Lösungen

Teil 1

- Frage 1.1: 435. Dazu scrollt man ganz an das Ende und guckt sich den Index der letzten Datenreihe an. Eventuell muss man den Zahlen durch die Vergrösserung der Spaltenbreite genügend Platz verschaffen, damit sie angezeigt werden. Oder man rechnet im Preprocess-Tab unter dem Attribut-Eintrag ‘Class’ die zwei Vorkommen der Klasse Republicans und Democrats zusammen.
- Frage 1.2: 'n' (no). Dies ist direkt abzulesen in der vierten Zeile, die Titelzeile nicht mitberechnet.
- Frage 1.3: Demokrat, ebenso direkt abzulesen. Diesmal in der letzten Spalte.
- Frage 1.4: 267 Democraten, 168 Repbulikaner. Dies ist ersichtlich, wenn man im Explorer in der Attribute-Auflistung auf 'class' klickt.
- Frage 1.5: Ja, weil fast keine Republikaner mit 'no' abgestimmt haben.
- Frage 1.6: 170. dies ist ersichtlich wenn man im Weka-Explorer in der Attribut-Auflistung auf 'crime' klickt.

Teil 2

- Frage 2.1: Die Wahrscheinlichkeit des mehrdimensionalen Datenpunkts xi zur Klasse Y zu gehören ist mit den trainierbaren Modell-Parametern βi wie im Bild. Die finale Klasse Y ist diejenige, mit der höchsten Wahrscheinlichkeit.

$$\mathrm{P}(Y = 1 \mid X_i = x_i) = \frac{\exp(\beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \ldots + x_{ik}\beta_k)}{1 + \exp(\beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \ldots + x_{ik}\beta_k)}.$$

- Frage 2.2: 98.8506% (kann evtl. leicht abweichen). 86 richtig klassifiziert, 1 falsch.
- frage 2.3: 87 Test-Datenreihen insgesamt. Beachte: 86 richtig + 1 falsch = 87 gesamt.
- Frage 2.4: Der Anteil der richtig klassifizierten Datenreihen ändert sich. Ausserdem sinkt die Gesamtanzahl der Test-Datenreihen auf 43.
- Frage 2.5: Zum einen ist die Anzahl von 43 Test-Datenreihen so gering, dass ein einzelner anders klassifizierter bereits eine Auswirkung von 2.32% (100% / 43 Datenreihen) auf die Klassifizierungsgenauigkeit hat. Zum anderen ist der Prozess auch ein wenig zufällig, die Anzahl der korrekt oder falsch klassifizierten Datenreihen kann leicht abweichen.
- Frage 2.6: Test-Datenreihen = Gesamtdatenreihen * (100 - pecentage-split), Ge-

samtdatenreihen = 435

- Frage 2.7: 50%, die Fälle, wo sie auf dem Rand landet, mal vernachlässigt.
- Frage 2.8: Ja, weil wir die Fehler über mehr Datenreihen berechnen können und damit statistische Ausreisser das Resultat weniger beeinflussen können.
- Frage 2.9: Die Anzahl der Totalen Test-Datenreihe ist 435. Diese errechnet sich aus total 8 Durchgängen zu je 54 oder 55 Datenreihen, je nach Anteil.
- frage 2.10: Ja, wähle dazu beim Drop-Down-Menü ein anderes Attribut als 'Class' aus.
- Frage 2.11: Die Verteilung von ja und nein stimmen bei 'immigration' korrelliert fast nicht mit der Parteizugehörigkeit, während andere Attribute dies sehr wohl tun. Daher ist es sehr schwierig (statistisch verrauscht), immigration mit den anderen Attributen vorherzusagen. Dementsprechend auch die Genauigkeit von 55%.

Teil 3

- Frage 3.1: 48 von 1200, also relativ wenig. Ja, 'vote' war ausgeglichener.
- Frage 3.2: Für 'tested-postive' = 'n' sind all die Attribute klar, wo keine Fälle 'tested_positive' = 'y' auftauchen. Für 'tested_positive' = 'y' hingegen gibt es abgesehen von 4 Datenreihen bei 'body_temp' keine Attribute, welche nur 'tested_positive' = 'y' enthalten. Daher wird die vorhersage von 'y' schwerer sein.
- Frage 3.3: Wie in der zweiten Teilfrage angedeutet, können durchaus Datenbeispiele auftauchen, die unser trainierter Algorithmus aufgrund der verfügbaren Trainingsdaten klar als 'n' (oder 'y') klassifiziert, die aber eigentlich 'y' (oder 'n' im anderen Fall) wären. Der Grund dafür liegt in der Tatsache, dass unser Datenset nur eine „zufällige“ Ziehung der Gesamtpopulation ist und dementsprechend auch statistische Variationen aufweisen kann. Ausserdem sind die Fälle mit 'tested_positive' = 'y' sehr selten und bei gewissen Attributenwerten nicht ausreichen repräsentativ. Ein Beispiel dafür ist, dass in den Daten kein Sample für 'body_temy' < 45 und 'tested_positive' = 'y' vorkommt, dies jedoch auftreten könnte. Wichtiger Lernschritt: Diese Algorithmen sind limitiert auf das, was sie gesehen haben und können in diesen Fällen nicht generalisieren und dazulernen, ohne dass sie ausdrücklich dafür konstruiert werden. Dieses 'online-learning' hat jedoch andere Nachteile, z. B. dass sie über die Zeit unüberwacht falsche Dinge lernen können.
- Frage 3.4: unter 'J48 pruned tree' ist der Klassifzierungsvorgang ausgewiesen. Zuerst guckt er, ob 'body_temp' grösser als 37.76 ist. Falls ja, dann ist das Resultat 'n' (mit 1121 korrekt klassifizierten Trainings-Daten und 22 falsch). Falls 'body_temp' kleiner als 37.76 ist, gehen wir einen Schritt weiter und vergleichen 'days_since_dec_190' <= 100. Falls nein geben wir 'y' aus. Falls ja gehen wir noch einen Schritt weiter. Der Rest sollte selbsterklärend sein. Wichtiger Lernschritt: Das

Klassifikationsvorgehen muss für Menschen nicht zwingend Sinn machen. Es ist lediglich für den entsprechenden Algorithmus die sinnvollste Methode gewesen. Maschinen haben kein rationales Reflektionsvermögen.

- Frage 3.5: 96.667%. ja, das ist relative hoch. Er klassifiziert 96.667% der Fälle korrekt. Optimal wäre natürlich 100%.
- Frage 3.6: 1148 Datenreihen (sieht man in der Confusion Matrix), Anteil ist 99.7% (TP Rate für Klasse 'n').
- Frage 3.7: 12 Datenreihen, Anteil ist nur 25% (TP rate für Klasse 'y'). Diese Vorhersage-Genauigkeit von 25% ist deutlich niedriger, obwohl die Gesamt-Genauigkeit des Algorithmus 97% entspricht. Der Grund dafür liegt im seltenen Vorkommen der Klasse 'tested_positive' = 'y'. Diese Asymmetrie der Genauigkeiten ist bekannt als Baserate-Fallacy (Die Basisraten-Falle oder Prävalenzfehler).

Teil 4

- Frage 4.1: Eine Einführung gibt es hier: https://www1.se.cuhk.edu.hk/~eclt5810/lecture/weka_tutorial/weka-tutorial-3.pdf
- Frage 4.2: vielleicht bist du bei deiner Internetrecherche auf den Term 'Epochen' gestossen. Dieser gibt an, wie oft über das gesamte Trainins-Datenset iteriert wird. Diese 'Training Time' ist genau die Anzahl der Epochen.
- Frage 4.3: Die Gesamtgenauigkeit ist 82.25%. Einfaches Raten führte bei den 26 Klassen zu 100% / 26 = 3.8%. Das Netzwerk ist also gut.
- Frage 4.4: Diese Zahlen geben für jeden Node (Sigmoid-Node) die Gewichte an, mit denen der Wert des zugehörigen vorherigen Nodes multipliziert und dann mit all den anderen Gewichten für den selben Node addiert werden, bevor dann eine Sigmoid-Aktivierungs-Funktion darauf angewandt wird um den wert des derzeitigen Nodes für den nächsten Schritt (Layer) auszuwerten.