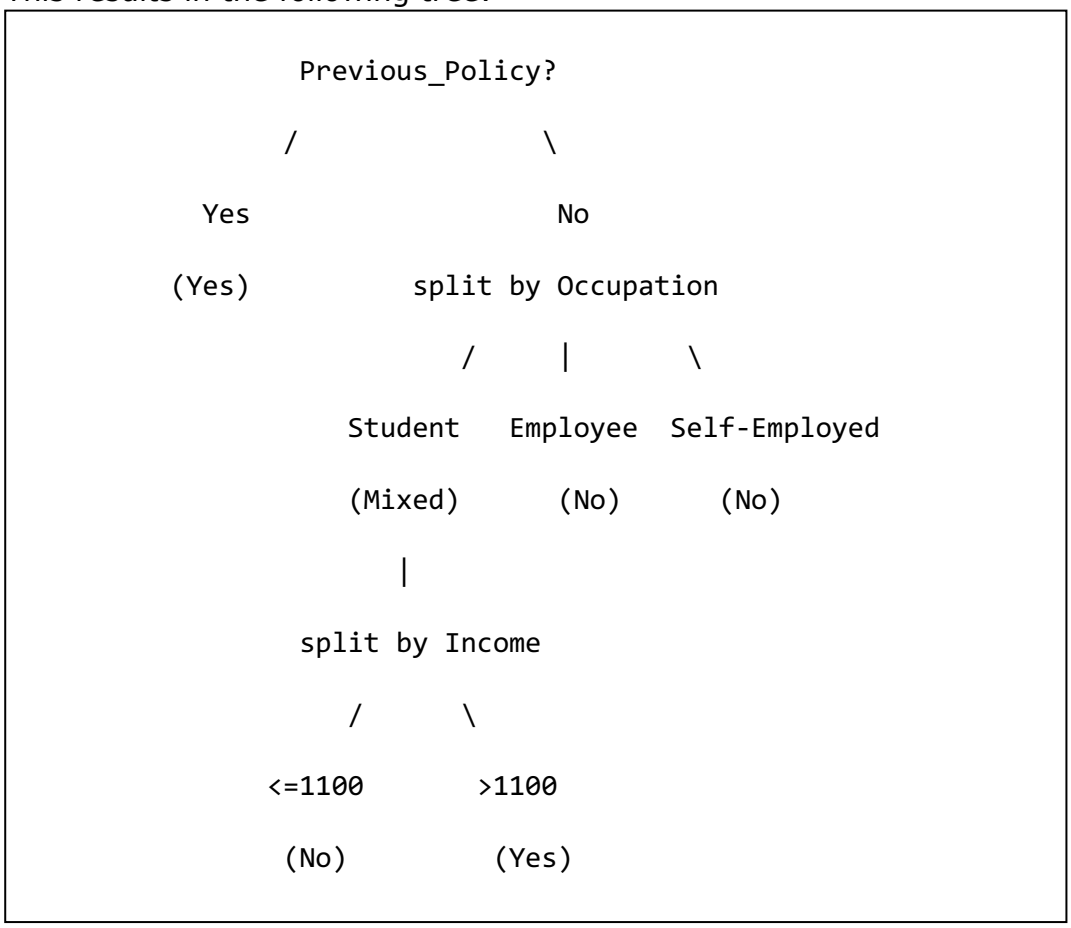
For another split criterion, you now have the choice between `Age`, `Income`, and `Occupation`. The graphs in task part a) show that none of the three provide good decisions, so we choose `Occupation`.

```
|Occupation = Student                    | ID: 3, 5    | Yes, No |
|Occupation = Self-Employed Or Employee | ID: 2, 7, 8 | All No |
```

As after the first split, we still have one incorrectly sorted ID.
We use Income as the third level:

```
|Income <= 1100  | ID: 5    | No  |
|Income > 1100   | ID: 3    | Yes |
```

This results in the following tree:




d) Applying the tree to the training cases yields the following result:

| ID | Income | Previous _Policy | Actual Purchase | Predicted Purchase | Correct |
|---|---|---|---|---|---|
| **1** | 3000 | Yes | Yes | Yes | Correct |
| **2** | 6500 | No | No | No | Correct |
| **3** | 1200 | No | Yes | No | Incorrect |
| **4** | 4800 | Yes | Yes | Yes | Correct |
| **5** | 1000 | No | No | No | Correct |

| | | | | | |
|---|---|---|---|---|---|
| **6** | 5800 | Yes | Yes | No | Incorrect |
| **7** | 3200 | No | No | No | Correct |
| **8** | 5000 | No | No | No | Correct |
| **9** | 2100 | Yes | Yes | Yes | Correct |
| **10** | 4500 | Yes | Yes | Yes | Correct |

The confusion matrix is therefore as follows:

| **actual \ predicted** | **pred Yes** | **pred No** |
|---|---|---|
| **actual Yes** | 4 | 2 |
| **actual No** | 0 | 4 |

And the accuracy for:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{4 + 4}{4 + 4 + 0 + 2} = \frac{8}{10} = 80\%$$

IDs 3 and 6 are incorrectly classified.

ID 3: Income = 1200 (≤4800) and Previous_Policy = No → Tree says No, but it is actually Yes. In this case, the customer made a purchase despite not having a previous contract and having a low income — an outlier that the rule does not capture.

ID 6: Income = 5800 (>4800) → Tree says No, but it is actually Yes. This customer has a high income but still made a purchase; the simple income cut-off led to a false prediction in this case.

*(4 points for correct confusion matrix, 2 points for correct accuracy, 1 point per explained misclassified ID)*

e) Type of model:

A decision tree classification model (rpart) that predicts the target variable `Purchase` based on the predictors `Age`, `Income`, `Occupation`, and `Previous_Policy`.

Key parameters:

- method = "class": Model for classification tasks (categorical target variable).
- control = rpart.control(minsplit = 20, cp = 0.01)
  - minsplit = 20: A node is only split if it contains at least 20 observations.

- cp = 0.01: Complexity parameter to avoid overfitting and unnecessary computation time; controls the tree size through pruning.

*(1 point for type of model, one point per parameter)*

f) Bagging (bootstrap aggregating) is a method in which **many decision trees** are trained on **different, randomly drawn samples (bootstrap samples)** from the training data. In random forests, only a random subset of the features is considered for each split so that the trees are as different as possible. The final prediction is made by **majority decision (classification) or average (regression)** of all trees. This combination **reduces** the **variance of the model** and **improves accuracy** by reducing overfitting of individual trees.

*(For full marks, the topics marked in bold must be mentioned.)*

**Task 3** [4.2 Unsupervised learning 1] (34 points)

You present some of the results of your new cluster study to your board and answer some of their questions.

a) [12 points] Name three cluster methods that are as different as possible and explain the approach of each of the three methods in two sentences (your board does not have much time!). The aim is to clearly present the basic principle, not the specific definitions and algorithms. Name two significant advantages and two disadvantages for each of the three methods.

b) [5 points] Outline the individual steps of the k-means algorithm in pseudo code. Instead of using the quadratic Euclidean distance or L2 metric as is usually the case, use the Manhattan distance or L1 metric.

   *Notes:*

   1. The Manhattan distance between two points is defined as the sum of the absolute differences of their individual coordinates.

   2. In Manhattan distance, the median is the optimal representative of a cluster (not the mean).

c) [5 points] How should the k-means algorithm generally deal with the issues of

   - outliers and

   - differently scaled features

   and why? How does the choice of metric (L2 vs. L1) affect these two issues? Briefly justify your answer.

d) [12 points] Given the following simple data set with the two standardised features Feature_1 and Feature_2:

| Item | Feature_1 | Feature_2 |
|---|---|---|
| 1 | -0.77 | -0.05 |
| 2 | -0.42 | -0.85 |
| 3 | 1.76 | 1.86 |
| 4 | -0.24 | -0.25 |
| 5 | 1.29 | 1.03 |
| 6 | -0.54 | -0.57 |
| 7 | -1.07 | -1.17 |

Perform the first iteration of k-means clustering based on the Manhattan distance/L1 metric.

Use items 1, 2 and 3 as starting centroids. What are the positions of the updated centroids after the first iteration? Your calculation steps should be comprehensible, at least as examples.

In a k-means clustering of the same data based on the Euclidean distance/L2 metric, the following cluster memberships result after the first iteration:

| Item | Cluster |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 1 |
| 5 | 3 |
| 6 | 2 |
| 7 | 2 |

Determine the updated centroids using the L2 metric and interpret the result.

**Suggested solution:**

a) Three different types of clustering methods:

- Centroid-based methods (e.g. k-means clustering)
- Connectivity-based methods (e.g. hierarchical agglomerative clustering)
- Density-based methods (e.g. DBSCAN)

*(It is sufficient to mention one example for each type; however, all three types of clustering methods must be represented in order to receive full marks. In some cases, more than two advantages or disadvantages are listed below; two advantages and two disadvantages are sufficient to receive full marks.)*

**k-means clustering:**

*Basic principle:*

The starting point is a selection of k observations (objects) as initial cluster centres (known as centroids), to which all objects are assigned in such a way that the objects within the clusters are as similar as possible. Subsequently, the new centre is determined for each cluster as the centre point/centre of gravity of the objects it contains, and all objects are reassigned to these new cluster centres.

*Advantages:*

Easy to understand, explain and implement.

Efficient with high-dimensional data, quickly finds the centres of the groups.

Quick adaptation to new observations.

*Disadvantages:*

Number of clusters must be specified (k is a hyperparameter).

Clusters are highly dependent on the initial choice of cluster centres.

Depending on the distance measure, leads to clusters of a certain shape (e.g. circular/spherical clusters with Euclidean norm); generally , these are convex shapes in the sense of the chosen norm – other shapes are not recognised.

Poor performance in regions with varying observation densities.

k-means is sensitive to outliers.

**Hierarchical clustering (agglomerative):**

*Basic principle:*

Starting from the individual points (observations), a tree structure is developed based on similarities. Depending on the level at which the "cut" is made, more or fewer clusters are obtained.

*Advantages:*

- No need to specify the number of clusters in advance.
- Reproducibility of results (e.g. independent of an initial choice of centroids).
- Good explainability due to the tree structure of the results.

*Disadvantages*

- High time and memory requirements compared to k-means, even for medium-sized data sets.
- High sensitivity to outliers.
- Interpretation of the dendrogram can be subjective.

**DBSCAN:**

*Basic principle:*

All points (objects, observations) are divided into core objects, dense reachable objects and noise objects, whereby this division is based on ε-neighbourhoods and minimum numbers of points in the ε-neighbourhood. All objects connected by the same core objects form a cluster and the noise objects represent outliers.

*Advantages:*

- No need to specify the number of clusters in advance.
- Only a few parameters are required.
- Handles outliers well.
- Irregularly shaped clusters can be detected.

*Disadvantages:*

- Non-deterministic assignment of non-core objects, i.e. those that are not dense.
- Runtime problems with high-dimensional data.
- Optimal determination of neighbourhood radius and minimum neighbourhood score can be challenging, especially when areas with very different densities are present.

b) A possible pseudocode for k-means with Manhattan distance looks like this:

Input:

Observations $x_1$ , …, $x_n$

Number of clusters k

1. Select k cluster centroids $\mu_1$, …, $\mu_k$ randomly from $\{x_1, \ldots, x_n\}$.

2. Repeat until there are no or only minimal changes:

   a) Assignment step:

      For each point $x_i$:

         Assign $x_i$ to the cluster $C_j$ whose centroid $\mu_j$ has the smallest Manhattan distance to $x_i$; which is calculated as

         $d(x_i, \mu_j) = \Sigma\ |x_{im} - \mu_{jm}|$ over all features m

   b) Update step:

```
        For each cluster C_j:

            Define the new centroids μ_j
            by the median of each feature of the points in C_j

    Return:

        Clusters C_1, …, C_k

        Centroids μ_1, …, μ_k
```

c) In general, in the assignment step of k-means, the metric determines how strongly large deviations influence the clustering. With the L2 metric (Euclidean distance), large deviations are weighted quadratically (i.e. more strongly), whereas with the L1 metric (Manhattan distance), they are only weighted linearly.

**Outliers:**

Outliers should generally be (temporarily) removed from the data set before the actual cluster analysis, as outliers distort the location of the cluster centres or, otherwise, each outlier claims its own cluster, thus driving up the necessary k.

With the L1 metric (Manhattan distance), large deviations and thus outliers have less of an effect on the cluster centres than with the L2 metric. In addition, medians are more robust than means (update step). Overall, outliers distort the result of k-means less with L1 than with L2.

**Differently scaled features:**

With both L1 and L2, differently scaled features lead to distorted distances; therefore, normalisation or standardisation of the data is generally necessary in both cases.

The need for standardisation is greater with the L2 metric, as it emphasises large deviations quadratically and is therefore strongly distorted by differently scaled features, while the L1 metric considers deviations linearly and is therefore somewhat more robust to differences in scale.

d) First, **the distance** of each item **to items 1, 2 and 3** is calculated. The smallest of these three distances then indicates to which item/cluster the respective item is to be assigned.

| Item | F_1 | F_2 | Dist. to 1 | Dist. to 2 | Dist. to 3 | Cluster |
|---|---|---|---|---|---|---|
| 1 | -0.77 | -0.05 | 0.00 | 1.15 | 4.44 | 1 |
| 2 | -0.42 | -0.85 | 1.15 | 0.00 | 4.89 | 2 |
| 3 | 1.76 | 1.86 | 4.44 | 4.89 | 0.00 | 3 |
| 4 | -0.24 | -0.25 | 0.73 | 0.78 | 4.11 | 1 |
| 5 | 1.29 | 1.03 | 3.14 | 3.59 | 1.30 | 3 |
| 6 | -0.54 | -0.57 | 0.75 | 0.40 | 4.73 | 2 |
| 7 | -1.07 | -1.17 | 1.42 | 0.97 | 5.86 | 2 |

Example: Calculation of the distance from item 5 to item 1:

$$|1.29 - (-0.77)| + |1.03 - (-0.05)| = 2.06 + 1.08 = 3.14.$$

Note: For items 1 to 3, the mutual distance does not need to be determined in order to determine the cluster membership, since by definition item 1 belongs to cluster 1, etc. Therefore, the values in the table above are greyed out.

The **updated centroids** are calculated using the median of the feature values in the respective cluster as follows (rounded to two decimal places):

| Cluster | Centroid |
|---|---|
| 1 | (-0.51 ; -0.15) |
| 2 | (-0.54 ; -0.85) |
| 3 | (1.53 ; 1.45) |

Example: The value of Feature_1 for the new centroid of cluster 1 is calculated as the median of the values -0.77 and -0.24; i.e. ((-0.77) + (-0.24)) / 2 = -0.505. The value of Feature_1 for the new centroid of cluster 2 is calculated as the median of the values -0.42, -0.54 and -1.07, i.e. -0.54.

When using the **L2 metric,** the updated centroids are calculated using the mean value of the feature values in the respective cluster as follows:

| Cluster | Centroid |
|---|---|
| 1 | (-0.51 ; -0.15) |
| 2 | (-0.68 ; -0.86) |
| 3 | (1.53 ; 1.45) |

Example: The value of Feature_1 for the new centroid of cluster 1 is calculated as the mean value of the values -0.77 and -0.24; i.e. ((-0.77) + (-0.24)) / 2 = -0.505.

**Observation/interpretation:**

The cluster memberships after the first iteration are identical for the L1 and L2 metrics.

The centroids of clusters 1 and 3 are identical because the mean and median coincide here, particularly due to the very small sample size and the even number of items contained in these clusters.

The centroid of cluster 2 differs slightly because the mean used in L2 is slightly more attracted to the outlier (item 7) than the median used in L1. However, since this is not an extreme outlier and the data are relatively close together overall, the effect is relatively small.

**Task 4** [1.1 Fundamentals, 3.1 Data Mining 1, 2.1 Data Management 1] (26 points)

a) [7 points] Explain the differences between structured, unstructured and semi-structured data (providing a simple example for each), as well as the significance of this distinction in the context of data science. Additionally, list at least three as diverse as possible examples of data sources in the actuarial environment.

b) [7 points] Given the following sample data set with the features offer number, name, start of insurance, entry age, gender, sum insured and previous insurance (y/n):

| QuotationNumber | Name | InceptionDate | EntryAge | Sex | SumAssured | PreviousCover |
|---|---|---|---|---|---|---|
| 001200345 | Peter Meier | 01.12.2024 | 23 | M | 150000 | TRUE |
| 001200856 | Petra Huber | 01.01.2025 | 63 | F | 35000 | FALSE |

For each of the seven features, specify the characteristic type (qualitative, etc.), the scale level (ordinal scale, etc.) and a meaningful data type (character, etc.).

c) [6 points] To process the above sample data set using the given information to the maximum extent possible:

What might this sample data set look like as a csv file? What about as a JSON file?

d) [6 points] Which command can you use to read the csv file test.csv with R? Name five elements that need to be specified in addition to the file name – even if they are often assigned default values. We are not looking for the parameter identifiers for the function call, but a brief description of the content.

**Suggested solution:**

a) **Structured data:**

... have a predefined format (a normalised form) in which all information is organised.

Example: relational database with defined row and column positions for all data.

**Unstructured data:**

... does not have a clearly comprehensible structure.

Examples: texts, images, audio and video files.

**Semi-structured data:**

... is neither completely structured nor completely unstructured.

Example: The message header of an email has a clear structure (sender, recipient, etc.). The actual content is usually unstructured text.

**Data sources in the actuarial environment:**

These include, for example,

- Information from inventory management systems, DWH systems, etc.
- Information from wearables, data from the use of apps, homepages, etc.
- Customer data (in addition to classic inventory data, also from correspondence with the customer), e.g. information about marital status and family members, etc.
- Weather data
- Accident data, data on burglaries, etc.
- Data from accident reporting systems in cars

*(Citing three different sources is sufficient here.)*

**Significance of the distinction:**

Structured data can be analysed relatively easily, e.g. with SQL. Much of the data generated or used in companies today is unstructured (see the list of data sources above). The less structured the data is, the more complex its processing and analysis is in the context of data science tasks.

b) The following table shows the assignments for the individual features:

| | Feature type | Scale level | Data type |
|---|---|---|---|
| QuotationNumber | Qualitative | Nominal scale | Character |
| Name | Qualitative | Nominal scale | Character |
| InceptionDate | Quantitative | Interval scale | Date |
| EntryAge | Quantitative | Ratio scale | Integer |
| Sex | Qualitative | Nominal scale | Character |
| SumAssured | Quantitative | Ratio scale | Integer/Float |
| PreviousCover | Qualitative | Nominal scale | Logical |

c) The sample data set in csv format:

```
QuotationNumber;Name;InceptionDate;EntryAge;Sex;SumAssured;PreviousCover
001200345;Peter Meier;01.12.2024;23;M;150000;TRUE
001200856;Petra Huber;01.01.2025;63;F;35000;FALSE
```

And the same in JSON format:

```
[
  {
    "QuotationNumber": "001200345",
    "Name": "Peter Meier",
    "InceptionDate": "01.12.2024",
    "EntryAge": 23,
    "Sex": "M",
    "SumAssured": 150000,
    "PreviousCover": true
  },
  {
    "QuotationNumber": "001200856",
    "Name": "Petra Huber",
    "StartDate": "01/01/2025",
    "EntryAge": 63,
    "Sex": "F",
    "SumAssured": 35000,
    "PreviousCover": false
  }
]
```

*Note:* Brackets, commas, and quotation marks in the JSON representation are essential for the correctness of the solution. If they are missing, corresponding points will be deducted. Indentation, however, is not technically mandatory.

d) The command for reading test.csv into R is:

```
read.csv("test.csv")
```

In addition to the file name, you can specify the following information:

• Separator (in the solution to sub-task c, ";" is used, but TAB or "," are also common)

• Heading (y/n)

• Line break – normally "\n" in Unix-based operating systems and "\n\r" in Windows

• Quote character

• Escape character

• Decimal separator and, if applicable, thousand separator for numbers

• Encoding

*Note:* Mentioning five of these pieces of information is sufficient to receive full marks.

**Task 5** [3.1 Analytics 1] (15 points)

a) [5 points] Explain what is meant by the bias-variance tradeoff. The (local) formula for this must be explicitly stated and the individual components must be explained. All under the assumption of $y = f(x) + \epsilon$ with N(0,1) distributed noise.

b) [4 points] Visualise the possible combinations of high/low bias/variance in bull's-eye diagrams (see appendix). Explain the resulting diagrams.

c) [6 points] Make a sketch in which the (model) error is plotted on the y-axis and the model complexity on the x-axis. Explain how the typical curves of total error, variance and bias run. Where would you determine the optimal model complexity?

**Suggested solution:**

a) In machine learning problems, the bias-variance trade-off describes the problem of simultaneously minimising two sources of error that prevent supervised learning algorithms from generalising beyond their training data. Both components are found in the following formula (local consideration for an $x$ ):

$$E(y - f(x))^2 = Var\left(\hat{f}(x)\right) + Bias\left(\hat{f}(x)\right)^2 + Var(\varepsilon)$$

Where $y = f(x) + \varepsilon$ . In addition $\hat{f}(x)$ denotes a function that approximates a true function $f(x)$ as closely as possible (various methods for determining $\hat{f}$ have been presented in the course).
The bias measures how much the model's predictions systematically deviate from the actual target value.
The variance measures how much the model predictions fluctuate when training with slightly different training data.

b) In the bull's-eye diagram, the interpretation is as follows: The point in the centre corresponds to a model that predicts the data perfectly. The further away from this point, the worse the prediction values become. Each black dot shown below corresponds to the realisation of a model (i.e. the model building process was carried out for different training data sets). The following situations can then be distinguished:

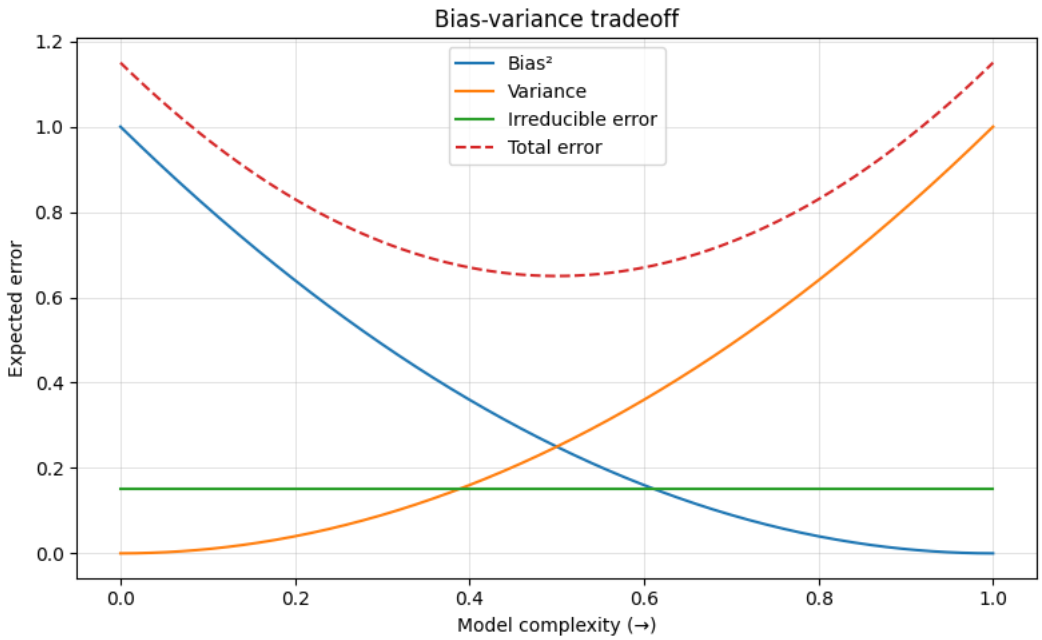
Low bias, low variance:

Low bias, high variance:

High bias, low variance:

High bias, high variance:

c) The elements mentioned can be found in the following graph:




With low model complexity, the bias is high and the variance is low (e.g. in linear regression). The ratio reverses as model complexity increases (i.e. the models contain more parameters, e.g. additional parameters are gradually added to a linear model). The total error (dashed line, as the sum of the remaining components) typically has a "U-shape". Optimal model complexity is achieved when the total error is minimal: the model is complex enough to keep the systematic error (bias) small, but simple enough to avoid overfitting due to high variance.

**Task 6** [4.3 Deep Learning 1] (27 points)

a) [5 points] The softmax activation function is given by:

$$s: \mathbb{R}^n \rightarrow \mathbb{R}^n, s_i = \frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}}, \forall\, 1 \leq i \leq n$$

Calculate the output of this function for the vector

$$z = (z_1, z_2, z_3)^t = (0,1,-1)^t$$

What are the two fundamental properties of the softmax function $s$?

*Note:* The results should be rounded to two decimal places.

b) [12 points] For training a model with activation function from Part a), backpropagation is used to update the weights. Among other things, this requires the calculation of

$$\frac{\partial \mathcal{L}}{\partial z_j}, \forall\, 1 \leq j \leq n,$$

where $\mathcal{L}$ denotes the loss function. However, the Jacobian matrix of the softmax function is required first. This must be calculated by strictly following the steps below:

i. Use the expression $ln(s_i)$ to determine a formula for $\frac{\partial s_i}{\partial z_j}$ .

ii. Using the formula for the softmax function, calculate the expression $ln(s_i)$ and simplify it as much as possible.

iii. Use the results from i. and ii. to calculate $\frac{\partial s_i}{\partial z_j}$ and simplify this expression as much as possible.

c) [2 points] Using the result from b), specify the Jacobi matrix for the softmax function s, i.e. $J_s$ for the case $n = 3$ in general form.

*Note:* The results should be rounded to two decimal places.

d) [8 points] The calculation of

$$\frac{\partial \mathcal{L}}{\partial z_j}, \forall\, 1 \leq j \leq n,$$

should be performed using the cross-entropy loss, i.e.

$$\mathcal{L}(y, s) = -\sum_{l=1}^{c} y_l \, ln(s_l).$$

$s_i$ is the softmax function specified above and $y$ is a one-hot vector (in the case of $n = 3$ , for example $y = (y_1, y_2, y_3)^t = (0,0,1)^t$ ). $c$ denotes the number of classes to be considered. Simplify the result as much as possible. Use the result from part c) for the calculation.

**Suggested solution:**

a) With the specified vector $z = (z_1, z_2, z_3)^t = (0,1,-1)^t$ , the specified formula for the softmax activation function $s$ results in:

$$s = (0.24, 0.67, 0.09)$$

Basic properties of the softmax function:

i. The function generates probabilities (entries add up to 1).

ii. The ranking of the elements is retained, i.e.

$$-1 < 0 < 1 \rightarrow 0.09 < 0.24 < 0.67$$

b) The individual steps yield the following results:

i. First, calculate: $\frac{\partial ln(s_i)}{\partial z_j} = \frac{\partial ln(s_i)}{\partial s_i} \frac{\partial s_i}{\partial z_j} = \frac{1}{s_i} \frac{\partial s_i}{\partial z_j}$ .

This results in: $\frac{\partial s_i}{\partial z_j} = s_i \frac{\partial ln(s_i)}{\partial z_j}$

ii. $ln(s_i) = ln(e^{z_i}) - ln(\sum_{k=1}^{n} e^{z_k}) = z_i - ln(\sum_{k=1}^{n} e^{z_k})$

iii. From point ii., the following is initially obtained:

$$\frac{\partial ln(s_i)}{\partial z_j} = \frac{\partial}{\partial z_j}\left(z_i - \ln\left(\sum_{k=1}^{n} e^{z_k}\right)\right) = 1_{\{i=j\}} - s_j,$$

Where $1_{\{i=j\}}$ denotes the indicator function.

Point i. therefore yields the following simplified expression:

$$\frac{\partial s_i}{\partial z_j} = s_i \left(1_{\{i=j\}} - s_j\right)$$

c) Using the results from part c), we obtain:

$$J_{s_3} = \begin{pmatrix} s_1(1-s_1) & -s_1 s_2 & -s_1 s_3 \\ -s_2 s_1 & s_2(1-s_2) & -s_2 s_3 \\ -s_3 s_1 & -s_3 s_2 & s_3(1-s_3) \end{pmatrix}$$

Note: Using the results from part b), we obtain:

$$\begin{pmatrix} 0.18 & -0.16 & -0.02 \\ -0.16 & 0.22 & -0.06 \\ -0.02 & -0.06 & 0.08 \end{pmatrix}$$

d) First, we obtain:

$$\frac{\partial \mathcal{L}}{\partial z_j} = -\frac{\partial}{\partial z_j}\sum_{l=1}^{c} y_l ln(s_l) = -\sum_{l=1}^{c} \frac{y_l}{s_l}\frac{\partial s_l}{\partial z_j}$$

Using the results from part b), we obtain:

$$\frac{\partial \mathcal{L}}{\partial z_j} = -\sum_{l=1}^{c} \frac{y_l}{s_l} s_l\left(1_{\{l=j\}} - s_j\right) = -\sum_{l=1}^{c} y_l\left(1_{\{l=j\}} - s_j\right)$$

Further transformations then yield:

$$\sum_{l=1}^{c} y_l s_j - \sum_{l=1}^{c} y_l 1_{\{l=j\}} = \sum_{l=1}^{c} y_l s_j - y_j = s_j - y_j,$$

where in the last step, the fact that $\sum_{l=1}^{c} y_l = 1$ is used.

Overall, this results in:

$$\frac{\partial \mathcal{L}}{\partial z} = s - y$$

**Task 7** [5.3 Programming languages for data science] (18 points)

a) [5 points] Name two principles that are fundamental to understanding the programming language R. Then name three characteristics of dynamic typing that are used in R.

b) [13 points] You work as a data scientist in an insurance company. Your task is to calculate the loss ratio (claims/premiums ratio) per insurance line and year.

The data for this is determined from two CSV files:

`premium.csv` contains the columns `policy_id`, `line`, `year`, `earned_premium`

`claims.csv` contains the columns `policy_id`, `year`, `paid`, `outstanding`

Control output `premium.csv`:

| policy_id <chr> | line <chr> | year <dbl> | earned_premium <dbl> |
|---|---|---|---|
| P1 | Motor | 2022 | 1000 |
| P2 | Motor | 2022 | 2000 |
| P3 | Motor | 2022 | 3000 |
| P4 | Motor | 2023 | 4000 |
| P5 | Home | 2022 | 1500 |

5 rows

Control output `claims.csv`:

| policy_id <chr> | year <dbl> | paid <dbl> | outstanding <dbl> |
|---|---|---|---|
| P1 | 2022 | 500 | 200 |
| P2 | 2022 | 1000 | 300 |
| P3 | 2022 | 2000 | 500 |
| P4 | 2023 | 1000 | 200 |
| P5 | 2022 | 400 | 100 |

5 rows

A colleague has prepared the following R code for this purpose:

```
prem <- read.csv("premium.csv")

claims <- read.csv("claims.csv")

incurred <- transform(claims, incurred = paid + outstanding)

by_line_year <- aggregate(incurred ~ year + line, data = merge(incurred, prem), sum)

prem_agg <- aggregate(earned_premium ~ year + line, data = prem, mean)

loss_ratio <- merge(by_line_year, prem_agg)

loss_ratio$lr <- loss_ratio$incurred %/% loss_ratio$earned_premium

loss_ratio <- loss_ratio[order(loss_ratio$year, loss_ratio$line), ]

print(loss_ratio[, c("year","line","lr")])
```

i. First, describe the process logic for calculating the loss ratio in 2-3 sentences.

ii. Unfortunately, the R code does not calculate the expected values. Identify two errors in the R code. Explain why these errors lead to incorrect loss ratio values. Currently, the output of the programme with the code above is as follows:

| | year <dbl> | line <chr> | lr <dbl> |
|---|---|---|---|
| 1 | 2022 | Home | 0 |
| 2 | 2022 | Motor | 2 |
| 3 | 2023 | Motor | 0 |

3 rows

The following values are expected:

| | year <dbl> | line <chr> | lr <dbl> |
|---|---|---|---|
| 1 | 2022 | Home | 0.3333333 |
| 2 | 2022 | Motor | 0.7500000 |
| 3 | 2023 | Motor | 0.3000000 |

3 rows

iii. Provide a corrected R code line for each error that fixes the error.

**Suggested solution:**

a) The two principles (according to John Chambers) are:

i. Everything that exists is an object.

ii. Everything that happens is a function call.

Properties of dynamic typing:

i. An object is automatically created in memory during interpretation by the value to be assigned.
ii. A unique object type is automatically determined based on the value to be assigned.
iii. When a name is assigned, the reference to the storage location is implicitly created for it and the object type of the object is adopted.
iv. The object exists at the allocated storage location as long as it is referenced (i.e. has at least one name); otherwise, it is automatically deleted and the memory is freed.

b)

i. The programme is designed to calculate the loss ratio, i.e. the ratio of claims expenditure to earned premiums, for each insurance line and each year. To do this, claims data and premiums are first merged using common keys (policy_id, year), then the totals per year and line are calculated and the ratio is calculated from this (with lr = incurred / earned_premium). The result shows how profitable each line of business was in a given year.

ii. The first error is in line 5: instead of summing the premium values, the mean is used. The second error is in line 7: integer division is used here. Both errors contribute to the incorrect loss ratio values.

iii. The errors identified in point ii. can be corrected as follows:

```
prem_agg <- aggregate(earned_premium ~ year + line, data=prem, sum)

loss_ratio$lr <- loss_ratio$incurred / loss_ratio$earned_premium
```