Programmentwurf
Bayes Netz

Informatik an der Dualen Hochschule Baden Württemberg Stuttgart

Von Michael Müller und Jan-Nicolai Geistler

April 13, 2020

Bearbeitungszeitraum:	06.04.2020 – 11.05.2020
Matrikelnummern	3222652, 4881231
Kurs:	STG-TINF17A - Künstliche Intelligenz
Unternehmen:	Hewlett-Packard Enterprise
Dozent:	Dirk Reichardt

Inhaltsverzeichnis

1 Installation **1**

2 Design des Bayesian Networks (Graph) **2**

3 Berechnen der CPTs **8**

4 Implementierung der Bayesian Networks **10**

5 Test, Evaluation und Fazit **10**

1 Installation

Das Programm ist in Python geschrieben. Bevor es verwendet werden kann, sollten diese Packages installiert sein:

```
pandas
pomegranate
```

Wird das GitHub-Repository verwendet, kann dies ganz leicht mit folgendem Befehl durchgeführt werden:

```
pip install -r requirements.txt
```

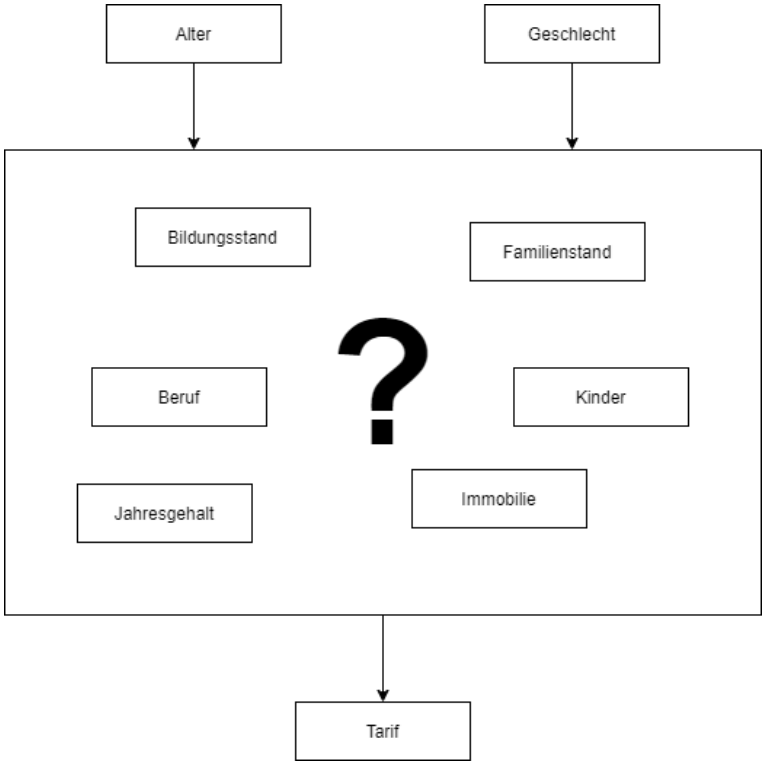
Der Programmverlauf ist relativ selbsterklärend, da das Programm einen durch die verschiedenen Phasen durchführt.Es gibt auch die Möglichkeit diese Dokumentation in einer geführten Version als Jupyter Notebook zu lesen. Diese ist im entsprechenden Ordner zu finden und beinhaltet den gleichen Code. Alle Projektdateien können auch unter `https://github.com/Ovakefali13/Project_KI_2017` gefunden werden. Im folgenden wird nun genauer auf den Code und die Implementierung eingegangen.

2 Design des Bayesian Networks (Graph)

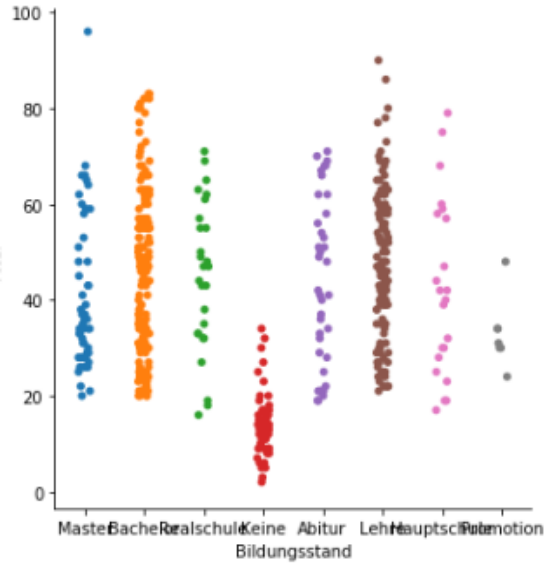
Gegeben sind 9 Variablen: Geschlecht, Familienstand, Alter, Kinder, Bildungsstand, Beruf, Jahresgehalt, Immobilienbesitz und Versicherungstarif. In diesem Kapitel möchten wir schrittweise ein 'sinnvolles' probabilistisches Netz entwickeln. Wir starten mit folgenden Annahmen:

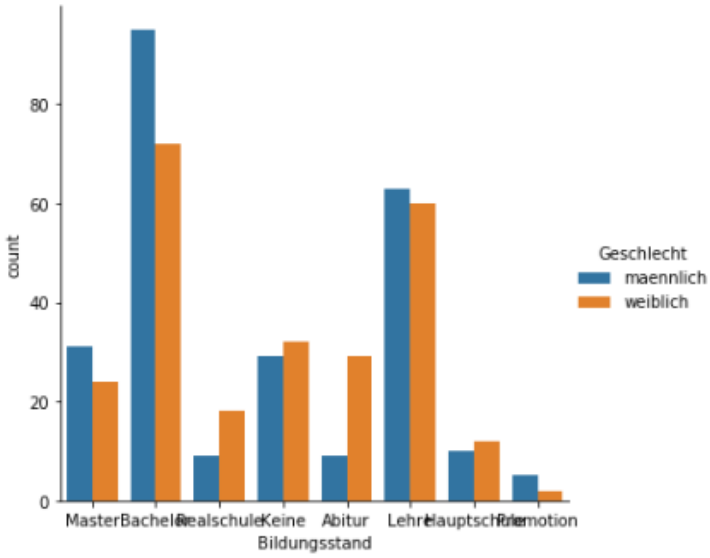
1. Unsere Zielvariable ist der Versicherungstarif (A, B, C oder abgelehnt)
2. Die einzigen Variablen, welche unabhängig von allen anderen Variablen sind und somit keine Elternknoten im Graph haben werden, sind Alter und Geschlecht.

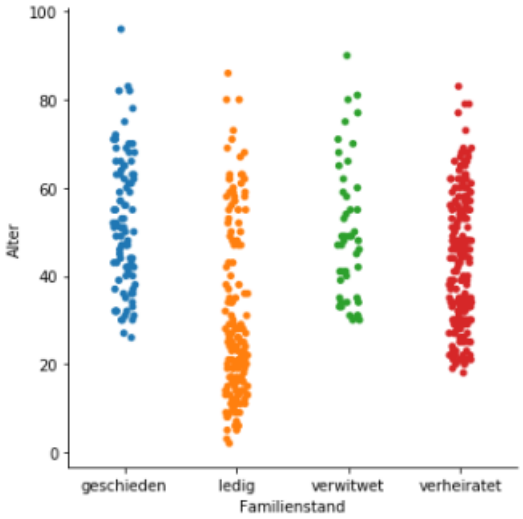
Somit ergibt sich der folgende Graph:

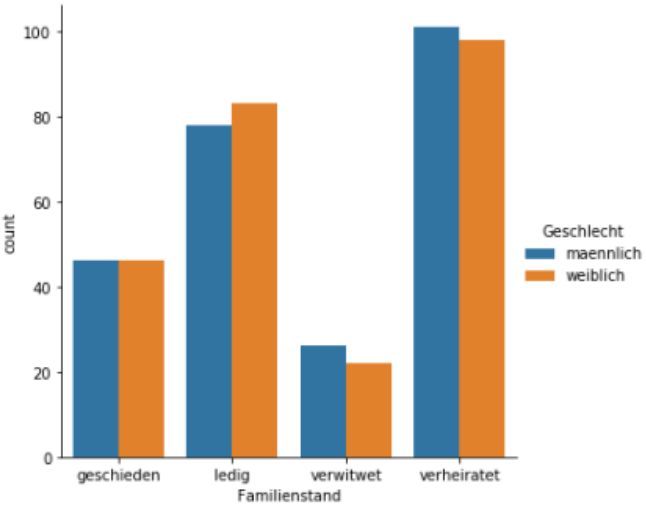


Nachfolgend untersuchen wir mithilfe von Datenvisualiserungstechniken, wie sich das Alter und Geschlecht auf die verbleibenden Variablen auswirkt. Diese können leider nicht in der Konsolenanwendung gezeigt werden. Besteht Interesse die Graphen selbst zu erzeugen empfehlen wir das Jupyter Notebook.

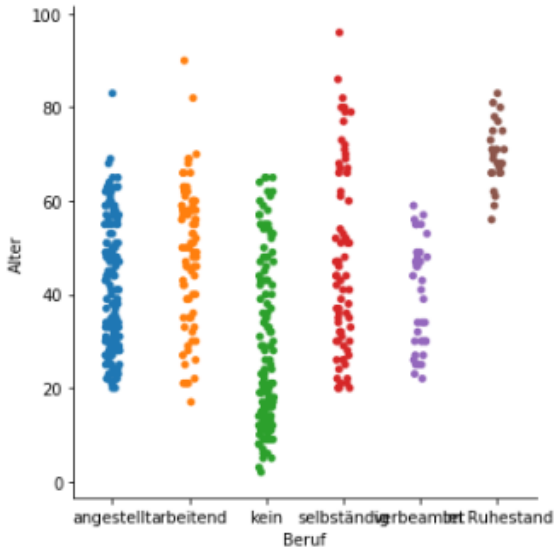


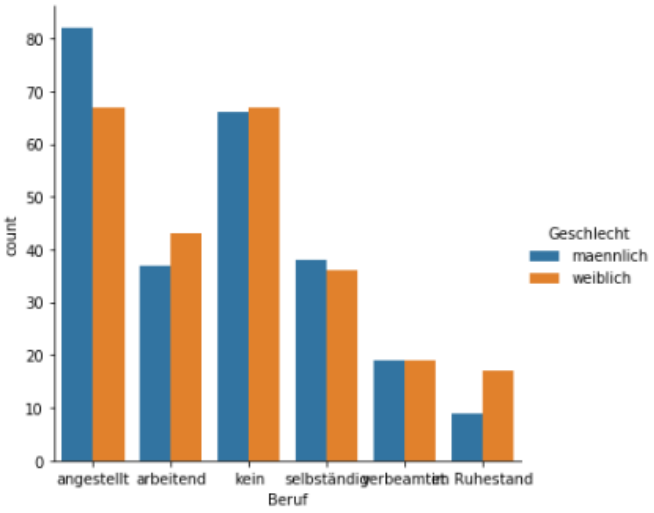


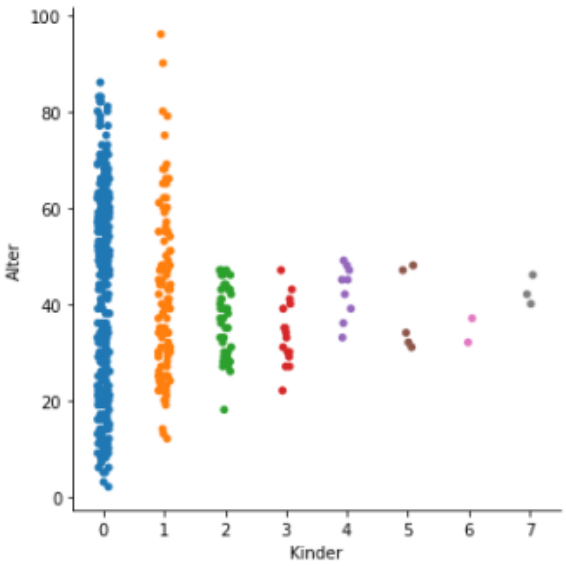


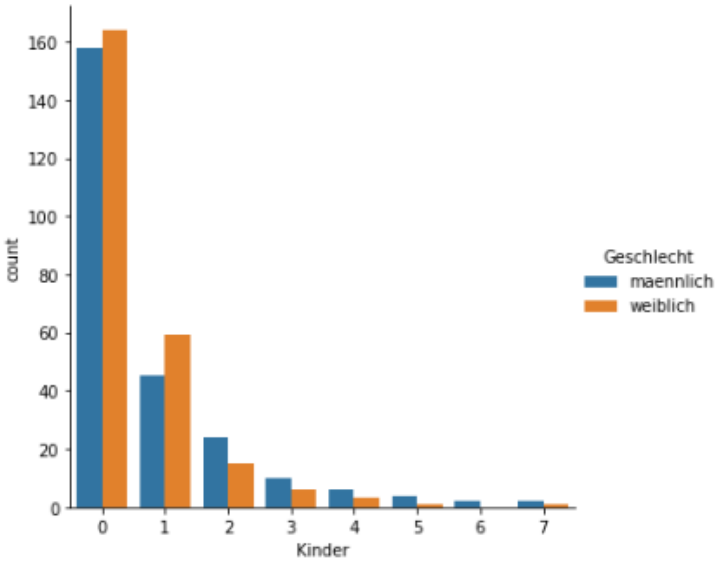


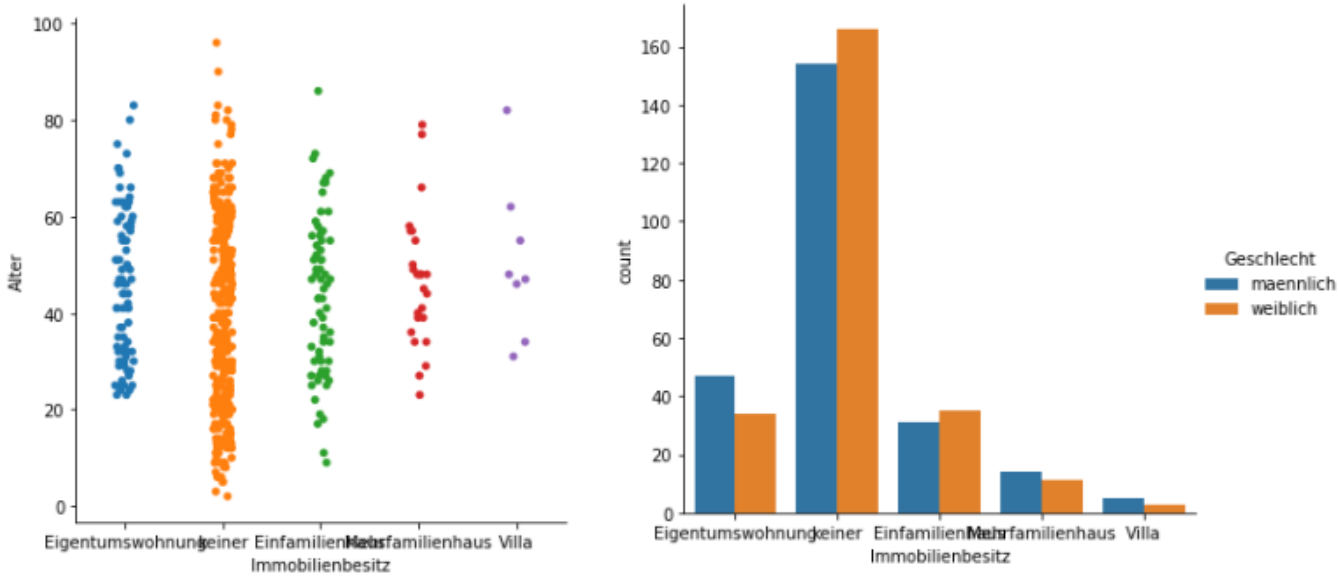
Design des Bayesian Networks (Graph)











Aus den obigen Plots lesen wir folgende Dinge ab:

1. Die Verteilungen der einzelnen Kategorien sind stark geprägt durch das Alter der Personen. Wir gehen davon aus, dass das Alter aufgrund der zugrundeliegenden Biologie und gesetzlichen Normen der direkte Elternknoten von jedem einzelnen anderen Knoten ist.

2. Insbesondere können wir beim Alter mehrere 'Levels' erkennen: Personen unter 20 Jahre und Personen über 65 Jahre können jeweils gesondert betrachtet werden.

3. Das Geschlecht macht meist nur einen geringen Unterschied aus. Allerdings können wir auch hier eine leicht ungleichmäßige Verteilung feststellen, insbesondere bei Bildungsstand, Immobilienbesitz und Beruf.

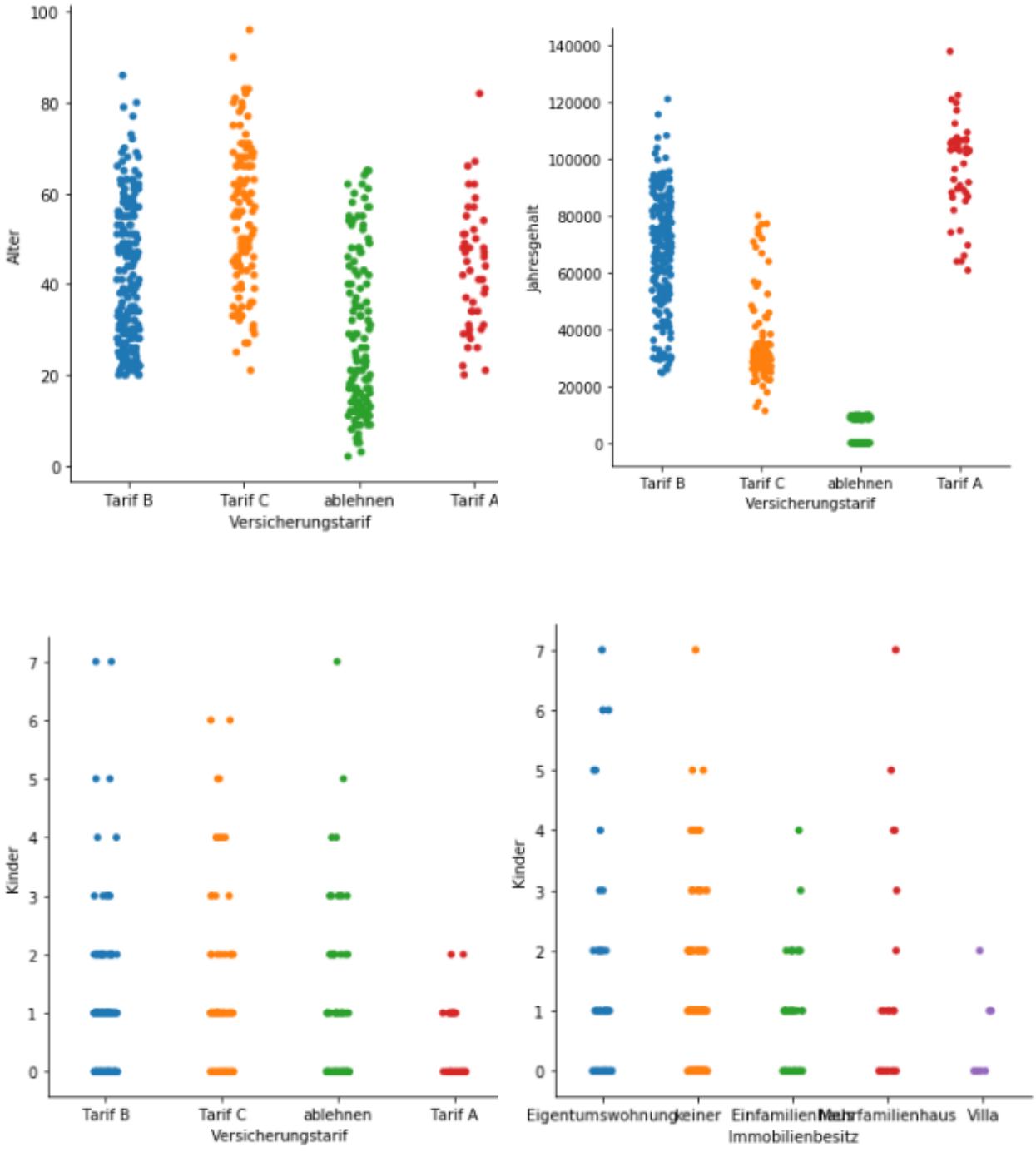
Weiterhin treffen wir folgende Annahmen:

1. Ein weiterer Elternknoten von Beruf ist der Bildungsstand der Person.

2. Die Elternknoten von Gehalt sind neben dem Alter der Bildungsstand und der Beruf der Person. Wir stellen ebenfalls einen leichten Gender Pay Gap fest, können jedoch nicht ausschließen, ob dieser direkt durch das Geschlecht oder bereits durch die ungleichmäßigen Verteilungen beim Bildungsstand und Beruf entstanden ist.

3. Ein weiterer Elternknoten von Immobilienbesitz ist das Jahresgehalt.

Nachfolgend wollen wir bestimmen, welche Variablen sich auf den Tarif auswirkt. Zunächst betrachten wir die numerischen Variablen:

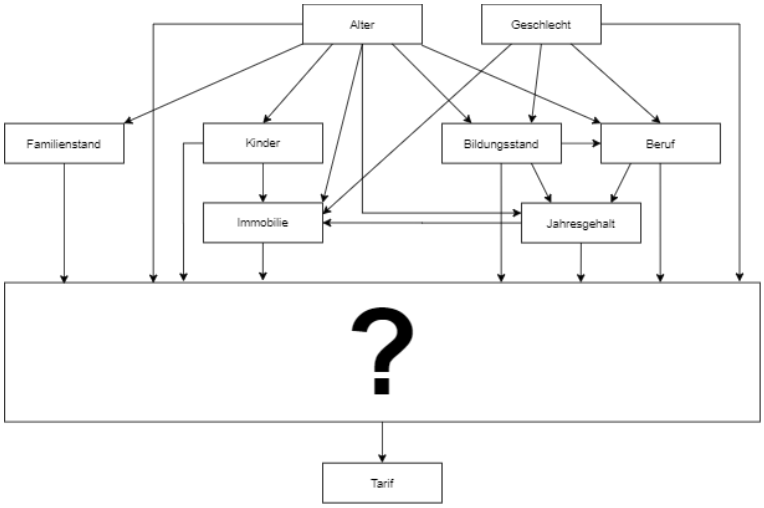


Nun können wir aus den vier Plots folgende Dinge ablesen:

1. Niemand unter 20 Jahre erhält einen Tarif (alle unter 20 werden abgelehnt).
2. Niemand mit einem Gehalt unter 10000 erhält einen Tarif und nur Personen mit einem hohen Gehalt erhalten Tarif A.

3. Niemand mit mehr als 2 Kindern erhält Tarif A. Wir denken jedoch nicht, dass die Kinderanzahl ausschlaggebend für den Tarif ist, sondern der Besitz einer Immobilie. Der letzte Plot zeigt, dass große Ähnlichkeiten zwischen Kinder-vs.-Tarif und Kinder-vs.-Immobilienbesitz bestehen.

Wir schließen daraus, dass das Alter und Gehalt direkte Elternknoten für den Tarif sind, Kinder jedoch nicht. Wir nehmen außerdem Kinder als zusätzlichen Elternknoten für Immobilien auf, da diese Graphen beinahe gleich sind. Es ergibt sich folgender Graph:



Für die restlichen (kategorischen) Variablen berechnen wir die bedingte Wahrscheinlichkeit, einen der Tarife zu bekommen, gegeben dass jeweils eine der restlichen Variablen eintritt. Dadurch soll verhindert werden, dass wir einen Elternknoten für Tarife aufnehmen, der eigentlich besser nur indirekt wirkt (Großvater/Urgroßvater/...). Für die kategorischen Variablen benutzen wir one-hot encoding, sodass für jede Kategorie eine einzelne True/False Spalte erzeugt wird. Diese ist im Code und im Jupyter Notebook zu finden.
$P(A|B) = \frac{P(A \cup B)}{P(B)}$
Ereignis A ist unsere Zielvariable, also Tarif A, Tarif B, Tarif C, oder ablehnen. Ereignis B sind unsere unabhängigen Variablen; für's erste wir nehmen hierfür alle categorical Variablen.
Wir iterieren über alle Zielvariablen und unabhängigen Variablen, und berechnen für jede Kombination die bedingte Wahrscheinlichkeit. Ergebnisse dieser Berechnung können im Programm gefunden werden. Wir testen mit folgender Formel, ob manche der Zielereignisse unabhängig (oder nur schwach abhängig) von Inputvariablen sind und filtern diese

Inputvariablen heraus. Diese Inputvariablen können nicht mehr direkte Elternknoten für den Tarif werden.

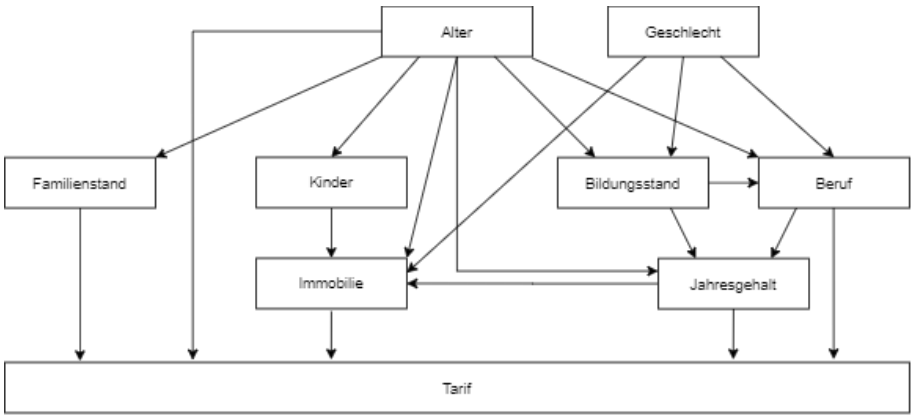
$P(A|B) \approx P(A)$

Oder etwas präziser: wir definieren die Ereignisse A und B als unabhängig ("nicht genug abhängig"), falls $0.8 < \frac{P(A|B)}{P(A)} < 1.25$

Die Ergebnisse der Berechnung können im Code oder im Jupyter Notebook gefunden werden. Aus diesen Ergebnissen ziehen wir folgende Schlüsse:

1. Die Tarife sind unabhängig vom Geschlecht. Somit ist Geschlecht kein direkter Elternknoten vom Tarif.
2. Manche Tarife sind teilweise unabhängig von Familienstand und Bildungsstand. Wir entscheiden uns dazu, den Familienstand als Elternknoten vom Tarif zu behalten; den Bildungsstand aber nicht als Elternknoten aufzunehmen, da der Bildungsstand noch indirekt über Gehalt und Beruf mit dem Tarif verwandt ist.
3. Alle Tarife sind abhängig von den kategorischen Variablen Immobilienbesitz und Beruf, sodass wir diese zwei Knoten als Elternknoten einbauen.

Es ergibt sich folgender finaler Graph:



3 Berechnen der CPTs

Mit der Funktion P() in der Klasse Vertex können wir sehr einfach die bedingten Wahrscheinlichkeiten für alle Nodes ausrechnen (CPT). Dafür haben wir die Klasse CPTCalculatro erstellt. Dieser führt die Berechnungen der CPTs durch. Hier als Beispiel die Berechnung des Alter CPTs:

```
self.age = Vertex(self.data 'Alter',
['Alter < 20', 'Alter >=20 & Alter < 65', 'Alter>=65'])
self.age.update_cpt()
self.age.print_cpt()
```

Das Geschlecht CPT wird ähnlich zum Alter CPT berechnet, allerdings ohne den Elternknoten. Den Familienstand CPT berechnen wir durch die bedingte Wahrscheinlichkeit für Familienstand und Alter: P(Familienstand|Alter). Bei der Anzahl der Kinder wird in zwei Gruppen unterteilt. Die Grenze liegt dabei bei drei Kindern. Diese Grenze begründen wir dadurch, dass Personen mit $>= 3$ Kindern keine Villa besitzen und auch keinen Tarif A bekommen. Den Bildungsstand CPT berechnen wird durch die bedingte Wahrscheinlichkeit des Bildungsstands gegen Alter und Geschlecht. Der Beruf CPT wird durch die bedingte Wahrscheinlichkeit Beruf gegen Alter, Geschlecht und Bildungsstand berechnet. Beim Jahresgehalt CPT erstellen wir erneut mehrere diskrete Gruppen basierend auf den Erkenntnissen im ersten Teil. Diese Gruppen sind:

1. Personen mit < 10.000
2. Personen mit $>= 10.000$ und < 60.000
3. Personen mit $>= 60.000$ und < 80.000
4. Personen mit $>= 80.000$ und < 100.000
5. Personen mit > 100.000

Schlussendlich werden die CPTs von Immobilienbesitz und Tarif basierend auf den bedingten Wahrscheinlichkeiten der vorangegangenen Kategorien berechnet - genaueres im Code zu finden.
Um die Ergebnisse zu überprüfen haben wir eine Testing-Funktion geschrieben, die Stichprobenartig verschiedene CPTs prüft. Die Ergebnisse stimmen mit den „manuell“ gerechneten Ergebnissen überein. Unsere Tests beschränken sich auf Alter, Familienstand und Bildungsstand.

4 Implementierung der Bayesian Networks

Für das Bayesian Network benutzen wir die Bibliothek pomegranate. Zuerst müssen für alle Variablen die möglichen Ereignisse/Ausgänge und dazugehörige (bedingte) Wahrscheinlichkeiten angegeben. Danach werden entsprechende Nodes erzeugt, einem Modell hinzugefügt und miteinander verbunden. Dies kann in der BayesNetwork Klasse betrachtet werden. Dort ist auch das folgende Beispiele für die Verwendung zu finden:

```
model.predict([['Geschlecht=="maennlich"', None, 'Alter <
20', None, None, None, None, None, None]])

model.predict_proba([['Geschlecht=="maennlich"',
'Familienstand=="verheiratet"', 'Alter >=20 & Alter < 65',
'Kinder < 3','Bildungsstand=="Master"', 'Beruf=="selbständig"',
'Jahresgehalt >= 100000', 'Immobilienbesitz=="Einfamilienhaus"', None]])
```

5 Test, Evaluation und Fazit

Wir haben eine Personen Klasse für eine Testperson integriert und dabei im Hintergrund eine Tabelle liegen. Der Beispielcode kann die fehlenden Werte der Personen vorhersagen.

ID	Geschlecht	Familienstand	Alter	Kinder
0	männlich	verheiratet	20 <= x < 65	< 3
1	weiblich	ledig	20 <= x < 65	< 3
2	männlich	verwitwet	20 <= x < 65	< 3
3	weiblich	geschieden	20 <= x < 65	< 3
4	männlich	ledig	x < 20	< 3

Tabelle 1: Vorhergesagte Werte - Teil 1

Bildungsstand	**Beruf**	**Jahresgehalt**	**Immobilienbesitz**	**Tarif**
Master	angestellt	80.000 <= x < 100.000	Eigentumswohnung	Tarif B
Lehre	arbeitend	10.000 <= x < 60.000	keine	Tarif C
Master	angestellt	60.000 <= x < 80.000	keine	Tarif B
Lehre	kein	x < 10.000	keine	abgelehnt
keine	kein	x < 10.000	keine	abgelehnt

Tabelle 2: Vorhergesagte Werte - Teil 2

Mithilfe von Pomegranate und verschiedener Hilfsklassen konnten wir ein Bayesian Network entwickeln, das fehlende Variablen vorhersagt. Basierend auf Erkenntnissen, die durch Datenvisualisierung und Unabhängigkeitstests gewonnen wurden, haben wir zunächst die Beziehungen der Variablen untersucht und einen passenden Graph entwickelt. Daraufhin konnten wir die Conditional Probability Tables automatisiert berechnen und das Netzwerk implementieren und testen.
Aus dieser ersten Testrunde ziehen wir folgende vorläufigen Schlüsse (eine abschließende Bewertung des Bayesian Networks würde jedoch eine deutlich umfangreicherer und systematischere Untersuchung voraussetzen):

1. Unsere Vorhersage stimmt in den meisten Fällen mit den echten Werten überein, oder liegt zumindest nur knapp daneben.
2. Manche Variablen sind schwerer vorherzusagen als andere, da sie wahrscheinlich einen weniger großen Einfluss auf die restlichen Variablen haben. Falls eine Person bspw. unter 10 Jahre alt ist, lassen sich fast alle anderen Variablen sehr zuverlässig vorhersagen, während bei einer Person mit 50 Jahren nur schwer vorhergesagt werden kann, ob sie geschieden ist oder verheiratet/ledig/verwitwet.
3. Je mehr Daten einer Person fehlen, desto ungenauer werden die Vorhersagen.

Das Design des Graphs ist maßgeblich für die weitere Berechnung und Implementierung der Conditional Probability Tables und des Bayesian Netzwerk. Es gibt einige Entscheidungen, über die man sicherlich streiten kann:

- Sollte Kinder ein Elternknoten von Immobilienbesitz sein ? (-> muss man sich als Erwachsener entscheiden, ob man ein Kind oder ein Haus will?)
- Ist das Geschlecht nicht doch direkt mitverantwortlich für den Tarif (Erfahrungen aus der echten Welt würden dies wohl bestätigen)?

- Ab wann sind Ereignisse unabhängig? Ist $0.8 < \frac{P(A|B)}{P(A)} < 1.25$ ein zu hoher Schwellwert für stochastische Abhängigkeit?
- Wurden die numerischen Variablen (Alter, Kinder, Gehalt) in zu große Gruppen eingeteilt? Sollte man hier feinere Abstufungen implementieren oder gar eine kontinuierliche Eingabe/Vorhersage ermöglichen?
- Sollte die CPT an Stellen, an denen der Nenner $\frac{P(A|B)}{P(B)}$ null war (und somit die bedingte Wahrscheinlichkeit aufgrund von mangelnden Beispielen nicht berechnet werden konnte), mit Nullen aufgefüllt werden oder sollten diese Zeilen von der Tabelle gelöscht werden?
- ...

Für viele dieser Designfragen versuchten wir, eine möglichst klare Antwort aus den Daten ablesen zu können. Wir sind zuversichtlich, ein passendes Netzwerk implementiert zu haben.