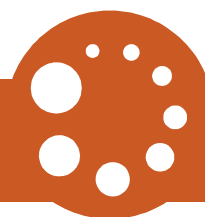




Plausi++: Automatische Plausibilitätsprüfung der Qualität und Zuverlässigkeit von Administrativ- und Umfragedaten

Pilotprojekt der Dateninnovationsstrategie:
Automatische Entdeckung möglicher Fehlermuster
anhand Machine Learning Algorithmen

EXPERIMENTAL STATISTICS



Neuchâtel, 2019

Herausgeber: Bundesamt für Statistik (BFS)
Redaktion: Dr. Christian Ruiz
Inhalt: Dr. Christine Ammann Tschopp, Elisabeth Kuhn, Laurent Inversin, Dr. Mehmet Aksözen, Stefan Rüber
Themenbereich: 00 Statistische Grundlagen
Originaltext: Deutsch
Übersetzung: Sprachdienste BFS

Layoutkonzept: Sektion DIAM
Download: www.statistik.ch
Copyright: BFS, Neuchâtel 2019
Wiedergabe unter Angabe der Quelle
für nichtkommerzielle Nutzung gestattet

Ein herzliches Dankeschön an
Prof. Dr. Diego Kuonen für die tatkräftige Unterstützung
Dr. Vera Herrmann für die Unterstützung als Bereichsleitende

Inhaltsverzeichnis

1	Executive Summary	1
1.1	Ausgangslage	1
1.2	Problematik	1
1.3	Zielsetzung	1
1.4	Zielerreichung	2
1.5	Projektorganisation	2
2	Projektbewertung nach CRISP-DM	3
2.1	Business Understanding Overview	3
2.2	Data Understanding Overview	3
2.3	Data Preprocessing Overview	4
2.4	Modeling Overview	4
2.5	Evaluation Overview	4
2.6	Deployment Overview	5
3	Lessons learned	5
4	Ausblick	6
A.	Anhang: Anwendungsbeispiel	0
B.	Anhang: Die ausführliche Dokumentation	3

1 Executive Summary

1.1 Ausgangslage

Plausibilitätsprüfungen («Plausis») werden durchgeführt, um die Qualität und Zuverlässigkeit von Administrativdaten und Umfragedaten zu überprüfen. Daten, die dabei entweder eindeutig falsch oder zumindest fraglich erscheinen, werden an die Datenlieferanten mit Bitte um Korrektur oder Kommentar zurückgeschickt. Bisher wurden solche Plausibilitätsprüfungen auf drei komplementäre Arten durchgeführt:

- 1) **Manuelle Regeln:** Dabei schauen Mitarbeitende die Datensets durch und überprüfen sowohl inhaltliche wie auch formelle Fehler. Der Prozess ist arbeitsintensiv.
- 2) **‘Harte’ automatische Regeln:** Sie betreffen meist formelle Elemente wie die Respektierung gültiger Liefercodes, das Ausfüllen leerer Datenfelder oder die Einhaltung von klaren numerischen Grenzen. Für inhaltliche Elemente sind sie weniger gut geeignet.
- 3) **‘Weiche’ automatische Regeln:** Diese Art von automatischen Regeln setzt Grenzwerte aufgrund von Erfahrungswerten und Durchschnittswerten. Sie sind approximativ und können von den Datenlieferanten in den meisten Fällen überschrieben werden.

1.2 Problematik

Kriterium 1: Zeitdauer. Es gibt zwei zeitbezogene Probleme. Einerseits dauert insbesondere die manuelle Plausi lange, da meist mehrere Kontrolldurchgänge nötig sind. Andererseits gibt es auch mit den automatischen Plausis ein Zeitproblem, da das «Hin und Her» zwischen den Datenlieferanten und dem BFS sich über mehrere Wochen erstrecken kann.

Kriterium 2: Datenqualität. Die Datenlieferanten müssen die automatischen Regeln des BFS nur in den ‘harten’ Fällen zwingend umsetzen. In den ‘weichen’ Fällen gibt es einen Ermessensspielraum und die Datenlieferanten können die Korrekturvorschläge ignorieren. Die Fachverantwortlichen meinen, dass es sowohl Datenlieferanten gibt, die unsere Fehlervorschläge gewissenhaft durcharbeiten, wie auch solche, die nicht auf diese Vorschläge eingehen.

Kriterium 3: Inhaltliche Fehler. Eine automatische Plausi ist stärker auf formelle Fehler ausgerichtet. Das Finden von inhaltlichen Fehlern ist tendenziell zeitintensiv. Das betrifft sowohl Individualdaten wie auch systematische Fehler von gewissen Datenlieferanten. Solche systematischen Fehler benötigen eine detailliertere Analyse durch die Mitarbeitenden.

1.3 Zielsetzung

Das Pilotprojekt zur Erstellung einer neuen Plausi («Plausi++») beabsichtigte diese drei Probleme anzugehen und die Plausibilitätsprüfungen im BFS anhand von Machine Learning (ML) Algorithmen zu erweitern und zu beschleunigen, bei gleichzeitiger Verbesserung der Datenqualität.

Ziel 1: Die Ermittlung von Angaben mit weniger plausiblen Inhalten erfolgt in zwei Schritten. Zunächst wird ein ML-Algorithmus entwickelt, der den Zielvariablen eine Fehlerwahrscheinlichkeit zuweist. Das ist das Herzstück, da dies der Mechanismus ist, um mögliche Fehler zu entdecken.

Ziel 2: Im zweiten Schritt wird die Begründung des möglichen Fehlers ermittelt und eine im besten Falle menschlich verständliche und möglichst überzeugende Rückmeldung für die Datenlieferanten generiert.

Ziel 3: Die Dokumentation des Projekts ist so angelegt, dass eine Skalierbarkeit in möglichst vielen Sektionen des BFS ermöglicht werden soll, und nicht nur spezifisch für BILD-P und BILD-S.

Ziel 4: Die Produktionsintegration spielte in den Überlegungen zur Projektentwicklung ebenfalls eine wichtige Rolle, bleibt aber durch die technischen Vorgaben leider nur theoretisch.

1.4 Zielerreichung

- ✓ **Ziel 1 ML-Algorithmus, Kriterium Datenqualität auf individueller Ebene:** Die Benutzung des ML-Algorithmus konnte die Datenqualität deutlich verbessern: Im Baseline-Vergleich zur alten Plausi konnte die neue Plausi **29% mehr Fehler entdecken**. Der ML-Algorithmus funktioniert mit einer **Genauigkeit von über 94%**.
- ✓ **Ziel 1 ML-Algorithmus, Kriterium Ergänzung:** Es handelt sich bei der neuen Plausi **ausschliesslich um inhaltliche Fehler**. Die neue Plausi **ergänzt sich somit** mit der alten Plausi, bei der nur 21% der Regeln inhaltlicher Natur sind und 79% **formeller Art**¹. **Die neue Plausi ersetzt somit nicht die alte Plausi, da eine Symbiose beider Plausis sowohl inhaltliche wie auch formelle Fehler abdeckt.**
- ✓ **Ziel 1 ML-Algorithmus, Kriterium Datenqualität auf aggregierter Ebene:** Ein weiteres Indiz für die erhöhte Datenqualität ist **die Ermittlung von systematischen Fehlern auf aggregierter Ebene**. Wie im Anhang detaillierter beschrieben, **findet die neue Plausi insgesamt mehr systematische Fehler**. An einem Beispiel im separaten Anhang B.5.1 wird skizziert, **wie doppelt so viele**² solcher systematischen Fehler gefunden wurden.
- ✓ **Ziel 1 ML-Algorithmus, Kriterium Beschleunigung:** Es gibt im «supervised ML» (überwachtes ML) zwei Phasen. In der Anwendungsphase, **braucht die neue Plausi nur wenige Minuten**, um ausgeführt zu werden. Zusätzlich müsste sich somit auch eine Zeitersparnis durch die Reduktion der «Hin und Her» mit den Datenlieferanten ergeben, was aber nur nach Implementation in der Produktionsumgebung getestet werden kann.
Aus Ressourcengründen wurde ein Datenset mit einer automatischen Plausi ausgewählt. Die grösstmögliche Zeitersparnis ist aber mit der Ergänzung einer arbeitsintensiven manuellen Plausi zu erwarten. Dies müsste, zusammen mit der Unterstützung der Verantwortlichen, bei einem geeigneteren Datenset getestet werden.
- ✓ **Ziel 2 Feedback-Mechanismus:** Ein automatisierter Feedback-Mechanismus erklärt das Resultat des ML-Algorithmus einem Datenlieferanten auf eine menschlich nachvollziehbare Weise. Eine solche Antwort, welche auf die mögliche Problemvariable hinweist (Beispielsweise Alter oder Personalkategorie), konnte mit einem Verfahren vollständig implementiert werden. Weitere Ansätze könnten noch getestet werden.
- ✓ **Ziel 3 Dokumentation:** Die Dokumentation ist in drei Teilen aufgebaut, mit dem Ziel einer Skalierbarkeit. Der erste Teil ist ein Anwendungsbeispiel, der zweite die ausführliche Dokumentation und der dritte Teil ist ein kommentierter Computercode, damit andere Personen diesen nachvollziehen und adaptieren können.
- ✓ **Ziel 4 Produktionsintegration:** Das Pilotprojekt hat die Machbarkeit einer Produktionsintegration in den möglichen Grenzen ausgewertet. Es gibt natürlich viele Unbekannte wie die schlussendliche IT-Infrastruktur-Lösung und andere Elemente, die nicht Bestandteil des Pilotprojekts sind. Wir haben dennoch Überlegungen aufgezeichnet und denken, dass die neue Plausi gut in den Prozess der alten Plausi integriert werden kann.

1.5 Projektorganisation

Um die oben genannten Ziele zu erreichen, haben sechs wissenschaftliche Mitarbeitende aus den Sektionen BILD-P und BILD-S zusammengearbeitet. Geleitet wurde das Team von Dr. Christian Ruiz. Der Projektleiter hat dabei als einziger eine fixe wöchentliche Arbeitszeit von 20% zur Verfügung bekommen. Die anderen fünf Mitarbeitenden haben deren Teilnahme nach deren Zeitverfügbarkeit, abhängig von deren regulären Arbeitsaufgaben, eingeplant. **Das hat die Organisation des Projekts stark erschwert, da man nicht mit festen Zeitkontingenten oder festen Deadlines arbeiten konnte.** Die Zusammenarbeit hat sich aber positiv gestaltet und das Knowhow und die statistische Affinität der wissenschaftlichen Mitarbeitenden hat dazu geführt, dass die ML-Elemente relativ rasch assimiliert wurden.

¹ Beispielsweise die Gültigkeit verwendeter BFS-Codes, keine fehlenden Datenfelder, usw.

² Das darf aber auch nicht überbewertet werden, da es von 2 auf 4 anstieg. Der Trend ist aber klar.

Die Benutzung von «CRISP-DM» als Prozessmodell für die Dateninnovation hat uns geholfen den Analyseprozess möglichst agil zu gestalten. Besonders die Datenvorbereitungs-, Modellierung- und Evaluationsphasen von CRISP-DM bildeten einen Zyklus, den wir mehrmals durchlaufen haben. Die Evaluation brachte Erkenntnisse, wie man die Vorphasen kontinuierlich verbessern kann. Das führte zu einem agilen Projektmanagement und laufender Adjustierung von gewissen Elementen. Es wurde möglichst flexibel auf Confluence gearbeitet, wobei der Projektleiter mit einer «Kanban»-inspirierten Methode aus dem «Lean Management» Ansatz, die Aufgaben geleitet und verteilt hat. Die Benutzung Projektmanagement-Methoden wie «SCRUM» wurde ebenfalls in Betracht gezogen, aber aufgrund der oben genannten fehlenden zeitlichen Kontinuität der Arbeitsressourcen nicht durchgeführt, da dies die Institutionalisierung täglicher Meetings bedeutet, und eine intensive und ununterbrochene Kooperation benötigt. Im Falle einer Fortsetzung des Pilotprojekts und der Umsetzung der ergänzenden Elemente und insbesondere im Falle einer Implementierung von Plausi++ in die Produktionsumgebung, bedarf es einer Erhöhung und klarer Zuteilung von Ressourcen (für alle Beteiligten).

2 Projektbewertung nach CRISP-DM

2.1 Business Understanding Overview

Damit das Ergebnis des Pilotprojekts die Bedürfnisse der Fachexperten nachfrageorientiert erfüllt, wurden diese von Anfang an und bei Bedarf hinzugezogen, wenn sie nicht sogar Teil des Teams waren. Damit sollte auch sichergestellt werden, dass die Datenlieferanten ein Feedback erhalten, das sie nutzen können. Als interne Baseline dienten von den bisherigen Kontrollen abgeleitete Indikatoren.

Regelmässige Risikoanalysen begleiteten das Projekt, um zeitnah auf Gefährdungen des Projekts zu reagieren. Es wurden vier grössere Risiken ermittelt und gehandhabt (Ressourcen, Inhalt, Produktion und Entwicklungsumgebung). Die zeitlichen Ressourcen der Mitarbeitenden basieren auf freiem Engagement und sind dementsprechend Schwankungen unterworfen und waren schwer planbar. Die geplante inhaltliche Rückmeldung an die Datenlieferanten ist modelltechnisch anspruchsvoll und nahe am «State-of-the-Science», so dass teilweise prototypisch vorhandene Anwendungen für das Projekt adaptiert wurden. Anfänglich wurden die Bereitstellungsschwierigkeiten der Entwicklungsumgebung als grosses Risiko für eine Produktionsintegration betrachtet. Bei der Produktionsumsetzung sind wir nun zuversichtlicher als am Anfang, dass eine technische Lösung auch praktisch möglich ist.

Je weiter das Projekt vorangeschritten ist, desto mehr interessierte Anfragen kamen aus anderen Sektionen, ob man die neue Plausi nicht auch für deren Daten verwenden könnte. Wir waren mit diesen Sektionen im Gespräch, konnten aber aus Ressourcengründen keine weiteren Tests machen. An dieser Stelle ist es wichtig zu betonen, dass der im Projekt verwendete «supervised ML» Ansatz auch für andere Zwecke ausserhalb von Plausibilitätskontrollen verwendet werden kann. Das Potential ist im BFS durchaus vorhanden.

2.2 Data Understanding Overview

Das Team hat zunächst verschiedene Datensets analysiert, um eine Reihenfolge der jeweiligen Priorität zu bestimmen. Kriterien zur Bestimmung der Priorität waren 'relevanter Informationsgehalt', 'Simplizität', 'Qualität des Datensets', 'Verknüpfungsfragen' und 'andere Stakeholder'. Drei Datensets kamen im Pilotprojekt in die engere Auswahl. Aufgrund der Ressourcen lag der Fokus auf dem Datenset mit der höchsten Priorität (Hochschulpersonal). Das Datenset mit der zweithöchsten Priorität (Statistik der Lernenden) muss mittelfristig angedacht werden, da insbesondere die Anwendungsrelevanz sehr hoch ist. Leider war eine Teilnahme des Fachexperten für das Datenset mit der dritten Priorität (Schulpersonal) zeitlich nicht möglich. Weitere Arbeiten hierzu wären sinnvoll und würden zusätzliche Erkenntnisse zu Plausi++ bedeuten.

Wir möchten betonen, dass die **Wahl des Datensets für dieses experimentelle Pilotprojekt sehr wichtig**

war und sich positiv auf die Zielerreichung auswirkte; ebenso die Unterstützung durch die Bereichsleitende. Wie bereits weiter oben erwähnt wurde, wurde kein Datenset ausgewählt, bei der momentan die Plausi manuell durchgeführt wird. Die Wahl eines solchen Datensets hätte die Ressourcen dieses kurzen Pilotprojekts überstiegen.

2.3 Data Preprocessing Overview

Die Vorhersagekraft von ML-Algorithmen verbessert sich, wenn Variablen mit zusätzlichen Informationen ergänzt werden. Solche Informationen können aus der Kombination von bestehenden Variablen erzeugt werden oder können aus intrinsischem Wissen der Fachexperten abgeleitet werden («feature engineering»). Beide Formen der Datenanreicherung sind iterativ erfolgt. Diese Kommunikation hilft darüber hinaus auch bei der Formatierung der Daten, so dass möglichst viele Informationen in den Daten vorhanden sind. Während wir mit sechs Variablen begonnen haben, konnten wir durch Verknüpfungen, durch das Expertenwissen und durch statistische Methoden, die Anzahl der Variablen auf etwa 1'000 erhöhen.

2.4 Modeling Overview

Die Modeling-Phase besteht bei uns, den beiden Hauptzielen folgend, aus zwei Teilen. Zunächst wurde der ML-Algorithmus implementiert (Ziel 1). Dabei haben wir uns insbesondere auf fortgeschrittene Baumalgorithmen wie «Gradient Boosting Machines» (GBM) und «Random Forests» (RF) limitiert. Diese Entscheidung besteht aus fünf Anforderungen, die diese Algorithmen erfüllen: Verwendung von «mixed-mode» Variablen; eine «variable importance» muss berechnet werden können; Ausreisser sowie irrelevante Inputvariablen dürfen keinen starken Effekt haben; Datenlücken sollten kein Problem sein.

Bei den ersten Tests fanden wir, dass der GBM Algorithmus bessere Werte in der Vorhersage unseres Datensets erzielt als der RF Algorithmus. Die Vorhersage konnte durch «tuning» der Hyperparameter aber auch durch andere technische Verbesserungen erhöht werden.

Im zweiten Teil geht es um den Feedback-Mechanismus (Ziel 2). Hier versuchen wir die «blackbox» des ML-Algorithmus zu öffnen und den Datenlieferanten eine Erklärung zu den ermittelten Fehlerwerten zu geben.

Um unser Ziel zu erreichen, haben wir zwei Ansätze näher verfolgt:

- «Ceteris-Paribus»: Hier handelt es sich um ein «Was wäre, wenn...»-Szenario. Wenn eine bestimmte Variable einen anderen Wert hätte, würde dann die Vorhersage des prädiktiven ML-Algorithmus (aus Ziel 1) mit der gelieferten Zielvariablen übereinstimmen?
- «Local Interpretable Model-Agnostic Explanations» (LIME): Ein zusätzliches lokales erklärendes Modell (nicht ein prädiktives Modell wie bei Ziel 1) sucht Variablen, die bei der Vorhersage der Zielvariablen eine besonders wichtige Rolle spielen. Wenn der ML-Algorithmus einen Fehler entdeckt, können somit auch andere Ursachen in Betracht gezogen werden.

Dabei wurde das Ziel erreicht und eine automatische Rückmeldung an den Datenlieferanten konstruiert, die ihm dabei helfen soll, die (mögliche) Fehlerquelle zu eruieren.

2.5 Evaluation Overview

Wir haben als Anwendungsbeispiel die Daten des Hochschulpersonals genommen (siehe auch Anhang A). Die Evaluation des Ziel 1 erfolgt durch Indikatoren wie der «prediction accuracy» (Vorhersagegenauigkeit), bei der vorhergesagte und gelieferte Werte verglichen werden. Hier konnten wir eine «prediction accuracy» von über 94% erreichen, ohne die möglichen Verbesserungsideen aus Zeitgründen vollständig auszuschöpfen. Beispielsweise konnten wir den «XGBoost» Algorithmus aus reinen Zeitgründen nicht versuchen und erhoffen uns durch diesen eine mögliche weitere Verbesserung. Im zweiten Ziel ist ein Feedback-Mechanismus («Ceteris-Paribus») implementiert, der aber noch weiter getestet werden muss – beispielsweise anhand der Reaktion eines echten Datenlieferanten, um zu sehen, ob das Resultat interpretierbar ist, und den gewünschten Effekt hat. Mit den vorliegenden Daten konnte dieses Feedback bei 60% des Personals mit einer abweichenden Personalkategorie zwischen der Datenlieferung und der Vorhersage (6% der Datensets) einen Erklärungs- bzw. Anpassungsvorschlag geben.

Ein zweiter Feedback-Mechanismus («LIME») ist momentan so implementiert, dass es ein technisches Feedback für die BFS-Mitarbeiter geben kann, aber noch nicht für den Datenlieferanten. Es gibt weitere Umsetzungsideen, die aus Zeitgründen noch nicht ausgeführt wurden. Ein Test mit Datenlieferanten wäre ebenfalls sinnvoll, um ein nachfrageorientiertes Produkt zu erstellen.

Wir konnten ebenfalls eine Baseline erstellen, bei der man die aktuelle Plausi mit der neuen vergleichen kann. Dies war keine simple Aufgabe und erforderte komplett neue Überlegungen, da ein solcher Vergleich bei der aktuellen Plausi nicht eingeplant war. Die folgende Tabelle zeigt eine Gegenüberstellung.

	Plausi++ findet keinen Fehler	Plausi++ findet Fehler
Alte Plausi findet keinen Fehler	75.1 %	5.6%
Alte Plausi findet Fehler	17.9%	1.4%

Tabelle: Aufteilung der entdeckten «Fehler» nach alter und neuer Plausi als Bruchteil aller Datenpunkte.

Die Tabelle zeigt, dass Plausi++ in 7% aller Datenpunkte einen inhaltlichen Fehler vermutet. In 1.4% aller Datenpunkte fanden sowohl die alte wie auch die neue Plausi einen Fehler – auch das ist ein Indiz, das sich die beiden Plausis gut ergänzen, da dieser Wert überraschend tief ist.

Dazu zeigt der Anhang A, wie auch mehr systematische Fehler gefunden werden können, die auf der Einsicht beruhen, dass es qualitative Erhebungsunterschiede zwischen den einzelnen Datenlieferanten gibt. Es gibt aber ein wichtiges methodisches Caveat, wenn es Fehler in den alten Daten gibt, die dadurch mittrainiert werden. Nach unseren Analysen ist aber die Performanz der Fehlerentdeckung dennoch sehr hoch. Wir vermuten aber, dass dies gewisse systematische Fehler kaschieren könnte.

2.6 Deployment Overview

Die Integration in die Produktionsumgebung ist durch die Rahmenbedingungen leider nicht möglich, da eine separate und unabhängige IT-Infrastruktur verwendet wird. Dennoch haben wir es als drittes Ziel angesehen, über die Integration nachzudenken, und planen unser Projekt auf eine Weise, um bei einer allfälligen Ausführung auf der BFS-Infrastruktur mit möglichst wenig Aufwand Plausi++ implementieren zu können. Die finalen IT-Schritte wurden daher bereits mit einigen zuständigen Stakeholdern diskutiert (Anhänge B.6 und B.7).

3 Lessons learned

Obwohl wir sehr wenig Zeit zur Verfügung hatten, konnten wir das Pilotprojekt erfolgreich durch die Ziellinie führen. Plausi++ kann inhaltliche Fehler automatisch erkennen und generiert ein automatisches Feedback an die Datenlieferanten. Gleichzeitig haben wir eine Produktionsintegration angedacht und halten diese unsererseits für realisierbar. Bezüglich Qualitätsmanagement, war die Erarbeitung einer systematischen Dokumentation und der Kommentierung des Codes essenziell: Somit könnten auch andere Mitarbeitende sich in die Arbeiten einlesen und gewisse Elemente Ihren Anforderungen anpassen.

Die Aneignung von neuartigen ML-Ansätzen der Mitarbeitenden hat sehr rasch funktioniert, was auch an der zugrundeliegenden statischen Ausbildung und Affinität liegt: «Machine Learning» wird nicht von ungefähr als «Statistical Learning» betitelt. Die Anwendung dieser Algorithmen liegt somit beim BFS am richtigen Ort.

Die Wahl des Datensets hatten wir uns gut überlegt, was Konsequenzen in der Zielerreichung hatte. Es gibt andere Datensets, die wahrscheinlich einen grösseren Effizienzzuwachs generieren könnten, aber auch mehr Zeitressourcen sowie mehr Unterstützung benötigt hätten.

Die Projektorganisation insbesondere eine Planung mit Deadlines und Vorgaben war sehr schwer durchzuführen, da der Projektleiter einerseits nur 20% zur Verfügung hatte, während die anderen Mitarbeitenden keine fixen Stellenprozente für das Pilotprojekt bekommen hatten. Die Kontinuität und ein konzentriertes Arbeiten an einer Sache war somit nicht durchgehend gegeben. Eine 'lessons learned' ist fixe Verantwortungen zu verteilen und durchgehend und möglichst Vollzeit an derselben Sache zu arbeiten.

Wir konnten das Ziel mit der Implementierung eines Feedback-Mechanismus erreichen. Wir befinden uns aber inmitten der Forschung und es erscheinen praktisch monatlich neue Möglichkeiten. Das Potential ist somit noch nicht ausgeschöpft und eine Verlängerung des Pilotprojekts sinnvoll.

Ein wichtiges Element war auch «on the job learning». Die Projektmitarbeitenden waren alle motiviert mehr zu lernen – sowohl über die ML-Algorithmen wie auch über die Programmiersprache R. Das Programm welches vom BFS angeboten wurde, wurde auch gut angenommen. Der Projektleiter hätte aber die Mitarbeitenden gerne selbst in mehr Elementen eingeführt – aber es fehlten ganz klar die Ressourcen, um beispielsweise projektinterne Workshops zu gewissen Elementen durchzuführen. Bei einem Projekt, welches mit mehr Ressourcen ausgestattet wird, könnte dies verbessert werden (gegebenenfalls sektionsübergreifend).

4 Ausblick

Es wurde zwischen den Sektionen BILD-P und BILD-S sehr produktiv zusammengearbeitet. Mittlerweile bekamen wir auch interessierte Anfragen von POP und REG, um herauszufinden, ob eine Plausi auch unter anderen Rahmenbedingungen anwendbar ist. Die Durchführung einer Plausibilitätskontrolle könnte somit auch in anderen Sektionen einen Mehrwert generieren. Darüber hinaus könnten sich auch weitere Anwendungsmöglichkeiten eines «supervised ML» Ansatzes – ausserhalb von Plausibilitätskontrollen - ergeben. Kurz: Die Nachfrage innerhalb des BFS ist vorhanden.

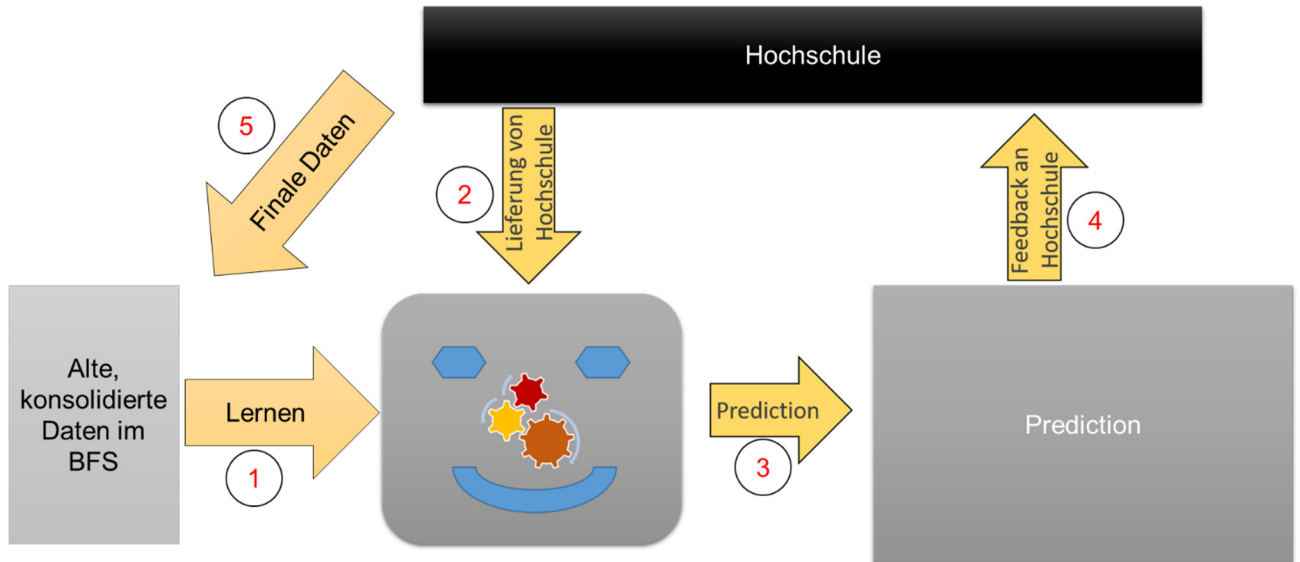
Der ML-Algorithmus stricto sensu war einfach einzuführen (Ziel 1). Schwieriger sind die Feedback-Mechanismen (Ziel 2). Hier können noch viele Tests durchgeführt werden und es gibt mehrere Plattformen, die in den letzten Monaten entstanden sind und vielversprechend sind. Hier könnten somit noch Ressourcen investiert werden, um bessere Feedback-Mechanismen zu generieren.

Das grösste Risiko besteht bezüglich der Produktionsintegration. Einerseits müssen IT-technische Grundsatzen geklärt werden. Andererseits müssen die Datenlieferanten in einer Testphase miteinbezogen werden, um ein nachfrageorientiertes Produkt herzustellen. Ein weiteres Risiko besteht in der Thematik «change management», die auch kulturelle Elemente beinhaltet.

Im Falle einer Fortsetzung des Projekts sollte ein Monitoring eingeführt werden, um die Algorithmen kontinuierlich anzupassen und systematische Abweichungen feststellen zu können.

A. Anhang: Anwendungsbeispiel

Die folgende Abbildung zeigt schematisch den Ablauf von Plausi++. Die einzelnen Schritte (1 bis 5) werden an einigen konkreten Beispielen illustriert³.



A.1 Lernen aus alten Daten

Aus den konsolidierten Daten der Vergangenheit wird ein ML-Algorithmus «trainiert». Dieser erlaubt die Vorhersage der Personalkategorie, sobald er mit neuen Daten konfrontiert wird. Die folgende Tabelle enthält beispielhaft die im BFS zur Verfügung stehenden Informationen, die für das Training verwendet werden (in der Dimension von Hundertausenden von Zeilen). Die Zielvariable ist «Personalkategorie», das heisst, der ML-Algorithmus lernt, diese Variable vorherzusagen.

Geburts-jahr	VZÄ	Fachbereich	Forschung	Lehre	...	Personal-kategorie
1985	76%	Recht	47%	42%	...	A
1979	13.5%	Medizin	93%	2%	...	A
1962	11%	Geistes- und Soz. Wiss.	00%	100%	...	U
1982	39%	Recht	42%	38%	...	P
1971	100%	Medizin	53%	9%	...	W
...	...	...	...	...	...	...

³ Es werden in diesem Anhang aus datenrechtlichen Gründen nur hypothetische Daten gezeigt.

A.2 Lieferung von Hochschule

Die Hochschule liefert neue Daten. Diese werden im BFS aufbereitet, um sie dem ML-Algorithmus übergeben zu können. Zum Beispiel werden Variablen transformiert, neue Variablen berechnet und Verknüpfungen mit vergangenen Daten vorgenommen. Die gelieferte Person 2 mit dem Geburtsjahr 1976 arbeitet mit einem Vollzeitäquivalent von 0.69% an der Hochschule und wird als Professor geliefert.

	Geburts-jahr	VZÄ	Fachbereich	Forschung	Lehre	...	Personal-kategorie
Person 1	1982	38.5%	Medizin	82%	5%	...	A
Person 2	1975	0.65%	Interdisziplinär	0%	0%	...	P
Person 3	1981	33.5%	Geistes- und Soz. Wiss.	69%	9%	...	P
Person 4	1984	71%	Geistes- und Soz. Wiss.	16%	73%	...	U
Person 5	1979	16.3%	Geistes- und Soz. Wiss.	0%	0%	...	W

A.3 Prediction

Der in Schritt 1 trainierte ML-Algorithmus berechnet (bedingte) Wahrscheinlichkeiten, mit denen sowohl die gelieferte als auch die anderen möglichen Personalkategorien korrekt sind. Die folgende Tabelle zeigt für unsere Beispielfälle, dass die gelieferte Personalkategorie der mit der grössten Wahrscheinlichkeit vorhergesagten Personalkategorie in vier von fünf Fällen nicht entspricht. Für Person 2 sagt der ML-Algorithmus mit einer Wahrscheinlichkeit von 0.3% die Kategorie Professor voraus, aber mit 99.4% die Kategorie übrige Dozierende. Bei Person 4 sagt der Algorithmus dieselbe Personalkategorie voraus, die auch geliefert worden ist.

	Gelieferte Personalkategorie	Wahrscheinlichkeit der gelieferten Personalkategorie	Vorhergesagte Personalkategorie	Wahrscheinlichkeit der vorhergesagten Personalkategorie
Person 1	A	3.4%	W 	88.9%
Person 2	P	0.3%	U 	99.4%
Person 3	P	4%	W 	94.4%
Person 4	U	76.6%	U 	76.6%
Person 5	W	6%	U 	89.5%

A.4 Feedback

Neben der eventuell fehlerhaften Personalkategorie können den Hochschulen noch weitere Informationen zurückgespielt werden. Es geht dabei in erster Linie darum, dass ja auch andere Variablen falsch geliefert worden sein könnten. Die folgende Tabelle illustriert dies an unseren Beispielfällen. Für die oben schon erwähnte Person 2 wäre die Kategorie Professor viel wahrscheinlicher, wenn die Variable «VZÄ» höher wäre. Eventuell liegt also nicht ein Fehler bei der Personalkategorie vor, sondern bei den «VZÄ». Für Person 4 wird kein Feedback generiert, weil der Algorithmus die Personalkategorie vorhersagt, die auch geliefert worden ist.

	Feedback an Datenlieferanten
Person 1	Das Alter (in Kombination mit den anderen Variablen) spricht dafür, dass es womöglich falsch ist oder sich um einen wissenschaftlichen Mitarbeitenden statt einem Assistierenden handeln könnte.
Person 2	Die niedrige VZÄ-Quote (in Kombination mit den anderen Variablen) deutet darauf hin, dass es sich hier um einen übrigen Dozierenden handelt oder die VZÄ-Quote zu tief ist.
Person 3	Die niedrige Anteil Lehre (in Kombination mit den anderen Variablen) macht es wahrscheinlicher, dass es sich hier um einen wissenschaftlichen Mitarbeitenden handelt oder der Anteil Lehre falsch ist.
Person 4	OK ✓
Person 5	Ein Anteil Forschung von 0% (in Kombination mit den anderen Variablen) deutet nicht auf einen wissenschaftlichen Mitarbeitenden hin. Entweder ist der Anteil der Forschung oder die Personalkategorie falsch.

A.5 Finale Lieferung ans BFS

Die von der Hochschule validierten Daten bilden nun die Grundlage, um den Algorithmus für das folgende Erhebungsjahr neu zu trainieren. Es ist wichtig, den ML-Algorithmus mit den jeweils aktuellsten Daten zu trainieren, um einerseits strukturellen Veränderungen über die Zeit hinweg Rechnung zu tragen sowie andererseits die verbesserte Datenqualität zu nutzen, um die Vorhersagekraft des ML-Algorithmus zu erhöhen.

B. Anhang: Die ausführliche Dokumentation

Die Beilagen zu diesem Schlussbericht beinhalten die Dokumentation und somit die technischeren Elemente des Pilotprojekts und sollen einen vollständigeren Überblick über die Arbeiten geben. Die Beilagen sind in zwei grössere Blöcke unterteilt. Im ersten Block werden die einzelnen Phasen nach dem Prozessmodell für die Dateninnovation «CRISP-DM» detaillierter als oben beschrieben. Im zweiten Block wird der kommentierte Code vorgestellt.

B.1 Business Understanding

Vor der Durchführung des Pilotprojekts sowie während der Durchführung wurde die SWOT-Analyse konstant aktualisiert. Die folgende Abbildung zeigt eine dieser SWOT-Analysen.

SWOT-Analyse	
STRENGTHS -Flexibilität in der Ausgestaltung -Interessante Themen -Know-How und Affinität vorhanden -Nullszenario macht keinen Schaden -Wir können bereits mehr als Null vorweisen -Nebenprodukte (USL; Interne SL ohne FBM?)	WEAKNESSES -Ressourcen/Management
OPPORTUNITIES - Interne Kooperationsmöglichkeiten - Externe Vernetzungsmöglichkeiten	THREATS -Interpretabilität des Algorithmus -Probleme bei Produktionsumsetzung -R und Packages auf Server benötigt!

Wir haben dabei mehrere Stärken und Chancen identifiziert, mit denen wir versucht haben das Projekt vorwärtszubringen. Beispielsweise haben die Flexibilität der Ausgestaltung und auch die Präsenz von interessanten Themen (wie auch eine Art Weiterbildungsmöglichkeiten dabei) zu Interesse wie auch Mitarbeit geführt. Dabei haben sich insbesondere das bereits vorhandene Know-How und die statistische Affinität als eine Stärke erwiesen.

Es wurde bereits früh bemerkt, dass die fehlende Kontinuität der Ressourcen ein Problem darstellen könnte (unter Schwächen). Als Risiken haben wir frühzeitig die Interpretabilität des Algorithmus, also den Feedback-Mechanismus, bezeichnet. Eine gesonderte Risikoanalyse (in der folgenden Abbildung) ist dabei genauer auf diese Risiken eingegangen.

Risikoanalyse

RISIKO	Eintrittswahrsch. (1-4)	Schadenspotenzial (1-4)	Risikopotenzial (E * S)	Massnahmen
Ressourcen / Management	4 (vorher 3)	3 (vorher 1)	12	Modularität; Motivation; Management?
Interpretabilität	2 (Auf Kurs!)	4	8	Einbezug Diego
Produktionsumsetzung	2	4	8	Kommunikation Fachexperten und IT
IT (Architektur; System2020)	0 alles OK!			

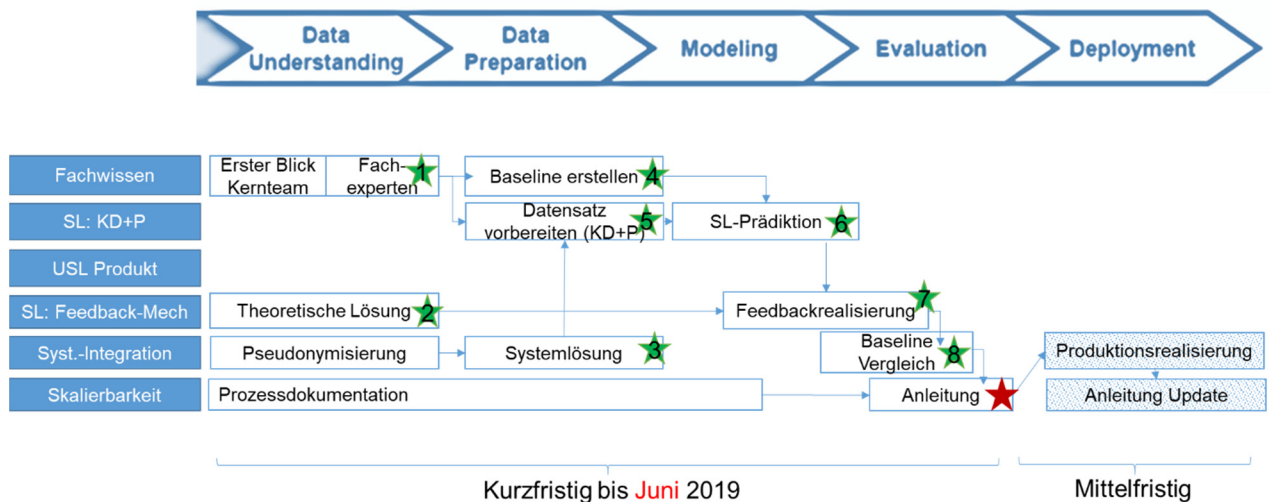
So zeigt die Abbildung einen gewissen Vektor, der entstanden ist, da gewisse Risiken in der Anfangsphase anders bewertet wurden als später. Beispielsweise ist während des Projekts klarer geworden, dass die unzulängliche Konstanz der Ressourcen ein Problem darstellt. So war die Eintrittswahrscheinlichkeit 4 (quasi sicher) und das Schadenspotenzial als 3 eingestuft, da es negative Konsequenzen hatte aber das Gesamtprojekt

DI Pilotprojekt Plausi++ BB

auch nicht gefährdet hat. Eine fehlende Entwicklung der Interpretabilität und die Thematik mit der Produktionumsetzung wurden mit einem grösseren Schadenspotenzial eingeschätzt, da diese auch die Zukunft einer solchen Technologie in Frage stellen würden. Die Eintrittswahrscheinlichkeit wurde dabei mittelmässig eingestuft. Was die sonstige IT-Architektur mit dem Atlantica-Server und den sonstigen Elementen betraf, hat sich herausgestellt, dass alles kein Problem war.

Die Arbeitsschritte und Meilensteine des Pilotprojekts wurden wie folgt definiert. Zumindest diese Meilensteine konnten in einem Wasserfall-artigen Modell interpretiert werden.

Arbeitsschritte und Meilensteine



Dabei zeigt sich wie wir zunächst mit den Fachexperten Zeit verbracht haben, um die Datenlage und die Betriebselemente besser zu verstehen. Es wurde auch schon früh mit der Erarbeitung einer theoretischen Lösung für den Feedback-Mechanismus begonnen – wissend, dass es ein schwieriges Unterfangen sein würde. Der grösste Teil des Projekts machte dann insbesondere die SL-Prädiktion, die Erstellung der Baseline und die Feedbackrealisierung aus. Der letzte Meilenstein war die Erstellung des Schlussberichts und dieser ausführlichen Dokumentation.

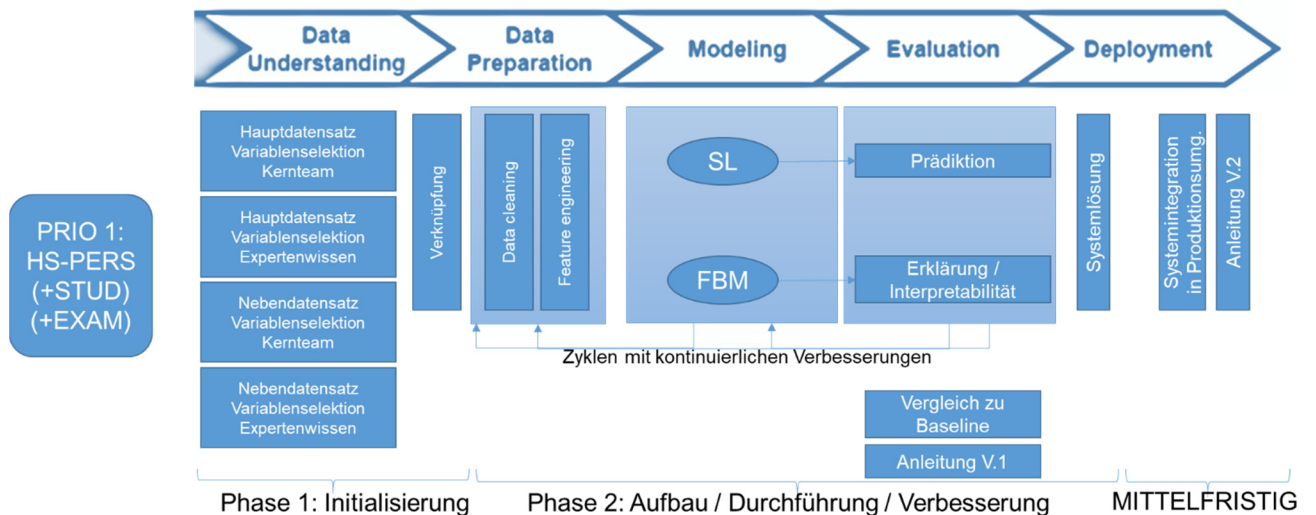
Die grobe Terminplanung hat sich dabei wie folgt zugetragen. Zunächst haben wir drei verschiedenen Datensets verschiedene Prioritäten zugewiesen. Das Hochschulpersonal in der ersten Zeile hat dabei die höchste Priorität erhalten. Die zweite Priorität wären die nahe beieinanderliegenden Datensets Statistik der Lernenden), Statistik der Bildungsabschlüsse (SBA) und Statistik der beruflichen Grundbildung (SBG) gewesen. Die dritte Priorität wäre Schulpersonal gewesen. Wie man sieht konnten wir die Meilensteine bei der Priorität 1 erreichen. Bei den Prioritäten 2 und 3 konnten wir u.a. aus Zeitgründen nicht weitere Tests durchführen.

Die letzte Zeile beinhaltet querschnittsübergreifende Elemente zu Networking, Publikation und Konferenzen (Ne-Pu-Ko). Es zeigt auf, dass Plausi++ die Zwischenergebnisse auch an den Schweizer Statistiktage 2018, an einem UNECE-Workshop, an der NTTS 2019 in Brüssel sowie an der ITACOSM in Florenz präsentiert hat.

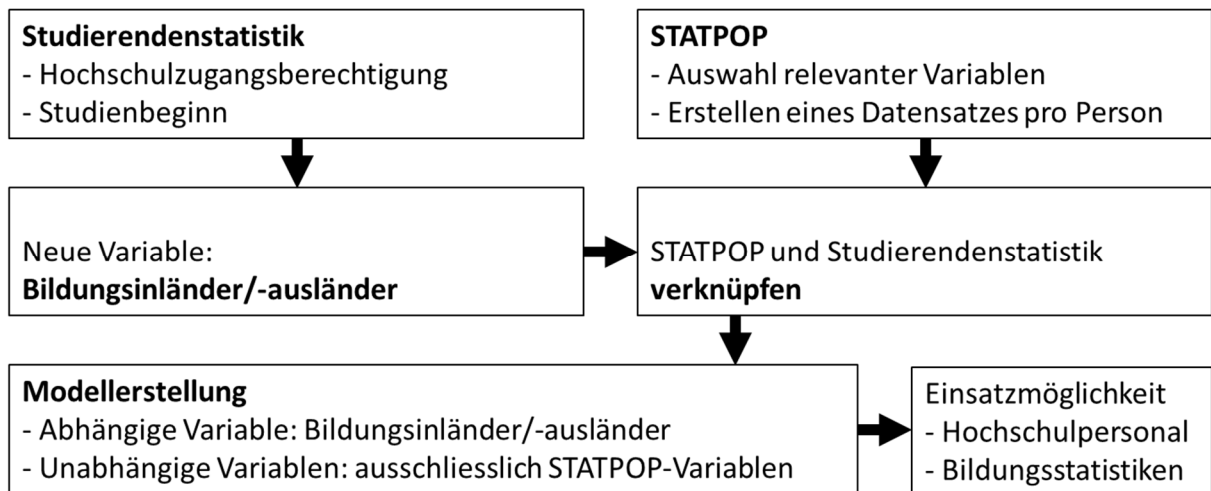
Grobe Terminplanung

WAS	APR	MAI	JUN	JUL	AUG	SEP	OKT	NOV	DEZ	J-F	M-A	M-J	+++++
	Data Understanding		Data Preparation			Modeling			Evaluation		Deployment		
PRIO 1 HS-PERS /STUD/ EXAM		M1: Fachexper- ten (CR) Verknüp- fung & FE (MEA)	M2: Feedback Lösungen (DK+CR)	M3: Techni- cal solution	M4: Baseline Erstellen (CAM) M5: Datensatz vorbereiten (CAM)		Präsen- tationen LIME und DALEX			M6: SL Prädikti- on	M8: Baselin- e Vergl.	M7: Feedback- Lösung Dokumenta- tion	Produktions realisierung
PRIO 2 SDL/SBA /SBG	Erstes Screen- ing		M1: Fachexpe- rten (CR)			Entscheidu- ng: Keine Zeitressou- rcen							
PRIO 3 SSP						JE signalisiert Interesse später zu testen						Entscheidu- ng: Keine Zeitressour- cen	
NE- PU-KO	Kickoffs			UNECE -WP Info1	SSS Ende Aug	UNECE Ende Sept		Zwisch- enberic- ht	Präs. GL		NTTS	ITACOSM Schlussberi- cht	

Die nächste Abbildung zeigt genauer, wie wir uns innerhalb des CRISP-DM bewegt haben. Insbesondere haben wir festgestellt, dass es kontinuierliche Zyklen zwischen den Phasen Data Preparation, Modeling und Evaluation gab. Konkret bedeutet dies, dass die Arbeiten bei der Modellierung und bei der Evaluation uns immer wieder gezeigt haben, dass wir beispielsweise die Daten anders aufbereiten müssen oder andere «features» generieren müssen. Es ist somit eine agile Vorgehensweise entstanden, da wir auf diese Elemente reagieren mussten.



Wie bereits oben beschrieben, gab es mehrere Datensets, die in Betracht gekommen sind, um derartige Plausibilitätskontrollen durchzuführen. Wir wurden ebenfalls von anderen Seiten im Amt kontaktiert. Des Weiteren könnte es aber auch weitere mögliche Anwendungsfälle für eine solche «supervised ML» geben, die sich nicht auf Plausibilitätskontrollen fokussiert. Ein Beispiel dazu ist der folgende, bei dem man aus STATPOP-Daten ableiten kann, ob jemand Bildungsinländer oder Bildungsausländer ist.



1. Aus den Daten der Studierenden (Hochschulzugangsberechtigung und Wohnort vor Studienbeginn) ableiten, wer Bildungsinländer und Bildungsausländer ist.
2. STATPOP-Daten vorbereiten, so dass sich pro Person die wichtigsten Angaben aus den letzten Jahren in einem Datensatz befinden.
3. STATPOP-Angaben verknüpfen mit den Daten der Studierenden.
4. Modellerstellung und -Analyse, um ausschliesslich aus den STATPOP-Angaben zu entscheiden, ob eine Person Bildungsinländer oder –ausländer ist.
5. Beispielanwendung des Modells auf das Hochschulpersonal

B.2 Data Understanding

Das Datenset, welches wir schlussendlich ausgewählt haben, ist das Hochschulpersonal der universitären Hochschulen. Dieses besteht aus etwa 100'000 Datenpunkte pro Jahr. Dieses haben wir verknüpft mit den Datensets der Studierenden und der Studienabschlüsse, um zu wissen, ob eine Person in den letzten Jahren studiert hat oder einen Abschluss gemacht hat. Dies würde die Wahrscheinlichkeit einer wissenschaftlichen Assistenz erhöhen und beispielsweise eine Professur unwahrscheinlicher machen. Aufgrund der Verknüpfungen, welche basierend auf den AHVN13-Nummern seit 2013 stabil verfügbar sind, haben wir als Zeitraum 2013 bis und mit 2017 genommen.

Die Daten, die wir trainiert haben sind die konsolidierten Daten. Diese Daten wurden bereits durch die alte Plausi und den dementsprechenden Antworten der Datenlieferanten korrigiert. Das bedeutet auch, dass die konsolidierten Daten Fehler haben, die nicht korrigiert wurden (und wir nicht genau kennen). Ein trainierter ML-Algorithmus wird somit auch teilweise aus Fehlern trainiert. Wir möchten aber nochmals betonen, dass die Resultate dennoch positiv sind. Wir glauben, dass dies aber womöglich gewisse systematische Fehler kaschiert. Für die verwendeten Testdaten haben wir dafür die Rohdaten verwendet. Es war ein technisch anspruchsvolles Unterfangen die Rohdaten zu verwenden, da dies nie so vorgesehen war. Nur durch die Verwendung der Rohdaten können wir eine Baseline erstellen und Vergleichen welche Daten von der alten Plausi als Fehler angesehen waren – und welche dann auch korrigiert wurden.

Ein wichtiges Element des "Data Understanding" war es die aktuelle Plausi zu verstehen. Die folgende Tabelle beschreibt die Regeln der aktuellen Plausi. Es gibt 24 Regeln wobei einige bestätigbar («confirmables») sind: Wenn diese nicht bestätigbar sind (eine '0' in der Tabelle unten) müssen die Datenlieferanten diesen Datenpunkt zwingend korrigieren. Diese Regeln sind in der folgenden Tabelle rot eingezeichnet. Bei den «confirmables» können die Datenlieferanten bestätigen, dass es sich bei dieser Fehlermeldung um korrekte Daten handelt (beziehungsweise die Universität kann diese natürlich auch bestätigen ohne einen genaueren Blick geworfen zu haben).

Nun haben wir hier die «confirmables» Regeln in zwei Gruppen eingeteilt. Gelb eingezeichnet sind Regeln, bei der Plausi++ nicht helfen kann. Beispielsweise funktioniert Plausi++ pro Fall und schaut deshalb nicht ob

DI Pilotprojekt Plausi++ BB

eine Funktion zweimal auftritt (Regel 5). Ebenfalls spielt die Präsenz einer AHVN13 für Plausi++ keine Rolle, da es eine formelle Regel ist (Regel 6). Und so weiter. Grün eingezeichnet sind dafür diejenigen Regeln, bei der Plausi++ theoretisch ebenfalls einen Fehler finden könnte (aber nicht muss). Schlussendlich sind blau weitere inhaltliche Regeln dargestellt, zu denen aber Plausi++ auf ganze andere Rückschlüsse kommen könnte. Diese Regeln und Grenzen wurden willkürlich gewählt: Beispielsweise wenn der Beschäftigungsgrad einer Person über 1.2 ist (Regel 14).

Regel	Confirmable	Beschreibung
1	0	Der Vorlaufrecord der Dateien muss dem Lieferant und der Erhebung entsprechen.
2	0	Alle obligatorischen Felder müssen einen Wert enthalten.
3	0	Die Formate müssen den Richtlinien des technischen Handbuchs entsprechen.
4	0	Bei Variablen mit existierenden Nomenklaturen müssen gültige Codes verwendet werden.
5	1	Zwei Funktionen sollten nicht den gleichen Inhalt haben.
6	1	Die Personen mit einem Beschäftigungsgrad grösser als 5 Prozent sollten eine AHVN13 enthalten.
7	0	Die Kontrollziffer AHVN13 erfüllt die auto-Plausibilitätsregel.
8	1	Das Geschlecht und das Geburtsjahr der Personen mit AHVN13 sollten mit der UPI-Datenbank übereinstimmen.
9	1	Eine Person sollte während des Referenzjahres zwischen 15 et 70 Jahre alt sein.
10	0	Die Summe aller Tätigkeitsanteile jeder Funktion muss genau gleich 1.0 sein.
11	1	Der Beschäftigungsgrad am 31.12 muss grösser oder gleich 0 und kleiner oder gleich 1 sein (kein Wert unter Null).
12	1	Die Summe aller Beschäftigungsgrade einer Person am 31.12 muss grösser oder gleich 0 sein und kleiner oder gleich 1 sein.
13	0	Jedes Vollzeitäquivalent muss grösser als 0 sein.
14	1	Die Summe aller Vollzeitäquivalente der Person muss kleiner oder gleich 1.2 sein.
15	1	Die Personalkategorie 550 darf nicht geliefert werden.
16	1	Die Fachrichtung sollte in der Hochschule existieren. Als Grundlage dient der Fächerkatalog der universitären Hochschule.
17	1	Die gelieferte Anzahl Personen muss sich im Vertrauensintervall der Zeitreihen pro Hochschule und Fachbereich befinden.
18	1	Alle Personalkategorien sollen sich in der gelieferten Datei befinden.
19	1	Personen in der Personalkategorie 522, welche nicht den Anstellungsstatus 30 haben, sollten einen Beschäftigungsgrad am 31.12. von < 0.9 haben.
20	0	Die Variablen Geburtsjahr, Geschlecht und Nationalität müssen für jede Person identisch sein.
21	1	Es wurden keine Personen mit einem BG am 31.12 von 0.0 geliefert. Es ist nicht plausibel, dass keine Person aus der Hochschule ausgetreten ist.
22	1	Personen mit Personalkategorie zwischen 511 und 519 sollten nicht einen marginalen Beschäftigungsgrad haben.
23	1	Personen mit Anstellungsstatus 'fest angestellt' sollten nicht einen marginalen Beschäftigungsgrad haben.
24	1	Diese Variable ist nur für das Personal des Fachbereichs Humanmedizin zu liefern

Bereits durch die bisherige Plausibilisierung ist die Datenqualität insgesamt gut und steigt auch kontinuierlich. Dennoch vermuten die Fachverantwortlichen zum Hochschulpersonal Qualitätsunterschiede zwischen

DI Pilotprojekt Plausi++ BB

den Hochschulen bei den Datenlieferungen. Bei den jährlichen Treffen kommt es gelegentlich vor, dass eine Hochschule seine Kategorisierungssystematik verändert und dabei Unkorrektheiten rauskommen.

Daraus ergeben sich zwei verschiedene Herausforderungen:

1. Umgang mit dem Umstand, wenn die Resultate zwischen Plausi++ und der bisherigen Plausibilisierung voneinander abweichen (siehe folgende Tabellen)
2. Für die Entwicklung von Modellen basierend auf diesen Daten bedeutet das, wenn die Datenlieferungen systematisch Fehler beinhalten, dann könnten diese auch in das Modell integriert sein.

Um dieser Problematik entgegenzuwirken kam die Überlegung auf, neben dem Gesamtmodell, Modelle für jede einzelne Hochschule zu entwickeln. Aus diesen Modellen könnte das Modell einer HS gewählt werden, die besonders zuverlässig erscheint.

Für den ersten Fall bringt dieses zusätzliche Modell eine ergänzende Einschätzung ein, die bei der Entscheidung, ob nun ein Fehler in den Daten vorliegt (-+) oder das Gesamtmodell von der Realität abweicht (+-), weiterhilft und scheinbare Widersprüche auflöst, da eine zweite Einschätzung vorliegt (siehe folgende Tabelle).

Plausi++ Gesamtmo- dell	Aktuelle Plausibilisierung		Weiteres Modell für HS mit guter Qualität		
		+	-	+	-
	+	Anzahl ++	Anzahl -+	Anzahl ++ und -++	Anzahl -+-
	-	Anzahl +-	Anzahl --	Anzahl +-+	Anzahl -- und +--

Für den zweiten Fall, um systematische Fehler zu identifizieren könnten die drei Modelle verglichen werden: das Plausi++ Gesamtmodell, das Modell für die untersuchte HS und ein Modell einer Hochschule das besonders vertrauenswürdige erschien (siehe folgende Tabelle).

Plausi++ Gesamtmo- dell	Modell der untersuchten HS		Weiteres Modell für HS mit guter Qualität		
		+	-	+	-
	+	Anzahl ++	Anzahl -+	Anzahl ++ und -++	Anzahl -+-
	-	Anzahl +-	Anzahl --	Anzahl +-+	Anzahl -- und +--

Aus diesen Überlegungen entstanden auch Ideen weiterer Features/Variablen: der Grössenunterschied zwischen den Hochschulen wurde als zusätzliches Merkmal durch die Fachverantwortlichen vorgeschlagen, sowie die Sprachregion und strukturelle Organisationsstrukturen (ETH oder Universität).

Da die Hochschulnamen ein aggregiertes Feature für manche der neuen Features sind, kam die Überlegung auf, dass bei Hinzunahme der neuen Features die Variablen zu den Hochschulen eventuell weggelassen werden müssten. Für das aktuelle Gesamtmodell würde das auch bedeuten, dass bei Entfernen der Hochschulnamen die (Vorhersage-)Genauigkeit des Modells stark zurückgehen müsste, da die Hochschulnamen eine Sammlung wichtiger Informationen sind und diese dann fehlen würde.

Wegen dieser Bedenken und des zeitlich nötigen Umfangs bei der Entwicklung zusätzlicher Modelle wurden statt neuer Modelle die vermuteten Qualitätsunterschiede und zusätzlichen Ideen als neue Features eingebaut, um einen ersten Anhaltspunkt zu erhalten, ob diese zusätzlichen Informationen Relevanz für das Modell haben.

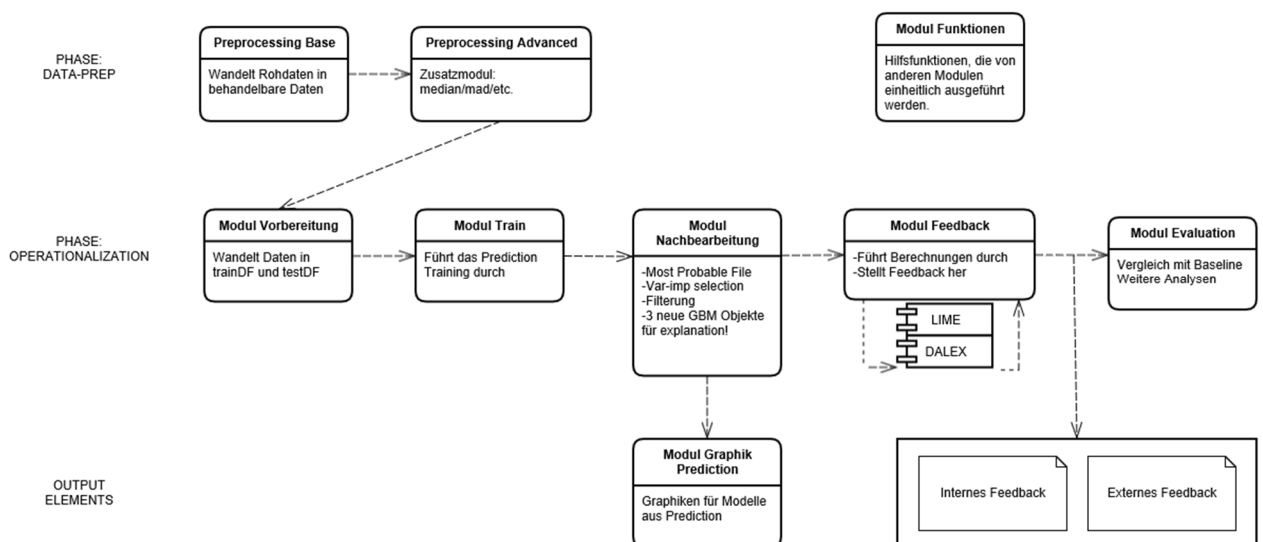
Ein weiterer wichtiger Managementaspekt ist der datenrechtliche Schutz der Individualdaten. Natürlich liegen die Identifikationen, die ursprünglich auf die AHV-Nummern zurückzuführen sind, nur in pseudonymisierter Form auf dem Atlantica Server vor.

B.3 Data Preprocessing

Wie bereits oben beschrieben haben wir die Daten des Hochschulpersonals in den universitären Hochschulen der Jahre 2013-2017 genommen und mit denen der Studierenden und der Studienabschlüsse verknüpft.

Das «Data Processing» startete mit einer ziemlich aufwändigen Umwandlung, die aber nur darauf zurückzuführen ist, dass wir eine Baseline berechnen wollten und eine solche in der alten Plausi nicht auf diese Weise vorgesehen war. Wir mussten also a posteriori eine Lösung erarbeiten. Zunächst müssen nämlich die Rohdaten aufbereitet werden. Das hat in mehreren Anläufen mehrere Wochen (nicht Vollzeit) in Anspruch genommen, da gewisse Zwischenschritte durchgeführt werden mussten, die bei der Erstellung der alten Plausi so nicht angedacht waren. Für jeden Datenlieferanten liegen mehrere Files sowie Plausibilisierte Rückfragen vor. Nun hatten wir die Idee, dass wir die ersten Files (wir nannten diese «FIRST») und die letzten Files («LAST») miteinander vergleichen müssen. Somit würden wir wissen, wie die Daten ursprünglich geliefert wurden, welche Daten durch die alte Plausi als falsch bezeichnet wurden, und welche davon korrigiert bzw. ignoriert wurden. Dieser Vergleich stellte sich als umständlicher heraus als erwartet. Die Universitäten nutzten die ersten Dateisendungen, um ihrerseits gewisse Tests bezogen auf ihre Datendurchzuführungen. In diesem Fall entspricht dann erst das dritte oder vierte File der inhaltlich wahren «FIRST». Erschwerend kommt hinzu, dass die Datenqualität nicht linear zunimmt. Im Laufe dieses Prozesses werden manchmal zuvor korrekte Werte verändert. Die Generierung der Rohdaten stellte sich durch weitere ähnliche Phänomene als recht aufwändig dar.

Die nächsten Schritte dann sind Teil des Plausi++-Prozesses und vollständig automatisiert. Die folgende Abbildung zeigt die Übersicht, über die Module, die wir verwendet haben. Sie werden unten beschrieben.



Modul Preprocessing Base: Hier findet die Umwandlung der Rohdaten statt. Das gelieferte Datenformat muss umorganisiert werden. Im Fall des Hochschulpersonals werden jährlich die Daten pro Zeile geliefert, wobei eine Zeile eine «Funktion» einer Personalkategorie sein kann, ein absolviertes Studienjahr einer Person oder ein Abschluss einer Person. Eine «Funktion» ist eine spezielle Definition für die Personalkategorien. Eine Person ist an einer Hochschule angestellt und kann dort verschiedene «Funktionen» ausüben. Dieselbe Person erscheint somit in mehreren Zeilen, wenn sie verschiedene «Funktionen» hat. Darüber hinaus kann dieselbe Person auch mehrmals im Datenset sein, wenn Sie an mehreren Hochschulen angestellt ist. Es kommt somit durchaus vor, dass die Person mehrere Funktionen an mehreren Hochschulen hat.

Somit ist ein erster Schritt, diese Daten so umzuwandeln, dass eine Zeile pro Person für ein bestimmtes Jahr entsteht. In dieser Zeile sollen alle Informationen zu den Funktionen, zu den verschiedenen Anstellungen in den verschiedenen Hochschulen, und zur Präsenz in gewissen Studiengängen oder dem Abschluss von Examen in den letzten Jahren vorhanden sein. Das erfolgt durch mehrere Transponierungen, d.h. durch Umwandlungen von Zeilen zu Spalten und umgekehrt.

Anschliessend entsteht ein Datenset, der alle Jahresinformationen der vergangenen X Jahre in einer Zeile

beinhaltet, z.B. eine weibliche Person war vor drei Jahren Doktorierende und wissenschaftliche Mitarbeitende, vor zwei Jahren erfolgte der Abschluss und vor einem Jahr wurde sie Professorin. Das Umwandlungsergebnis sind sehr viele Features. Dabei wird das Datenset bereits in ein Train-Set und ein Test-Set gesplittet. Die Train-Daten umfassen die Jahre 2013-2016 und haben als abhängige Variable die Personalkategorie im Jahr 2016. Der Test-Set besteht, wie in der simulierten Wirklichkeit, aus den letzten gelieferten Rohdaten und besteht aus den Daten der Jahre 2014-2017.

Es werden in diesem Modul auch einige «Fachexperten-Variablen» gebildet, da unsere Fachexperten ihr Feedback zu den vertrauenswürdigeren Hochschulen gegeben haben und noch weitere Variablen auf einer aggregierten Ebene hinzugefügt haben. Das hilft dem ML-Algorithmus herauszufinden, wo gewisse Muster bestehen könnten – beispielsweise wenn die Datenqualität etwas niedriger ist bei gewissen Universitäten.

Am Ende des Moduls wird auch ein separater Datenset mit verschiedenen Informationen abgespeichert, die nicht an den Algorithmus geschickt werden sollen, da diese nicht verarbeitet werden können. Diese sind aber wichtig und werden später im Feedback-Modul benötigt.

In den letzten Versionen dieses Moduls haben wir auch alle möglichen kategorialen Variablen in sogenannte «Faktorvariablen» (in der Programmiersprache R) umgewandelt. Es funktioniert technisch auch ohne diese Umwandlung, aber wir haben die Erfahrung gemacht, dass dies für den Feedback-Teil eine starke interpretative Hilfe bereitet. Es hat aber keinen Einfluss auf die prädiktive Kraft – nach unseren Informationen werden diese Variablen für den prädiktiven Algorithmus auch in binäre Einzelteile zerlegt. Wir haben ebenfalls geschaut, dass die binären Variablen im korrekten R-Format verwendet werden.

Modul Preprocessing Advanced: Dieses Modul ist so programmiert, dass es auch weggelassen werden kann. Im Prinzip werden hier die (im Fall des Hochschulpersonals) rund 1'000 «expert variables» gebildet. Das sind Features, die durch statistische Grössen aus anderen Variablen oder Variablengruppen gebildet wurden. Darunter gehören Durchschnitt, Median, MAD, Min, Max und andere. Wichtig war hier auch den Code zu optimieren, da es sonst ziemlich lange dauern würde (u.a. mit dem R-Package «Rfast»).

Modul Vorbereitung: Zunächst werden die Proportionen der Personalkategorien berechnet. Das wird benötigt, um entweder «priors» zu berechnen oder ein «downsampling» zu machen. Diese Elemente werden durchgeführt, um eine Unausgeglichenheit zwischen den abhängigen Variablen auszugleichen. Beispielsweise könnte (rein theoretisch) Variable A nur 5% mal vorkommen, während Variable B 60% vorkommt. Die Variable B wäre somit überrepräsentiert. Es gibt verschiedene Arten wie man dieses Problem lösen kann. Plausi++ sieht eine Option vor, mit der man verschiedene dieser Varianten testen kann: «downsampling», «upsampling» und «priors». Wir haben mit diesen experimentiert und sind (wie theoretisch erwartet) auf ähnliche Resultate gekommen. «Upsampling» sollte nicht verwendet werden, da der Train-Algorithmus viel länger brauchen wird und die Genauigkeit dennoch nicht verbessern kann. Wir verwenden aber als Standard die «priors», weil dies in der klassischen Statistik besser akzeptiert ist. Der Algorithmus aus dem R-Package «caret» verwendet dann die «priors» auch auf eine Standardweise.

Als nächstes werden die Variablen mit «non-zero variance» (NZV) berechnet. Dies wurde nach einiger Überlegung per Standard deaktiviert, da es nicht notwendig ist und womöglich die Accuracy reduzieren könnte. Falls man es aber aktiviert, könnte es die Train-Berechnungen etwas beschleunigen.

Dann wurden gewisse Elemente im Code gelöscht, welche die Korrelationen und die Skewness getestet haben. Diese Tests sind für die verwendeten Baualgorithmen nicht notwendig.

Im nächsten Schritt werden die beiden 'Dataframes' für den Trainings-Datensatz («trainDF») für den Test-Datensatz («testDF») gebildet. Wie bereits oben erwähnt, wurde dies im Modul Preprocessing Base bereits vorbereitet und ist deshalb bloss die Unterscheidung einer binären Variable. Es wurde aber auch eine Option programmiert, so dass ein klassisches Splitting, wie in der ML sonst üblich, gemacht werden könnte.

Modul Train: Das ist das Herzstück des gesamten Projekts, da hier die prädiktiven ML-Algorithmen ausgeführt werden. Dieses Modul wurde am häufigsten umgeschrieben, so dass es nun etwas komplexer ist. Vereinfacht gesagt, wurden verschiedene Arten implementiert, wie Optionen sowie Hyperparameter an den ML-

Algorithmus weitergegeben werden können. Zum Beispiel werden die Hyperparameter häufig mit Gittern («grids») an den Algorithmus weitergegeben, so dass er das gesamte Gitter durchtestet, um die passendste Hyperparameterkombination zu finden. Wenn verschiedene Hyperparameter mit verschiedenen Werten getestet werden sollen, dann steigt die Anzahl der Ausführungen schnell exponentiell an. Um gewisse Tests schnell auszuführen, gibt es somit eine «quick»-Option, die gewisse niedrigen Hyperparameter bereithält. Ein interessantes Element ist aber die Verwendung eines externen Files mit der man weitere Hyperparameter «füttern» kann. Die Idee entstand, da die Berechnung der Algorithmen sehr lange dauert. Man verliert aber Zeit, wenn man ein grosses Packet an Tests (beispielsweise mit Hyperparametern) in der Nacht beendet. Auch würde man in dieser klassischen Art in Paketen zu rechnen auch nur in grossen Schritten erfahren, wie die Hyperparameter wirken (bezüglich Accuracy). Deshalb wurde die Verwendung eines externen Files implementiert. Einerseits gibt das Programm laufend die Resultate heraus und speichert diese in einem File ab. Andererseits kann man in einem externen File bereits die nächsten «grids» für die Hyperparameter eingeben, da man bereits sehen kann, in welche Richtung man sich bereits bewegen könnte. Somit lässt sich sicherstellen, dass das Programm auf dem Server konstant läuft und weitere Hyperparameter testet. Somit lässt sich ebenfalls sicherstellen, dass man nicht ziellos riesige «grids» wählt, die eine exponentiell anwachsende Berechnungszeit benötigen. Stattdessen kann man die nächsten «grids» Smart setzen.

Es muss hier auch erwähnt werden, dass wir alle ML-Elemente verwenden, die «state-of-the-art» sind. Beispielsweise werden in diesem Modul die «cross-validations» durchgeführt. Wir haben nun ein sogenanntes «repeated cross-validations» hart implementiert, da es uns das sinnvollste in diesem Zusammenhang erscheint – man könnte dies aber auch sehr einfach in ein «bootstrapping» oder in eine andere Methode abändern. Als Optionen haben wir es so gesetzt, dass man die Anzahl der «cross-validations» und die Wiederholungen manuell setzen kann. Diese manuelle Setzung ist sehr nützlich, da man so kontrollieren kann, wie lange der ML-Algorithmus rechnen wird: Um zunächst gewisse Elemente zu testen eignen sich zunächst wenige «cross-validations». Um sich dann im nächsten Schritt bezüglich einer Accuracy sicher zu sein, muss man höhere Werte der «cross-validations» wählen.

Ebenfalls ist es wichtig, dass nach jedem berechneten Block, die Accuracy auch auf dem Test-Set herausgegeben wird. Der Test-Set ist während der Trainings-Phase nicht sichtbar für den Algorithmus – so wie es in der ML-Literatur als sine-qua-non bezeichnet wird. Das ist auch hier der Fall: Wir können damit die endgültige Accuracy beziffern, die aber auf keinen Fall in den Algorithmus zurückgespielt werden darf, um ein Over-Fitting zu verhindern.

Das Modul gibt verschiedene weitere Informationen heraus (z.B. «confusion matrix») und erstellt ein RDS-Objekt (RDS steht für «R Data Single» und ist ein binäres R-Format um Objekte abzuspeichern), welches alle Informationen des Algorithmus beinhaltet. Wichtig ist auch zu erwähnen, dass der Code so erstellt wurde, dass viele Computerprozesse gleichzeitig/parallel rechnen können. Das war wichtig, um die Berechnungen auf dem Atlantica Server zu beschleunigen.

Modul Nachbereitung: Mit dem Training-Modul steht nun das prädiktive Ergebnis bereits fest. Aber für die Erstellung von Graphiken, für die Durchführung der Feedback-Mechanismen, sowie für die Erstellung der Evaluationen/Analysen muss man noch eine Nachbereitung durchführen.

Zunächst werden die vorhergesagten Werte mit dem im Preprocessing ausgesonderten Datenset (mit den Identifikationsvariablen usw.) zusammengeführt, damit man wissen kann, um welche Personen es sich genau handelt.

Im zweiten Abschnitt werden verschiedene Modelle, die im prädiktiven Teil berechnet wurden, zusammengeführt. Dies wird gemacht um zu eruieren, welches die Variablen sind, mit der stärksten prädiktiven Kraft. Die Idee dahinter ist, dass verschiedene Algorithmen den Variablen auch verschiedene Gewichtungen zuteilen – sogenannte «variable importances». In diesem Teil werden diese Informationen somit gebündelt und in ein «top»-Modell zusammengefasst mit den besten der besten Variablen. Weiter unten wird das Schalenmodell noch genauer erklärt. Das «top»-Modell ist die Schale, die dort als «no human in the loop» genannt wird, da sie sehr technische Variablen (die vorher genannten «expert variables») beinhalten könnte, die für Menschen keinen Sinn machen könnten.

DI Pilotprojekt Plausi++ BB

Das Modul bereitet dann noch die anderen beiden Modelle vor, die die anderen beiden Schalen bilden. Für die unterste Schale werden nur diejenigen Variablen genommen, die von den Datenlieferanten geschickt werden. Für die mittlere Schale werden die Variablen aus der untersten Schale plus weitere Variablen genommen, welche die BFS-Mitarbeitenden verstehen können (aber nicht die Datenlieferanten).

Die Modelle werden am Ende noch mit dem ML-Algorithmus neu berechnet und drei neue RDS-Objekte werden somit generiert. Die Dauer ist hier aber relativ kurz verglichen mit dem zentralen Train-Modul, da hier keine Hyperparameter getestet werden und auch keine «cross-validations» usw. notwendig sind.

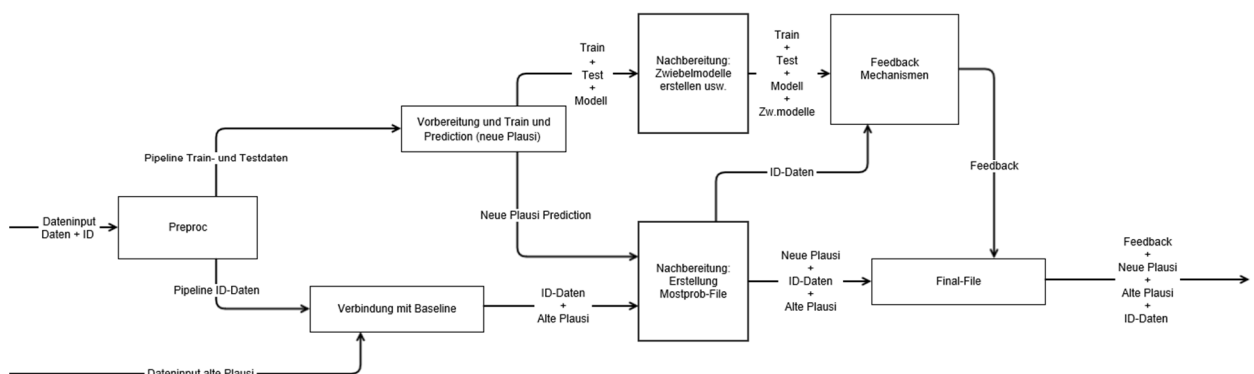
Modul Graphik Prediction: Dieses Modul produziert verschiedenste Graphiken und Outputs zur Prediction. So werden hier Graphiken zur «variable importance», Gains-Kurven, Vergleiche zwischen den verschiedenen Modellen usw. ausgegeben. Der Code berücksichtigt das Betriebssystem, da auf Linux und Windows leicht andere Rahmenbedingungen herrschen.

Modul Feedback: Hier werden die Feedback-Algorithmen zentral ausgeführt. Dabei kann man als Option einen bestimmten Algorithmus auswählen, oder eine Liste von vordefinierten Algorithmen ausführen. Als Resultat dieser Sub-Module wird das finale File gebildet. Im finalen File findet man die Kombination zwischen den ursprünglichen Variablen, den prädiktiven Elementen und den Feedback-Elementen. Die Sub-Module der Feedback-Algorithmen generieren aber auch zahlreiche Auswertungen und Graphiken, die intern von Nutzen sind.

Modul Evaluation: Das Modul erstellt weitere Auswertungen, die für den Vergleich mit der Baseline beziehungsweise der alten Plausi eine Rolle spielen. Hier kann man ebenfalls die Outputs der neuen Plausi und der alten Plausi gegenüberstellen. Dieses Modul spielt ebenfalls eine wichtige Rolle bei der Überprüfung von systematischen Fehlern – beispielsweise unterschiedliche Handhabungen von Datenlieferanten oder auch gewisse systematische Muster bei den Personalkategorien. Dies ermöglicht weitere Analyse für die BFS-Mitarbeitenden mit denen man auch gewisse Aspekte der Datenqualität genauer anschauen kann.

Erklärung zu Datenpipelines

Die Erklärung zu den Modulen ist wichtig, um die Struktur des Vorgehens und des Codes zu verstehen. Gleichzeitig gibt es aber auch die Sicht auf die Datenströme, die dabei entstehen. Die Graphik unten zeigt, wie die Datenströme, oder Datenpipelines, aufgebaut sind.



Im Prinzip sehen wir links in der Graphik wie zwei Datenströme auf den Atlantica Server kommen. Einerseits sind das bereits die Daten, die in einem internen vorformatierten Format sind (wie oben erwähnt mit den Transformationen für die Rohdaten). Andererseits brauchen wir für dieses Projekt (vielleicht nicht unbedingt für die spätere Produktionsintegration) auch die Informationen aus der alten Plausi – insbesondere, um die Tests zur Baseline bilden zu können.

Nun ist es so, dass einige Informationen am Anfang getrennt wurden vom «Hauptdatenstrom». Es handelt sich bei der Trennung um Personen identifizierende Angaben. Die wichtigste ist die bereits pseudonymisierte Personennummer (ein Identifikator, der auf der AHVN13 beruht aber nicht mehr zurück verfolgbar ist), die keinen Wert für das Trainieren von Modellen hat. Zusätzlich generieren wir beim Preprocessing noch eine eigene ID-Variable, damit wir die Fälle nachher eindeutig und mit derselben Reihenfolge zuordnen können. Auch diese ID wird abgetrennt und nicht an die «Hauptpipeline» (in der Graphik als «Pipeline Train- und Testdaten» bezeichnet) weitergeleitet. Da sowohl die Personennummer wie auch die ID beides Identifikatoren sind, sind diese im Diagramm als «ID-Daten» gekennzeichnet. Darüber hinaus befinden sich noch Daten darin, die ebenfalls zur eindeutigen Zuordnungen benötigt werden, wie die Leistungsnummer ANR und auch der Name der Hochschule. Für die eineindeutige Zuordnung der Funktion braucht man sowohl die Personennummer, die ANR wie auch den Namen der Hochschule. Dies wird alles benötigt um Falschverknüpfungen zu verunmöglichen.

Diese «Hauptpipeline» führt zu den Modulen «Vorbereitung» und «Train», welche die Datenframes «trainDF» und «testDF» für den ML-Teil vorbereiten und danach die Algorithmen damit ausführen. Diese beiden Datensets beinhalten sowohl die unabhängigen Variablen wie auch die abhängige– und zwar getrennt nach den Train-Daten und den Test-Daten.

In der Nachbearbeitungsphase wird nun einerseits das Rohr mit den Trainings- und Testdaten weiterbehandelt. Das führt beispielsweise zu den reduzierten Modellen des Zwiebelmodells, wie oben beschrieben. Andererseits wird hier das File «Mostprob» erstellt (ursprünglich sind damit die «most probable cases» gemeint). Diese Datei ist quasi die Verlinkung von Elementen aus beiden Rohren. Die Datei bringt die Identifikatoren (ID, PERSNR) zurück zu den Dateninformationen. Zusätzlich werden nun die neuen Plausidaten hier angehängt. Wir haben also bei «Mostprob» ein File, die sowohl die Identität, die alte Plausi, die neue Plausi und optional alle Datenvariablen beinhaltet. Das File «Mostprob» führt somit alle Fäden zusammen. Hier kann man nun pro Zeile schauen was die alte und die neue Plausi aussagen, was das prädiktive ML-Algorithmus ausgegeben hat, was laut den Feedback-Mechanismen falsch sein soll und was die gelieferten Variablen genau waren. Das bedeutet auch, dass die Datenlieferanten in der geplanten Produktionsintegration Informationen aus diesem Ort erhalten werden – hier ist nämlich alles enthalten was man braucht. Natürlich müsste das dann dementsprechend formatiert werden, so dass es einerseits in die alte Plausi integrierbar ist und andererseits in einer für die Datenlieferanten nachvollziehbaren Struktur ausgegeben wird.

B.4 Modeling

B.4.1. Prediction

Die «Modeling» Phase besteht in unserem Pilotprojekt aus zwei Unterphasen. Einerseits haben wir den Prediction-Teil, wo wir erweiterte Baumalgorithmen wie «Gradient Boosting Machines» (GBM) und «Random Forests» (RF) versucht haben. Dabei haben wir mit GBM bessere Resultate erzielt als mit RF. Wir wollten auch andere Boosting Mechanismen wie «XGBoost» versuchen, hatten dafür aber keine Zeit mehr. XGBoost könnte die Accuracy nochmals leicht verbessern. Die «generalization accuracies» für die beiden GBM und RF Modelle waren die folgenden (mit einem 95% Vertrauensintervall):

Generalization Accuracy GBM: 0.94 (0.935 - 0.945)
Generalization Accuracy RF: 0.928 (0.923 - 0.933)

Die Ergebnisse der Modelle können in einer «confusion matrix» dargestellt werden (siehe folgende Tabelle), dabei werden die vorhergesagten Personalkategorien den tatsächlichen gegenübergestellt. Die Diagonale entspricht den korrekt vorhergesagten Beobachtungen.

	Reference				
Prediction	A	D	P	U	W
A	32824	807	3	115	904
D	724	23884	29	138	371
P	26	92	4791	118	97
U	165	243	110	17245	248
W	1553	796	103	319	15562
Für Personalkategorie A					
Sensitivity	0.9301	=32824/Summe(■)			
Specificity	0.9723	=Summe(□)/Summe(■)			

Tabelle: Confusion matrix (GBM-Modell) und Berechnung der Sensitivität und Spezifität für die Personalkategorie A

Daraus lässt sich die Sensitivität und die Spezifität des GBM-Modells für die fünf Personalkategorien ermitteln. Die Sensitivität zeigt an wie viele Personen einer Personalkategorie korrekt eingeordnet wurden. Für die Personalkategorie A entspricht sie 0.9301, d.h. 32'824 der real korrekten 35'292 Personen (hellgraue Fläche in der Tabelle).

Die Spezifität gibt an wie viele Personen, die nicht einer Personalkategorie angehören auch als solche erkannt wurden. In den Daten gehören 65'975 Personen nicht der Personalkategorie A an (dunkelgraue Fläche in der Tabelle), von diesen wurden 1829 (=807+3+115+904) fälschlicherweise in diese Personalkategorie eingeordnet, d.h. 0.0277. Die Spezifität ist dann $0.9723 = 1 - 0.0277$.

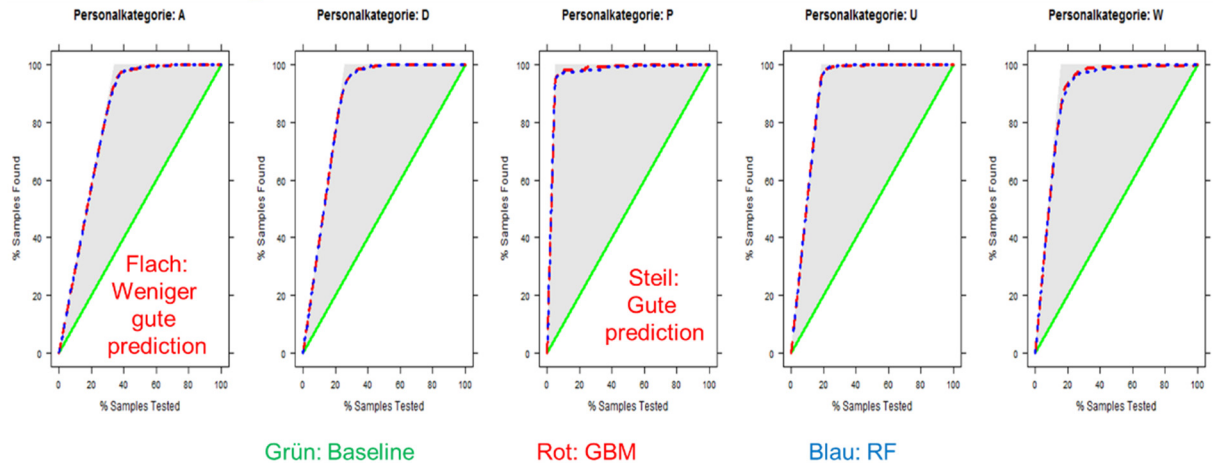
	Class: A	Class: D	Class: P	Class: U	Class: W
Sensitivity	0.9301	0.9249	0.95135	0.9615	0.9057
Specificity	0.9723	0.9833	0.99654	0.9908	0.9670

Tabelle: Sensitivität und Spezifität bei den fünf Personalkategorien (GBM-Modell).

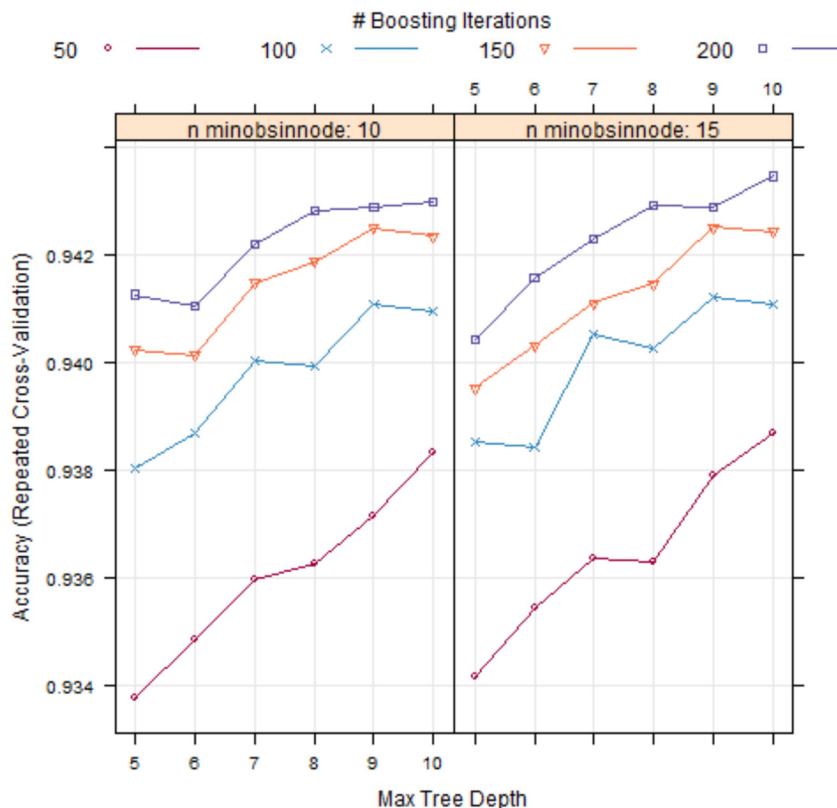
Es wird aus dieser Tabelle ersichtlich, dass gewisse Klassen wie Professoren und übrige Dozierende besser erkannt werden können. Gleichzeitig werden ihnen Personen am seltensten fälschlicherweise zugeordnet.

Die folgenden Gains-Kurven zeigen die Performanz der beiden Algorithmen GBM und RF im Vergleich zur Baseline. Zunächst sehen die beiden Kurven kaum unterscheidbar aus – nur in den vorhin genannten Werten lässt sich eine Unterscheidung feststellen. Man kann anhand dieser Kurven jedoch erkennen, dass die Modelle bei gewissen Personalkategorien besser funktionieren als bei anderen. Je steiler die Gains-Kurve, desto besser ist die Prediction bei dieser Personalkategorie. Demnach wird die Personalkategorie A am schlechtesten vorhergesagt. Dies ist auch keine Überraschung, wenn man bedenkt, dass wissenschaftliche Mitarbeiter (W) und assistierende (A) schwer voneinander zu unterscheiden sind (u. a. unterscheiden sie sich kaum in den Altersgruppen). Somit ist es auch verständlich, dass sowohl Mensch wie Maschine die Unterscheidung zwischen beiden schwer bewerkstelligen. Wenn W und A in eine gemeinsame Kategorie «wissenschaftliche Mitarbeiter und Assistierende» konvertiert werden, dann steigt auch die prädiktive Kraft.

Modellvergleich durch Gains-Kurven: Baseline / GBM / RF



Das oben verwendete Modell bedingt auch eine Wahl von Hyperparametern. Bei Hyperparametern handelt es sich um Vorgaben, an die das Modell sich halten muss. Ein Beispiel wäre die minimale Anzahl Beobachtungen, die ein neuer «Ast» eines «Baumes» haben soll. Die folgende Abbildung zeigt, wie systematisch unterschiedliche Werte für diese Hyperparameter getestet worden sind, um eine möglichst hohe Accuracy zu erhalten. So werden in der Abbildung drei der vier Hyperparameter dargestellt (für eine detaillierte Erklärung verweisen wir auf die Fachliteratur). Dabei stellt die Abbildung nicht alle unsere Tests dar, sondern nur eine Auswahl, die sich graphisch einfach darstellen lässt. Wie beim Train-Modul erklärt, haben wir sehr viele solcher «Gitter» nach und nach trainiert, um die besten Werte zu ermitteln.

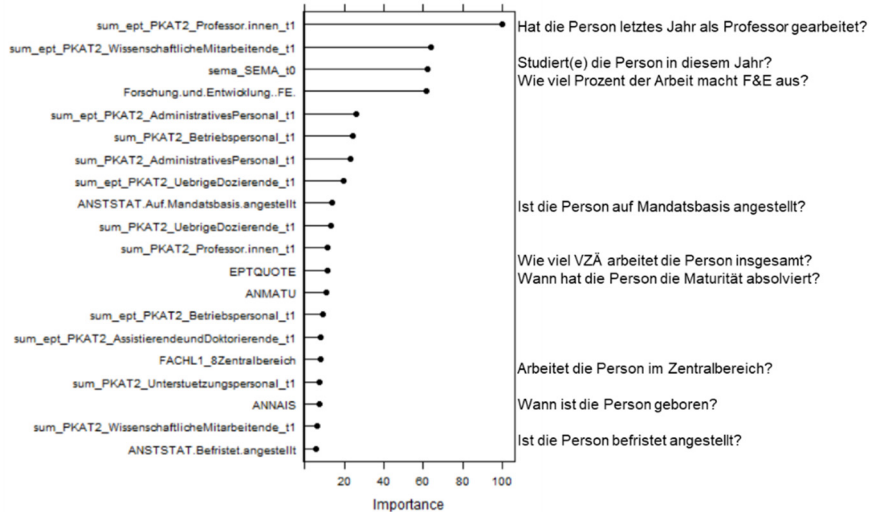


B.4.2. Explanation

Beim Thema «Explanation» geht es darum, die «blackbox» des ML-Algorithmus zumindest teilweise zu öffnen und zu verstehen, wie ein Resultat zustande gekommen ist. Dabei gibt es verschiedene Möglichkeiten. Eine «globale» Erklärung liefert beispielsweise die «variable importance». Dieser Wert gibt an, wie stark

die summierte prädiktive Kraft einzelner Variablen ist. Die folgende Graphik zeigt die «variable importance» für unser GBM-Modell.

Globale Erklärung: Variable Importance (bei GBM-Modell)



Die «variable importance» erklärt aber keinen individuellen Fall: Sie erklärt nicht, wieso eine Person als Professor oder wissenschaftlicher Mitarbeiter klassifiziert wird – sie sagt nur aus, dass beispielsweise das Alter eine wichtige Variable sein könnte. Wenn aber das Ziel ist, einem Datenlieferanten eine Erklärung für einzelne Beobachtungen melden zu können, muss auf die individuelle Ebene eingegangen werden können. Deshalb mussten wir nach weiteren Möglichkeiten Ausschau halten. Im Sommer 2018 haben wir uns den Stand der Forschung angeschaut und fanden LIME und DALEX als potentielle Lösungen.

B.4.3. DALEX

Das Package DALEX bietet mit dem Package «ceteris paribus» zusammen die Möglichkeit, den Einfluss einzelner Features auf die Wahrscheinlichkeit der Werte der abhängigen Variable zu ermitteln. Für jede Person liegt die Wahrscheinlichkeiten für die Zuweisung zu allen Personalkategorien vor, d.h. jede Person hat eine Wahrscheinlichkeit zwischen 0 und 100%, mit der sie einer Personalkategorie zugewiesen wird. In der Summe müssen diese Wahrscheinlichkeiten 100% ergeben.

Die «DALEX/ceteris paribus»-Funktion durchläuft nun für ausgewählte Variablen theoretisch mögliche Werte durch und berechnet, wie das Modell bei gleichbleibenden Werten für alle anderen Variablen die Wahrscheinlichkeiten für die Personalkategorien verschiebt. Eine Kombination aus Features, deren gleichzeitige Änderung einen Kategorienwechsel bringt, ist nicht möglich.

Die Abbildung links zeigt wie das Modell in Abhängigkeit des Geburtsjahrs eine Person in die unterschiedlichen Personalkategorien einordnet. In diesem Fall erfolgt die Einordnung als Professorin korrekt (die blaue Kurve zeigt die vorhergesagte Wahrscheinlichkeit der Personalkategorie «Professorin»). Wäre diese Person jünger (weiter rechts auf der Skala), dann kämen laut Modell auch die Kategorien wissenschaftliche Mitarbeitende (ab 1985) oder Assistierende (ab 1991) in Frage. Ausserdem reduziert sich die Wahrscheinlichkeit, dass diese Person Professorin ist, wenn der Forschungsanteil am Arbeitspensum steigt (mittlere Grafik der Abbildung).

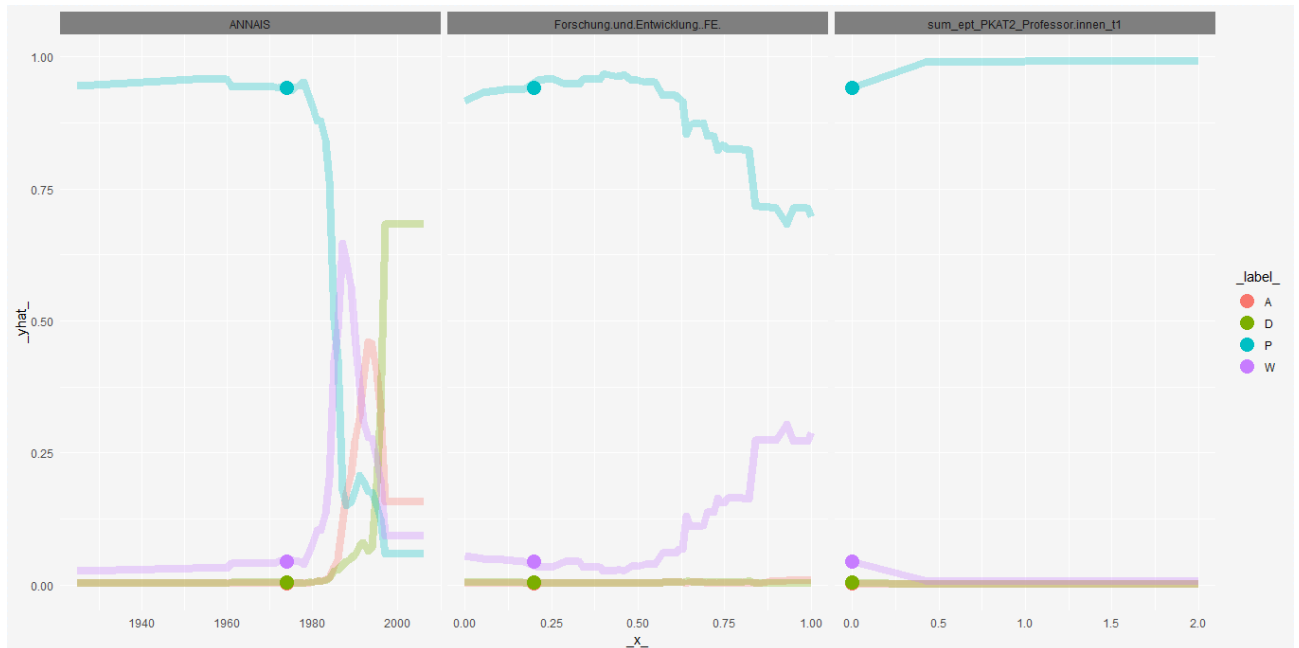


Abbildung Vorhergesagte Wahrscheinlichkeiten für die unterschiedlichen Personalkategorien (ohne Übrige Dozierende für eine bessere Übersicht). Der Punkt beschreibt den tatsächlichen Wert der jeweiligen Variablen. Die betrachtete Person ist in Wirklichkeit Professorin.

Zur Anwendung kam diese Funktion für die Auswahl der Personen, die durch das Modell einer anderen Personalkategorie zugeordnet wurden, als sie tatsächlich laut den gelieferten Daten hatten. Für jede Personalkategorie berechnete das Modell die Vorhersagewahrscheinlichkeit bei Wertänderung nur eines Features. Wenn die Wahrscheinlichkeit für die Personalkategorie, die tatsächlich auch geliefert wurde, höher als die Wahrscheinlichkeiten für die anderen Personalkategorien ist, dann wurde dieses Feature gemeldet.

Die folgende Abbildung zeigt einen Fall, in dem die Person dem Modell zufolge als wissenschaftliche Mitarbeitende angestellt ist, obwohl die Person eine Professorin ist. Eine korrekte Vorhersage könnte sich ergeben, wenn die Person bereits einige Zeit als Professorin beschäftigt wäre (Abbildung rechts), bei der Änderung anderer Features kommt es nicht zu einem Kategoriensprung in Richtung Professorin. Daraus kann geschlossen werden, dass entweder die Personalkategorie oder das untersuchte Feature fehlerhaft ist. Eine weitere Möglichkeit wäre, dass die Beobachtung korrekt ist, aber einen statistischen «Ausreisser» darstellt, den das Modell (noch) nicht korrekt einordnen kann.

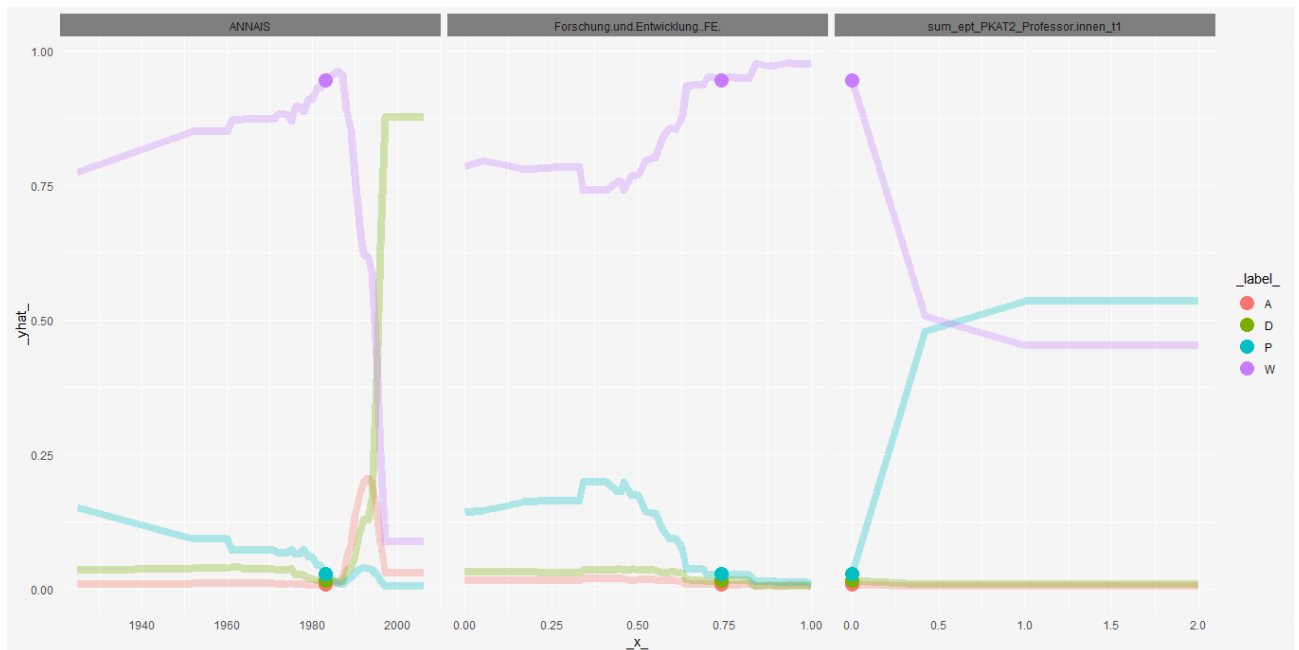


Abbildung Gelieferte Personalkategorie: Professorin, vorhergesagt: Wissenschaftliche Mitarbeitende (ohne Übrige Dozierende für eine bessere Übersicht).

Die Implementierung der «DALEX/ceteris paribus»-Funktion berücksichtigte Features, die jeweils im entsprechenden gelieferten Datensatz enthalten sind. Dabei werden für jedes analysierte Feature folgende Werte generiert:

- Höchste Wahrscheinlichkeiten für die einzelnen Personalkategorien bei unterschiedlichen Werten des Features
- Wert, des Features, der die Wahrscheinlichkeit der «wahrscheinlichsten» Personalkategorie maximiert

Existiert mindestens ein Feature für eine Beobachtung, die gemäss dem ML-Algorithmus eine fehlerhafte Personalkategorie aufweist, wird diese Beobachtung als «Kandidat» mit einem Flag versehen.

Die folgende Abbildung zeigt, wie viel fehlerhafte Beobachtungen Kandidaten sind. Für diese Beobachtungen existiert also mindestens ein Feature, dessen Wert fehlerhaft sein könnte, dessen Korrektur also die gelieferte und vorhergesagte Personalkategorie übereinstimmen liesse. Theoretisch könnten damit die Datenlieferanten auch auf andere Fehlerursachen aufmerksam gemacht werden. Interessant ist dabei, dass fast alle fehlerhaften Beobachtungen der Personalkategorie «Übrige Dozierende» auch Kandidaten sind, bei «Assistierenden» und «Wissenschaftlichen Mitarbeitenden» scheint es schwieriger zu sein, Features zu bestimmen, anhand derer die gelieferte Kategorie doch noch korrekt vorhergesagt werden könnte. Das «Erklärungspotenzial» dieses Ansatzes variiert offensichtlich in Abhängigkeit der gelieferten Personalkategorien, aber auch der Datenlieferanten selbst.



B.4.4. LIME

Beim Package LIME ("local interpretable model-agnostic explanations") handelt es sich um eine alternative Feedback-Methode, bei der ein lokales Modell berechnet wird. Also anders als bei DALEX, wo die Resultate aus dem Prädiktions-Modell verwendet werden, wird bei LIME für jeden Fall ein lokales lineares und ein somit einfacher zu interpretierendes erklärendes Modell berechnet. Die folgende Abbildung aus dem Originalartikel von Ribeiro *et al.* (2016; <https://arxiv.org/abs/1602.04938>) zeigt dies anschaulich. Das Ziel des ML-Algorithmus ist es in diesem Beispiel, «rot» oder «blau» vorherzusagen. Der betrachtete Fall wird durch das dicke rote Kreuz dargestellt. Die rote und blaue Fläche symbolisiert das komplexe Resultat aus

dem Prädiktionsmodell. Die anderen roten Kreuze und blauen Punkte symbolisieren die Punkte, die dem dicken roten Kreuz am nächsten sind und für die lokale Berechnung verwendet werden. Da ein lineares Modell berechnet wird, erkennt hier das Modell eine Grenze zwischen den roten Kreuzen und den blauen Punkten. Es ist eine vereinfachte lokale Erklärung, die man einem Menschen erklären kann. Sie bildet aber nicht die gesamte komplexe Realität ab – wie beschrieben wären das die roten und blauen Flächen, die nicht durch eine simple Linie erklärbar sind.

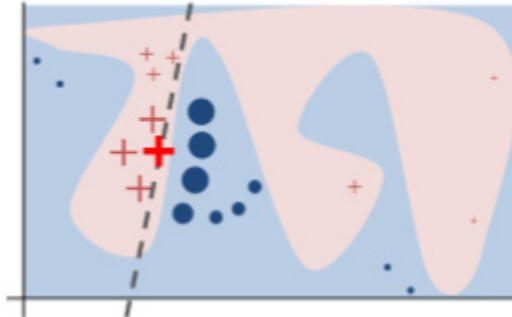
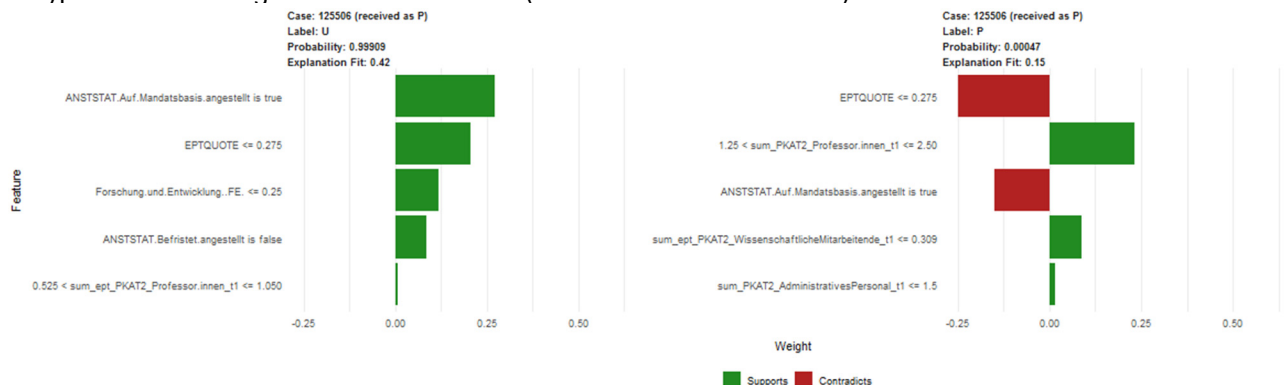


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f , and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

Die folgende Abbildung zeigt dabei ein Output aus den LIME-Berechnungen für unsere Beispiele. Wir haben den Standardoutput aus LIME stark verändert, damit es unseren Bedürfnissen im BFS entspricht. Dargestellt wird ein Fall, welcher als Professor geliefert wurde. Unser prädiktiver Algorithmus hat im ersten Teil berechnet, dass es sich hier nicht um einen Professor (da die vorhergesagte Wahrscheinlichkeit 0.00047% ist) sondern um einen übrigen Dozierenden mit einer Wahrscheinlichkeit von über 99% handelt. Nun sieht man links, ob gewisse verwendete Variablen für (grün) oder gegen (rot) die Kodierung als übrigen Dozierenden sprechen. Rechts sieht man wiederum gewisse Variablen, die für die Kategorie Professor sprechen würden. Wie man auf den ersten Blick sieht, ist das Resultat links durchwegs grün und bestätigt das prädiktive ML-Modell. Rechts sieht es gemischter aus für die Kategorie Professor. So hat die Person beispielsweise eine untypische Anstellungsintensität unter 27% (VZÄ-Stelle von unter 27%).



Dieses LIME-Ergebnis kann nun von den BFS-Mitarbeitenden verwendet werden, um das prädiktive Modell genauer anzuschauen und die genauen Gründe für das berechnete Resultat eruieren. Wir sind nun leider nicht weit genug gekommen, um dieses Ergebnis (wie bei DALEX) den Datenlieferanten zurückspielen zu können. Wir haben versucht eine Art «Wahlalgorithmus» zu erstellen, bei dem die oben visualisierten Informationen vermittelt werden (beispielsweise, wenn eine Variable besonders stark abweicht). Wir konnten dies aber noch nicht formalisieren.

Dafür haben wir versucht den LIME-Algorithmus weiter zu verbessern. Dabei haben wir ein Schalenmodell

konstruiert, welches wir hier erläutern. In der folgenden Abbildung wird bereits gezeigt, wie wir das oben beschriebene Vorgehen nicht einmal, sondern dreimal machen – nämlich für drei «Schalen». Alle drei Schalen zeigen drei verschiedene lokale Modelle mit verschiedenen Variablen. Im nächsten Abschnitt erklären wir die drei Schalen und die Wahl der Variablen im Detail.



B.4.5. Erklärung des Schalenmodells

Wie weiter oben beschrieben, führt der zusätzliche Einbezug der «expert variables» dazu, dass wir mit den 1000 dadurch generierten Variablen die Accuracy der Prädiktion deutlich erhöhen können. Diese vielen «expert variables» sind aber, wie der Name schon sagt, nicht für die Datenlieferanten gedacht und erschweren die Interpretation der Resultate tendenziell. Teilweise können sogar Experten im BFS mit diesen sehr abstrakten Variablen nichts unmittelbar anfangen. Stattdessen muss deshalb in der Feedback-Phase ein reduziertes Objekt mit weniger Variablen generiert werden, welches für Menschen (sowohl Datenlieferanten wie auch BFS-Mitarbeitende) nachvollziehbar ist. Dazu werden drei Schalen erstellt.

In der innersten Schale sind nur die Variablen drin, welche die Datenlieferanten uns schicken. Bei der äussersten Schale wählen wir die Top-X Variablen aus. Die mittlere Schale ist für BFS-Mitarbeiter gedacht.

1) Schale 1: “No humans in the loop”:

Bei dieser Schale werden die Top-X Variablen aus dem weiter oben beschriebenen Durchschnitt der «variable importances» genommen. Momentan werden 30 Variablen gewählt, aber man kann diesen Wert auch reduzieren.

Die "Top"-Schale hilft uns zu bestimmen, welche die am besten globalen Variablen für die Prädiktion sind. Aber es sind viele «expert variables» dabei, die menschlich nicht nachvollziehbar sind, und mit denen man eigentlich den Feedback-Mechanismus für Externe nicht implementieren kann.

2) Schale 2: “Statistical Institution human in the loop”:

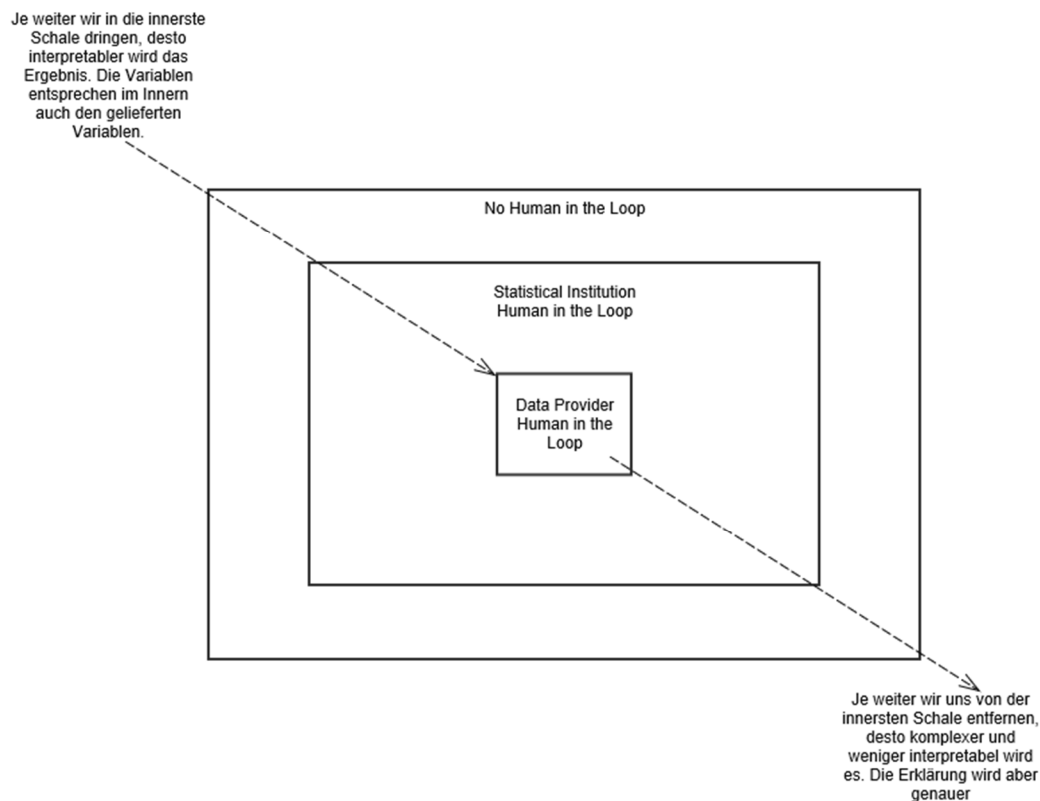
Die generelle Idee wäre also "nicht-sinnvoll interpretierbare" Variablen aus dem Datenset zu entfernen. Dafür ist Schale 2 implementiert. Diese haben wir so gestaltet, dass sie den Bedürfnissen der BFS-Mitarbeitenden entsprechen, die Analysen durchführen. Es beinhaltet somit keine «expert variables» aber verknüpfte Variablen, Variablen mit Aussagekraft zu vergangenen Jahren, von Fachexperten generierte Variablen usw.

Momentan bleiben 44 Variablen übrig in der mittleren Schale.

3) Schale 3: “Data-Provider human in the loop”:

Das Datenset wird hier komplett auf die gelieferten Daten reduziert.

Die folgende Graphik zeigt die Überlegung der Schalen in einer Graphik. Dabei gilt: Je weiter wir in die innerste Schale dringen, desto verständlicher wird die Erklärung. Die Variablen entsprechen in der innersten Schale auch den gelieferten Variablen. Je weiter wir uns von der innersten Schale entfernen, desto komplexer (und unmöglicher) wird die Interpretation - die Prädiktion hingegen wird genauer. In der äussersten Schale sind wir in der Schale, die viele «expert variables» enthält, die zwar die Prädiktion verbessern, aber auch die menschliche Interpretabilität reduzieren. In der mittleren Schale befinden sich die Variablen, die für BFS-Mitarbeitende noch verständlich sind, aber teilweise den Datenlieferanten nicht oder nur schwer interpretiert werden können⁴. In dieser Schale können wir die Daten analysieren und schauen, ob die Prädiktion Sinn macht oder nicht. Aber wenn wir am Ende ein Feedback an die Datenlieferanten liefern, dann befinden wir uns strikt in der innersten Schale mit den gelieferten Variablen.



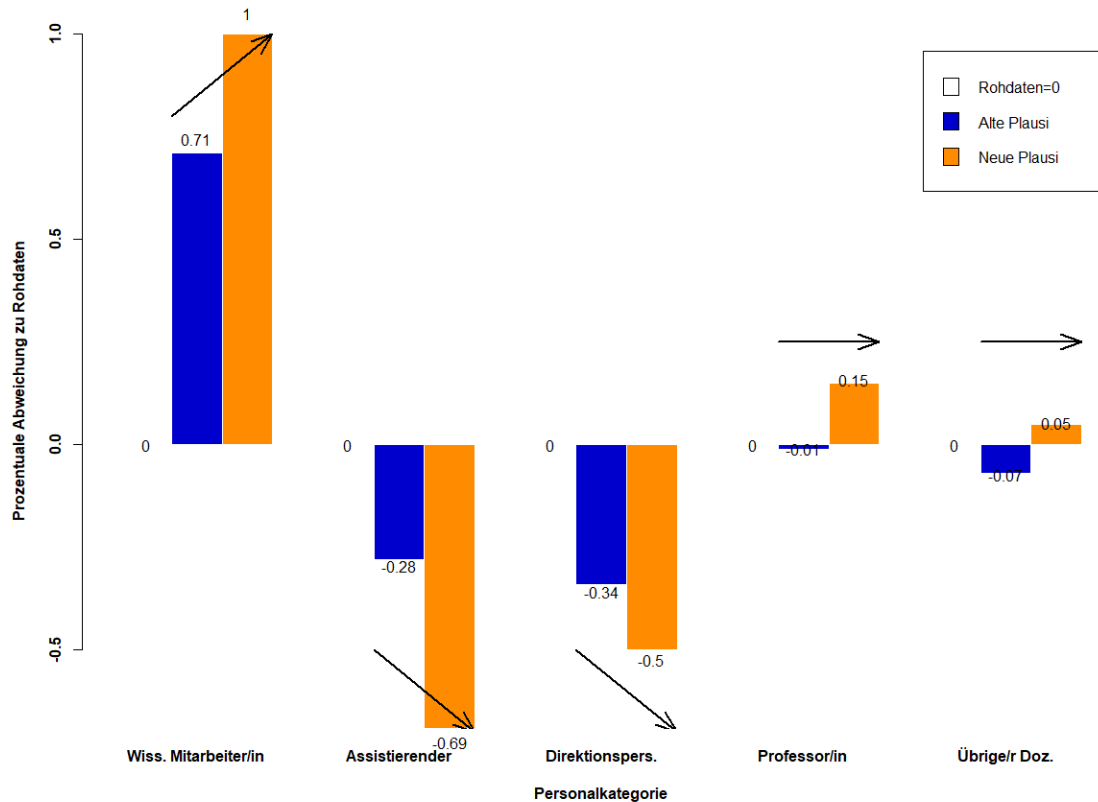
B.5 Evaluation

B.5.1. Analyse von systematischen Fehlern auf aggregierter Ebene

Eine Analyse auf aggregierter Ebene kann helfen systematische Fehler zu entdecken. Zunächst haben wir überprüft, ob sowohl die alte wie auch die neue Plausi einen Effekt auf die Verteilung der verschiedenen Personalkategorien haben. Sprich: Kann es sein, dass gewisse Personalkategorien systematisch zu häufig

⁴ Hinzu kommt, dass Datenlieferanten aus Gründen des Datenschutzes nicht alle Werte ihrer Angestellten wissen dürfen.

oder zu selten von den Datenlieferanten kodiert werden? Die folgende Abbildung zeigt die Veränderung der Verteilungen der fünf Personalkategorien in diesen drei Datensets. Dabei wird ersichtlich, dass sich die Verteilung mit den Plausis verändern würde, und zwar nicht auf eine zufällige Art, sondern beide Plausis verändern die Daten in dieselbe Richtung – nur zeigt die neue Plausi diesen Trend stärker als die alte, was ein Indiz auf eine bessere inhaltliche Fehlerfindung sein könnte. Konkret zeigt es, dass die Datenlieferanten die Personalkategorie «wissenschaftlicher Mitarbeiter» zu selten kodieren, und die Kategorien «Assistierende» und «Direktionspersonal» zu häufig⁵.



Der aggregierte Effekt birgt aber auch pro Universität interessante Resultate. Die Fachexperten im Bereich Hochschulpersonal wissen bereits aus den bisherigen veröffentlichten Daten, dass es eine grosse Heterogenität zwischen den Universitäten gibt. Das spiegelt sich aber auch in den folgenden Resultaten wieder.

So werden in der nächsten Tabelle die Veränderungen zwischen den Verteilungen in den konsolidierten Daten (nach alter Plausi und Validierung der Datenlieferanten) und den Rohdaten ersichtlich. Dabei sind nur die Veränderungen, die über zwei Prozentpunkte ausmachen, rot eingezeichnet: bei der Hochschule F und bei der Hochschule L. Dabei haben bei diesen zwei Fällen jeweils eine Personalkategorie einen positiven Wert über 2% und einen negativen Wert über -2%. Bei der Hochschule F sieht man, dass die Rohdaten dementsprechend modifiziert wurden, so dass es 2.78% mehr wissenschaftliche Mitarbeiter in den veröffentlichten Daten gab. Diese Werte wurden augenscheinlich bei den assistierenden abgezogen (-3.23%). Bei der Hochschule L findet sehr wahrscheinlich eine Umverteilung von Direktionspersonal zu Assistierenden statt. Es scheint somit nicht zufällig zu sein: Bei der Kodierung der Personalkategorien scheint es hochschulspezifische Besonderheiten zu geben.

	A	D	P	U	W
Hochschule A	-0.03	-0.09	0.14	0.41	-0.41
Hochschule B	-0.13	-0.36	-0.01	-0.15	0.66

⁵ Die tatsächliche Abweichung der Verteilung hängt am Schluss auch davon ab, wie viele vom ML-Algorithmus vorgeschlagene Korrekturen in Zusammenarbeit mit den Datenlieferanten dann tatsächlich auch vorgenommen werden. Es muss sich selbstverständlich nicht bei allen vorhergesagten Fehlern wirklich um Fehler handeln.

DI Pilotprojekt Plausi++ BB

Hochschule C	0	0	0	0	0
Hochschule D	1.76	-1.12	0.12	1.22	-1.97
Hochschule E	1.22	-0.73	-0.02	-0.4	-0.07
Hochschule F	-3.23	1.18	-0.27	-0.46	2.78
Hochschule G	0.94	-0.81	0.09	-0.19	-0.03
Hochschule H	0.23	0.09	0.13	0.92	-1.36
Hochschule I	-0.16	-0.66	-0.07	0.33	0.56
Hochschule J	-1.17	0.13	-0.05	0.45	0.63
Hochschule K	0.79	0.7	-0.12	-1.25	-0.12
Hochschule L	2.45	-2.85	0.3	-0.26	0.36
Hochschule M	0.14	-1.15	-0.16	-0.12	1.3

Tabelle: Differenz in Prozentpunkten zwischen Rohdaten und konsolidierten Daten

Das Interessante und Neue ist nun der Unterschied zwischen den Rohdaten und der neuen Plausi. Diese Werte werden in der nächsten Tabelle dargestellt. Hier sind dieselben Werte wie oben in Rot eingezeichnet plus weitere Werte in Gelb, die nun ebenfalls einen Unterschied über zwei Prozentpunkte ausmachen. Zunächst sind die roten Werte grösser als oben bei den konsolidierten Daten, das bedeutet, dass die Umverteilung durch die Korrektur dort grösser wäre. Die Richtung der Umverteilung ist aber dieselbe und es sind dieselben Personalkategorien betroffen, während die anderen weniger stark verändert werden. Es ist auch ersichtlich, dass es bei den meisten Fällen zwei Kategorien gibt, die sich in etwa aufheben. Das könnte tatsächlich ein Hinweis darauf sein, dass systematische Fehler bei gewissen Unis gemacht wurden. Das ist ein wichtiger Punkt, da ansonsten Universitäten, die strukturell anders sind automatisch «Korrekturen» zur Folge hätten, ohne dass diese gerechtfertigt wären.

	A	D	P	U	W
Hochschule A	-0.15	-0.42	0.15	0.07	0.37
Hochschule B	-0.08	-0.63	0.07	-0.11	0.76
Hochschule C	-3.13	0	0.09	0.71	2.34
Hochschule D	-3.84	-0.29	0.29	3.54	0.3
Hochschule E	0.43	-1.62	0.38	0.04	0.77
Hochschule F	-3.69	0.98	-0.21	-0.14	3.05
Hochschule G	0.42	-1.33	0.27	0.51	0.12
Hochschule H	0.84	-0.33	0.08	-0.02	-0.56
Hochschule I	0.19	-0.91	0.42	-0.63	0.93
Hochschule J	-1.66	0.55	0.62	-0.41	0.9
Hochschule K	0.62	-0.37	0.25	0.18	-0.68
Hochschule L	-3.32	1.83	0.54	1.8	-0.85
Hochschule M	-0.23	-1.57	0.12	0.07	1.6

Tabelle: Differenz in Prozentpunkten zwischen Rohdaten und Plausi++ Vorhersagen

Daraus kann man folgendes ableiten.

- Plausi++ generiert einen Korrekturtrend, der in die gleiche Richtung wie die alte Plausi geht. Dieser würde sogar etwas stärker ausfallen, was aber auch daran liegen könnte, dass die konsolidierten Daten auch auf den Validierungen der Hochschulen beruhen.
- Auf Level der Universitäten findet Plausi++ die gleichen beiden aggregierten Fehlerquellen wie die alte Plausi und weist auf zwei weitere Fehlerquellen hin.
- Generell weisen die Resultate aber auch darauf hin, dass die Korrekturen von Plausi++ nicht zufällig sind: es gibt auf aggregierter Ebene einerseits klare Muster. Diese fallen ausserhalb einer zufälligen Verteilung. Andererseits gibt es auch das Fehlen von Mustern bei anderen Personalkategorien und Hochschulen, was darauf hindeutet, dass bei einigen Universitäten und Personalkategorien die Plausi keinen systematischen Anlass zur Korrektur in eine Richtung gibt. Beide Elemente sind posi-

tiv zu bewerten, da die Plausibilitätskontrolle wahrscheinlich genau diese systematischen Fehler entdecken soll.

B.5.2. Weitere Qualitätskontrolle

Bei der Vorhersage der Personalkategorie für ein neues Jahr / eine neue Datenlieferung, gibt es drei Fälle, die pro Person möglich sind:

- Die Personalkategorie hat sich bislang nie geändert
- Es gab einen Wechsel der Personalkategorie in den Vorjahren
- Die Person wird zum ersten Mal als Personal erfasst

	Personen mit Vergangenheit ohne PK-Wechsel	Personen mit Vergangenheit und PK-Wechsel	Personen ohne Vergangenheit	Alle Personen
pred != obse	3279	1458	2224	6961
pred == obse	61091	19336	13879	94306
Total	64370	20794	16103	101267
%-Anteil korrekt	95%	93%	86%	93%

	Personen ohne Vergangenheit	Personen mit Vergangenheit und PK-Wechsel	Personen mit Vergangenheit ohne PK-Wechsel	Alle Personen
pred != obse	2224	1458	3279	6961
pred == obse	13879	19336	61091	94306
Total	16103	20794	64370	101267
	14%	7%	5%	7%

Der erste Fall wird durch das Modell überdurchschnittlich gut abgebildet (95%). Die Modellierung des zweiten Falles erfolgt nahe am Modelldurchschnitt (93%). Hier wäre die Berücksichtigung von Hochschulwechseln als eigenes Feature vielleicht von Bedeutung, da damit häufig auch ein Wechsel der Personalkategorie einhergehen kann. Der dritte Fall wird unterdurchschnittlich (86%) gut abgebildet. Für die Entscheidung, ob eine getrennte Modellierung der drei Fälle notwendig ist, müsste der Aufwand einer getrennten Modellierung in Bezug gesetzt werden. Ohne zusätzliche Modellierung könnte aber die Information, zu welchem Fall eine Person gehört in das Feedback miteinbezogen werden, so dass das Vorkommen des dritten Falles zuvor zusätzlich analysiert wird, z.B. ob ein Unterschied zwischen den Personalkategorien vorliegt.

B.6 Deployment

Im Projekt haben wir auch das «Deployment», also die Produktionsintegration, angedacht, auch wenn es auf vielen Annahmen und theoretischen Überlegungen beruht, da wir nicht wissen, wie beispielsweise die IT-Infrastruktur im BFS umgesetzt werden wird. Zum Zeitpunkt des Pilotprojekts wurde die gesamte Entwicklung auf einem Atlantica Server durchgeführt, der komplett vom BFS abgetrennt ist.

Für die Produktionsintegration, können die beiden Phasen des ML-Prozess getrennt betrachtet werden. Einerseits gibt es die Training-Phase, bei der der ML-Algorithmus durch die älteren Daten trainiert wird. Diese Phase ist rechenintensiv und speicherintensiv, kann aber zeitlich weit vor der eigentlichen Erhebung durchgeführt werden. Das Resultat dieser Phase ist ein 'Modell-Objekt'. In der zweiten Phase wird das 'Modell-Objekt' verwendet, um auf die neuen Daten anzuwenden und die entsprechenden Resultate (in unserem Fall primär die Fehlerwahrscheinlichkeiten) zu erhalten. Das wichtige ist, dass diese Phase nicht rechenintensiv ist. Theoretisch könnte man dieses Modell-Objekt in die aktuelle BFS-Umgebung «kopieren» und dort 'ausführen'. Für die Ausführung braucht es idealerweise dieselbe Programmiersprache, um sicher zu gehen, dass keine Fehler entstehen. Aber theoretisch könnte man dieses Modell-Objekt auch mit einer anderen Programmiersprache ausführen, wenn es in einem standardisierten Format abgespeichert wird.

Eine weitere und bessere Möglichkeit wäre es aber eine Schnittstelle auf dem Server zu erstellen. Das bedeutet, dass die neuen Daten an den Server (an die Schnittstelle) gesendet werden, dort verarbeitet werden und die Resultate/Feedbacks danach über die Schnittstelle zurück an die BFS-Umgebung geschickt werden.

Theoretisch könnte man somit auf diesen beiden Wegen die Plausibilisierung der neuen Daten in die bestehenden Pipelines einbauen und somit automatisieren. Es würde bedeuten, dass die aktuelle Pipeline an einem Punkt unterbrochen wird und über die Schnittstelle geleitet wird. Der Stream zurück würde dann an die bestehende Pipeline integriert.

Auch mit den Metadaten entsteht somit kein Problem. Wenn die Metadaten auf den Berechnungen auf dem «ML-Server» benötigt werden, dann könnten die aktuellsten Metadaten über die Schnittstelle ebenfalls abgefragt werden.

Technisch kann eine Schnittstelle auf einem Linux-Server durch eine «REST-API» implementiert werden.

B.7 Monitoring

Die Produktionsintegration benötigt aber auch die Einrichtung eines vorausschauenden Monitorings. Wir haben bereits verschiedene Überlegungen dazu angestellt, da periodische Überprüfungen und Aktualisierungen des Modells notwendig sind.

Der Algorithmus sollte periodisch (beispielsweise jährlich) aktualisiert werden, damit die Accuracy konstant hoch bleibt (oder sogar noch steigt, da mehr Daten verfügbar sind). Das Modell sollte somit neu trainiert werden. Der Programmieraufwand sollte dabei gleich Null sein, da die Scripts bereits geschrieben sind und nur ausgeführt werden müssten, damit das Modell neu trainiert wird beziehungsweise aktualisiert wird. Das Trainieren benötigt jedoch eine starke Rechenleistung für mehrere Tage. Durch eine solche Aktualisierung bleibt nicht nur die Accuracy hoch, sondern es werden auch neue Datentrends (gesellschaftliche, juristische, strukturelle Veränderungen) berücksichtigt. Beispielsweise fällt das Durchschnittsalter einer weiblichen Professorin an einer ETH beinahe jährlich. Eine Aktualisierung der Traindaten würde somit die Realität besser berücksichtigen, da die dazugehörenden Wahrscheinlichkeiten angepasst werden.

Gleichzeitig kann ein Monitoring verwendet werden, um systematische Fehler oder Outlier zu entdecken. Für das BFS ist es wichtig, solche systematischen Elemente zu erwähnen und mit den Universitäten zur Sprache zu bringen. Ein Beispiel für solche systematischen Elemente wurde weiter oben bei den aggregierten Elementen erwähnt. Die Outputs, auch die Feedback-Mechanismen, unseres Projekts können bereits wichtige Elemente für ein derartiges internes Monitoring bereitstellen. Somit haben wir ebenfalls bereits die ersten Weichen für ein Monitoring gesetzt und müssten dies noch ausweiten.

Ein Monitoring kann man auch im Bereich der technischen Infrastruktur andenken. Wenn ein ML-Modell trainiert wird, dann werden mehr Rechenpower und Speicherkapazität benötigt. Das bedeutet, dass sich Spitzenzeiten entwickeln, in der besonders viel gerechnet wird – es kann sich in unseren Fällen dabei um Tage handeln. In anderen Projekten kann es sich auch um Wochen handeln. Beim Atlantica Server, welches leider auf einem Windows Server lief, mussten wir auch manchmal Serverabstürze miterleben. Bei einem stabilen Server sollte so etwas nicht passieren, da automatisch die Last ausbalanciert wird und die Berechnungen vielleicht länger dauern, aber kein Absturz passieren sollte. Die Ausbalancierung geschieht auch durch den Fakt, dass unser Code 'parallel computing' verwendet, so dass verschiedene Prozessorenkerne gleichzeitig rechnen. Ein stabilerer (Linux) Server achtet dabei, dass der benötigte RAM nicht den verfügbaren RAM überschreitet, was zu Abstürzen führen kann.