Schmidt, Elena

Article

Korrektur des Tätigkeitsschlüssels der Bundesagentur für Arbeit mithilfe maschineller Lernverfahren

WISTA – Wirtschaft und Statistik

Provided in Cooperation with:
Statistisches Bundesamt (Destatis), Wiesbaden

Suggested Citation: Schmidt, Elena (2020) : Korrektur des Tätigkeitsschlüssels der Bundesagentur für Arbeit mithilfe maschineller Lernverfahren, WISTA – Wirtschaft und Statistik, ISSN 1619-2907, Statistisches Bundesamt (Destatis), Wiesbaden, Vol. 72, Iss. 6, pp. 37-47

This Version is available at:
https://hdl.handle.net/10419/228501

KORREKTUR DES TÄTIGKEITS-SCHLÜSSELS DER BUNDESAGENTUR FÜR ARBEIT MITHILFE MASCHINELLER LERNVERFAHREN

Einsatz der Verdienststrukturerhebung

Dr. Elena Schmidt

Dr. Elena Schmidt

ist Bioinformatikerin und als wissenschaftliche Mitarbeiterin im Referat „Maschinelles Lernen und Imputationsverfahren“ des Statistischen Bundesamtes tätig. Ihre Aufgabenschwerpunkte sind die Weiterentwicklung von maschinellen Lernverfahren und ihre Anwendung auf Daten der amtlichen Statistik.

Schlüsselwörter: Maschinelles Lernen – Boosting – Tätigkeitsschlüssel – Verdienststrukturerhebung

ZUSAMMENFASSUNG

Ziel des vorgestellten Projekts war, das Merkmal Vollzeit/Teilzeit des Tätigkeitsschlüssels in den Integrierten Erwerbsbiografien der Bundesagentur für Arbeit zu korrigieren. Dies sollte mithilfe des auch in der Verdienststrukturerhebung vorhandenen, aber manuell korrigierten Schlüssels erfolgen. Unter Berücksichtigung verschiedener projektspezifischer Anforderungen wurde ein überwachtes maschinelles Lernverfahren eingesetzt, welches das entsprechende Merkmal eines Beschäftigten anhand vorliegender Betriebs- und Mitarbeitermerkmale schätzen kann. Es zeigte sich, dass sich mit diesem Modell der Fehler in der Signierung des Tätigkeitsschlüssels bei einer aus dem Datensatz der Verdienststrukturerhebung erzeugten Testmenge um etwa 40 % reduzieren lässt.

Keywords: *machine learning – boosting – occupational code number – structure of earnings survey*

ABSTRACT

The aim of this project was to correct the “part-time/full-time” variable component of the occupational code number in the Integrated Labour Market Biographies of the Federal Employment Agency. The basic idea was to use the code number which is also employed in the structure of earnings survey but corrected manually. Against the background of various project-specific requirements, a supervised machine learning method was applied which can estimate the relevant variable for an employee based on the variables available for the local unit and the employee. A test of the model on a data subset of the structure of earnings survey showed that it can reduce the coding error in the occupational code number by about 40 %.

1

Einleitung

In den Integrierten Erwerbsbiografien (IEB) der Bundesagentur für Arbeit (BA) liegt die Information über die Arbeitszeit eines Beschäftigten als kategoriales Merkmal (Vollzeit/Teilzeit) innerhalb des Tätigkeitsschlüssels vor. Eine Änderung der statistischen Erfassung von Teilzeitarbeit (Bertat und andere, 2013) im Jahr 2011 machte deutlich, dass eine Untererfassung von Teilzeitbeschäftigten in den Jahren vor 2011 vorlag. Dies ist problematisch, da die Integrierten Erwerbsbiografien oft die Grundlage für Arbeitsmarktanalysen bilden (Fitzenberger/Seidlitz, 2020) und die Untererfassung zu fehlerhaften Auswertungen wie der Überschätzung des Niedriglohnanteils führt. Aus diesem Grund wurden bereits verschiedene Ansätze zur Korrektur entwickelt (Möller, 2016; Fitzenberger/Seidlitz, 2019).

In der Verdienststrukturerhebung (VSE) 2014 (Statistisches Bundesamt, 2016) hingegen liegen neben dem BA-Tätigkeitsschlüssel auch die bezahlten Arbeitsstunden als Erhebungsmerkmal vor. Diese konnten bei einer manuellen Korrektur im Rahmen der Plausibilitätsprüfung des Tätigkeitsschlüssels in der Verdienststrukturerhebung 2014 mitberücksichtigt werden. Somit ist von einer hohen Qualität der VSE-Daten auszugehen. Eine direkte Übertragung des korrigierten Schlüssels der VSE-Daten zur entsprechenden Korrektur der IEB-Daten ist allerdings nicht möglich, da es sich bei der Verdienststrukturerhebung um eine Stichprobe handelt. Aus Geheimhaltungsgründen wäre es aber auch rechtlich untersagt.

Das Ziel des Projekts war, den manuell korrigierten Tätigkeitsschlüssel des VSE-Datensatzes für eine Korrektur des entsprechenden Schlüssels der Integrierten Erwerbsbiografien zu nutzen. Die korrigierten VSE-Daten wurden als Trainingsdaten für ein überwachtes maschinelles Lernverfahren (ML-Verfahren) verwendet, welches das Merkmal Vollzeit/Teilzeit einer oder eines Beschäftigten anhand vorliegender Unternehmens- und Mitarbeitermerkmale schätzen kann. Dieses ML-Modell soll dem Institut für Arbeitsmarkt- und Berufsforschung der Bundesagentur für Arbeit zur Verfügung gestellt und dort auf die IEB-Daten angewendet werden, um falsche Kodierungen im Tätigkeitsschlüssel aufzudecken.

Kapitel 2 beschreibt zunächst den für die Berechnung des ML-Modells verwendeten Datensatz und erläutert die Besonderheiten und sich daraus ergebenden Anforderungen des vorliegenden Problems. Kapitel 3 geht näher auf die Grundlagen der verwendeten Methoden, das maschinelle Lernverfahren Tree Boosting und die zur Modellbewertung eingesetzte Metrik ein. Kapitel 4 stellt das Vorgehen im Hinblick auf die in Kapitel 2 genannten Anforderungen genauer dar. Die Kapitel 5 und 6 zeigen auf, mit welchem Erfolg der Einsatz des verwendeten maschinellen Lernverfahrens möglich ist, und weisen auf weitere Optionen bei dessen Einsatz hin.

2

Daten und Problemstellung

Das Training erfolgte auf dem VSE-Datensatz und das Modell soll anschließend auf den IEB-Daten angewendet werden. Daher waren nur die Merkmale, die in beiden Datensätzen enthalten sind, für das Training nutzbar. Diese sind die Betriebsmerkmale „Anzahl der Mitarbeiter eines Betriebes“, „Bundesland“, „Landkreis“ und „Wirtschaftszweig“ und die Mitarbeitermerkmale „Geschlecht“, „Alter“, „Dauer der bisherigen Beschäftigung“, „Bruttomonatsverdienst“, „Personengruppenschlüssel“ und „Tätigkeitsschlüssel“. Letzterer liefert Informationen über den Beruf, die Spezialisierung, das Arbeitsniveau, die Schulbildung, die Berufsausbildung und das Arbeitsverhältnis (einschließlich der fehlerhaften Vollzeit/Teilzeit-Kodierung).

Eine festgestellte Untererfassung der Teilzeitbeschäftigten wirkt sich einschränkend auf die Niedriglohnberechnung aus. Daher ist für die Auswertung der IEB-Daten für das Institut für Arbeitsmarkt- und Berufsforschung der Fall der Teilzeitbeschäftigten, die fälschlicherweise als Vollzeitbeschäftige geführt werden, von besonderer Bedeutung. Die hier gezeigten Rechnungen beziehen sich ausschließlich auf diesen Fall, es wurden daher nur in der Verdienststrukturerhebung geführte Beschäftigte in den Datensatz aufgenommen, die vor der manuellen Korrektur als Vollzeitkräfte geführt wurden.

Um eine Überanpassung eines ML-Modells an den Trainingsdatensatz zu erkennen, ist es üblich, den für die Erstellung eines Modells zur Verfügung stehenden Datensatz in einen Trainings- und einen Testdatensatz

aufzuteilen. Der Trainingsdatensatz wird für die Berechnung des Modells verwendet, mit dem Testdatensatz wird die Klassifikationsgüte des Modells überprüft. Dies führte bei einer 80 %/20 %-Teilung des Datensatzes zu einem Trainingsdatensatz mit 389 298 Beschäftigten und einem Testdatensatz mit 97 324 Beschäftigten.

Aus der beschriebenen Situation und der Beschaffenheit der Daten ergaben sich eine Reihe von Anforderungen für die Umsetzung des Projekts:

1. Geheimhaltung. Da das Ziel des Projekts die Berechnung eines ML-Modells war, das vom Institut für Arbeitsmarkt- und Berufsforschung eingesetzt werden kann, musste gewährleistet sein, dass mit dem Modell keine Einzeldaten aus der Verdienststrukturerhebung weitergegeben werden.
2. Die Struktur eines Trainingsdatensatzes sollte möglichst ähnlich zu den Daten sein, auf die ein entsprechendes Modell angewendet werden soll. Daher war zu berücksichtigen, dass es sich bei den IEB-Daten um eine Vollerhebung handelt, bei der Verdienststrukturerhebung dagegen um eine Stichprobe.
3. Die Teilzeitbeschäftigten machen nur etwa 5 % der Beschäftigten im Datensatz aus. Wenn eine vorherzusagende Klasse im Datensatz sehr viel stärker vertreten ist als die andere, spricht man von einem „Imbalanced Data Problem“. Beim Trainieren eines ML-Modells kann in einem solchen Fall ein Bias in Richtung der Mehrheitsklasse auftreten. Diesem Effekt sollte, falls nötig, entgegengewirkt werden.
4. Da das Ziel die Korrektur des Tätigkeitsschlüssels im IEB-Datensatz war, war zu berücksichtigen, dass der Gesamtanteil der fehlerhaft kodierten Tätigkeitsschlüssel nach der Korrektur mithilfe des ML-Modells geringer sein sollte als vorher.

3

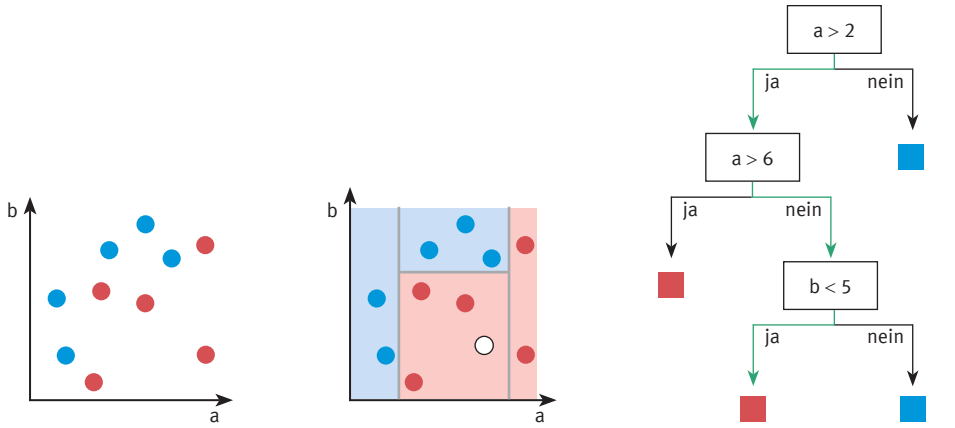
Methodik

Die Anforderung, dass keine Einzeldaten mit dem berechneten ML-Modell weitergegeben werden durften, schränkte die Auswahl der einsetzbaren ML-Verfahren bereits ein. Support Vector Machines (SVMs) können diese Anforderung nicht erfüllen, da einzelne Datenpunkte Teil der Modelldefinition und damit bei einer Weitergabe des Modells aus diesem auslesbar sind. Für den Vergleich mit den anderen Verfahren wurden SVMs aber zunächst mitberücksichtigt. Für die Modellerstellung wurden letztlich Testrechnungen mit den vier verschiedenen (ML-)Verfahren Logistische Regression, SVMs, Random Forests und Gradient Boosting durchgeführt. Gemessen an der zu optimierenden Zielgröße konnte mit Gradient Boosting das beste Modell berechnet werden, sodass die weiteren Untersuchungen mit diesem Verfahren durchgeführt wurden.

3.1 Gradient Boosting

Boosting (Schapire, 1990; Freund/Schapire, 1996; Breiman, 1998; Schapire/Freund, 2012) ist ein maschinelles Lernverfahren, bei dem eine Menge von schwachen Klassifikatoren (Base Classifiers) zu einem neuen starken Klassifikator kombiniert wird. Es gehört zu den überwachten Lernverfahren. Populär und in vielen ML-Wettbewerben erfolgreich ist in den letzten Jahren das Tree Boosting geworden (Chen/Guestrin, 2016), bei dem Entscheidungsbäume (Breiman und andere, 1984) als Base Classifiers eingesetzt werden. Die Grundidee eines Entscheidungsbaums ist, ein Regelwerk zur Entscheidungsfindung in Form eines Binärbaums zur Verfügung zu stellen, der auf einem Trainingsdatensatz basierend erstellt wird. Der Entscheidungsbaum kann anschließend dazu genutzt werden, neue unbekannte Datenpunkte zu klassifizieren. Das Beispiel in ↘ Grafik 1 zeigt einen Datensatz, für dessen Datenpunkte zwei Merkmale, a und b, gegeben sind und die zwei unterschiedlichen Klassen angehören, der positiven Klasse (blau) und der negativen Klasse (rot). Der aus dem Datensatz erzeugte Entscheidungsbaum enthält sogenannte Knoten. An diesen werden bei der Klassifikation eines neuen Datenpunkts

Grafik 1
Entscheidungsbaum



(weiß) bei der Wurzel[1] beginnend dessen Merkmalsausprägungen abgefragt, bis ein Blatt[2] des Baumes erreicht wird, wo schließlich eine der beiden Klassen, blau oder rot, dem Datenpunkt zugewiesen wird.

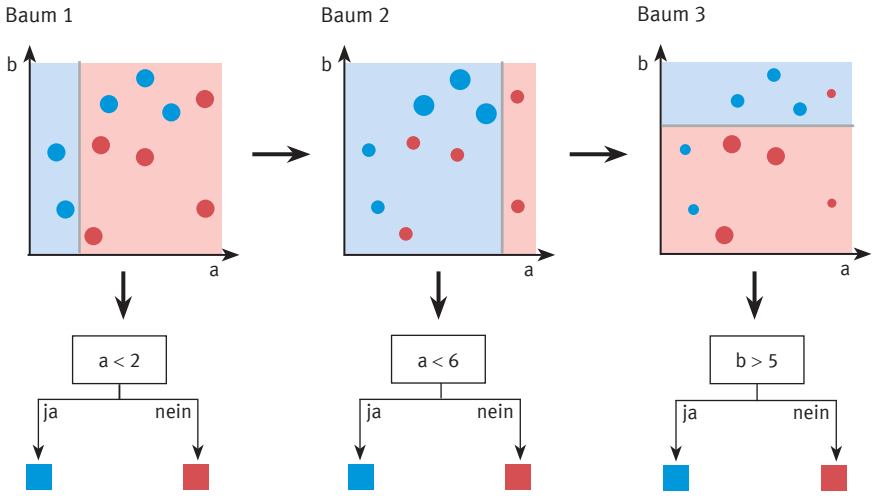
Wie mit Entscheidungsbäumen können mit Tree Boosting Modelle für Klassifikations- und Regressionsprobleme berechnet werden. Im Gegensatz zu Random Forests (Breiman, 2001), die ebenfalls auf Entscheidungsbäumen basieren, werden die einzelnen Klassifikatoren beim Tree Boosting sukzessive trainiert, wobei jeder neu trainierte Klassifikator vom vorherigen abhängt. Dies wird so umgesetzt, dass allen Datenpunkten für das Training des ersten Klassifikators zunächst das gleiche Gewicht, welches für den Algorithmus der aktuellen Wichtigkeit einer korrekten Zuordnung dieses Datenpunktes entspricht, zugeordnet wird. Für das Training der folgenden Klassifikatoren werden die Gewichte jeweils so angepasst, dass die Gewichte der vom vorherigen Klassifikator falsch zugeordneten Datenpunkte erhöht werden. Beim Tree Boosting werden Entscheidungsbäume geringer Höhe[3] eingesetzt. Um die vom Gesamtmodell vorhergesagte Klasse für einen neuen Datenpunkt zu erhalten, wird unter Einbeziehung der Klassifikationsergebnisse aller berechneten Bäume abgestimmt. Die Stimmen der einzelnen Bäume gehen mit einer baumspezifischen Gewichtung in das Ergebnis ein. Die Gewichtung berechnet sich aus ihrem individuellen Fehler auf den Trainingsdaten.

Im vorliegenden Beispiel (übernommen aus Schapire/Freund, 2012) soll nun das Boosting-Modell lernen, die richtige Klasse aus dem vorigen Beispiel anhand der gegebenen Merkmale vorherzusagen. Im ersten Trainingszyklus erhalten alle Datenpunkte des Trainingsdatensatzes das gleiche Gewicht (in ↘ Grafik 2 durch die Größe der jeweiligen Datenpunkte dargestellt). Der erste Entscheidungsbaum wird unter Berücksichtigung dieser Gewichte erstellt und anschließend der Fehler für diesen Baum und darauf basierend sein Gewicht berechnet. Für das Training des zweiten Entscheidungsbaums werden die Gewichte der Datenpunkte, die vom ersten Baum falsch klassifiziert werden, erhöht (blaue Datenpunkte, die bei Baum 1 in den roten Klassifikationsbereich fallen) und die Gewichte der richtig klassifizierten Datenpunkte verringert. Nach demselben Prinzip werden die Gewichte der Datenpunkte für das Training des dritten Entscheidungsbaums basierend auf der Klassifikation des zweiten angepasst (Erhöhung der Gewichte der drei roten Datenpunkte, die in den blauen Klassifikationsbereich von Baum 2 fallen, Verringerung der übrigen Gewichte).

Soll das trainierte Boosting-Modell dazu eingesetzt werden, einen neuen Datenpunkt (weiß in ↘ Grafik 3) mit den Merkmalen a = 5 und b = 2 zu klassifizieren, werden alle trainierten Entscheidungsbäume einzeln ausgewertet, sodass jeder Baum unabhängig eine Klasse für den unbekannten Datenpunkt vorhersagt.

1 Die Wurzel ist der Knoten, in den keine Kante hineinführt.
2 Ein Blatt ist ein Knoten, aus dem keine Kante hinausführt.
3 Die Höhe eines Entscheidungsbaums ist der längste Weg von der Wurzel zu einem Blatt.

Grafik 2
Erstellung eines Baum-Ensembles beim Boosting



Fehler	0,3	0,21	0,14
Gewicht	0,42	0,65	0,92

2020 - 0559

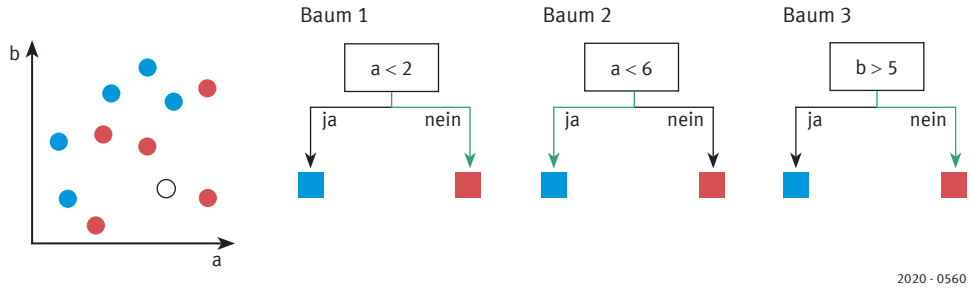
Unter Berücksichtigung der jeweiligen Baumgewichte werden die Stimmen der drei Entscheidungsbäume für die positive (blau) und negative (rot) Klasse verrechnet und so die Klasse des neuen Datenpunkts bestimmt:

$$-0{,}42 + 0{,}65 - 0{,}92 = -0{,}69$$

Der neue Datenpunkt wird somit der negativen Klasse (rot) zugeordnet.

Für die Berechnungen der für das vorliegende Projekt erzeugten Modelle wurde das in der Programmiersprache R verfügbare XGBoost-Paket (Chen/Guestrin, 2016) verwendet. Dieses bietet die Möglichkeit, über eine Vielzahl von Parametern Einfluss auf die Modellberechnung zu nehmen (Readthedocs Webseite von XGBoost, 2020). Um das beste Modell für den verwendeten Datensatz zu erhalten, wurde ein Teil dieser Parameter über eine Gittersuche variiert und so die optimalen Parameterwerte gesucht. Empfehlungen für zu variierende Parameter und deren Wertebereiche sind verschiedenen Quellen im Internet zu entnehmen (beispielsweise Hackerearth, 2020; Jain, 2016).

Grafik 3
Klassifizierung eines neuen Datenpunkts beim Boosting



2020 - 0560

3.2 Bewertung

Für die für den Vergleich notwendige Bewertung von ML-Modellen zur Klassifikation gibt es zahlreiche Möglichkeiten. Die Sensitivity gibt den Anteil der richtig als positiv klassifizierten Datenpunkte an allen tatsächlich positiven an, die Precision den Anteil der richtig als positiv klassifizierten an allen, die als positiv vorhergesagt wurden. Die Specificity gibt den Anteil der richtig als negativ klassifizierten an allen tatsächlich negativen an. Das Standardmaß ist die Accuracy (acc), die dem Anteil der richtig zugeordneten Datenpunkte an allen zugeordneten Datenpunkten entspricht. Gerade im Fall von starken Größenunterschieden der Klassen im Trainingsmaterial können aber bei der Accuracy unerwünschte Effekte auftreten. Geht man wie im vorliegenden Fall beispielsweise von einem 95 %-Anteil der Mehrheitsklasse im Gegensatz zu einem 5 %-Anteil der Minderheitsklasse aus, so würde ein Klassifikator, der grundsätzlich immer der Mehrheitsklasse zugeordnet wird, also völlig wertlos ist, eine Accuracy von 0,95 erreichen und damit eine gute Klassifikationsgüte suggerieren. Daher sollte zur Accuracy die „No Information Rate“ hinzugezogen werden, die angibt, welche Accuracy erreicht wird, wenn immer die Mehrheitsklasse vorhergesagt wird. Die Accuracy des Modells sollte diesen Wert übersteigen.

Grafik 4
Metriken zur Modellbewertung

		Tatsächliche Klasse	
		Positiv (P)	Negativ (N)
Vorhergesagte Klasse	Positiv	Richtig positiv (TP)	Falsch positiv (FP)
	Negativ	Falsch negativ (FN)	Richtig negativ (TN)

$$acc = \frac{TP + TN}{P + N}$$

$$sensitivity = \frac{TP}{TP + FN} = \frac{TP}{P}$$

$$specificity = \frac{TN}{TN + FP} = \frac{TN}{N}$$

$$precision = \frac{TP}{TP + FP}$$

$$bacc = \frac{sensitivity + specificity}{2} = \left(\frac{TP}{P} + \frac{TN}{N}\right) / 2$$

$$F1score = 2 \cdot \frac{sensitivity \cdot precision}{sensitivity + precision} = \frac{2TP}{2TP + FP + FN}$$

2020 - 0561

Andere Maße wie die Balanced Accuracy (bacc) oder der F1score bewerten ebenfalls in einem Wertebereich zwischen 0 und 1 das beste Modell mit 1. Sie gewichten die „Falsch Negativen“ nicht in dieser Form und liefern für das genannte Beispiel die Werte 0,5 beziehungsweise 0. Sie werden daher oft im Fall eines Imbalanced Data Szenarios eingesetzt. ↘ Grafik 4

4 Vorgehen

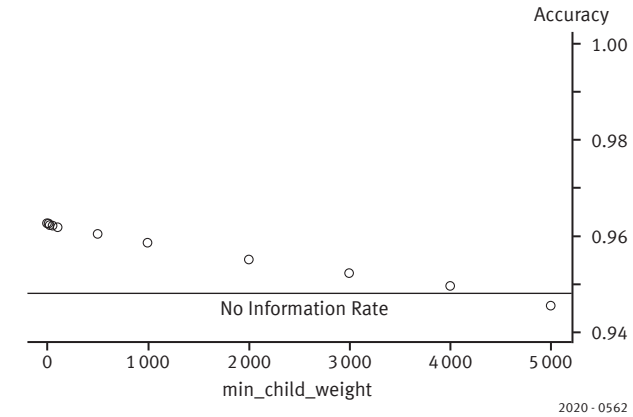
4.1 Geheimhaltungsaspekt

Auch beim Einsatz von Verfahren, die auf Entscheidungsbäumen basieren, ist es theoretisch möglich, anhand des berechneten Modells Rückschlüsse auf Einzeldaten zu ziehen, sofern ein Baum nur einen einzelnen Datenpunkt in einem Blatt enthält. Um das zu verhindern, ist sicherzustellen, dass sich eine Mindestzahl von Datenpunkten in jedem Blatt des Entscheidungsbaums befindet. Dies kann über einen Parameter beim Training des Modells gesteuert werden. In Absprache mit der für die Verdienststrukturerhebung zuständigen Fachabteilung wurden für die vorliegenden Rechnungen 25 Datenpunkte je Blatt als Minimum gewählt.

Dieser Parameter wird normalerweise für die Regularisierung[4] des Modells eingesetzt. Um seinen Einfluss auf die Accuracy der Ergebnisse einschätzen zu können, wurde eine Testreihe erstellt, bei der der betreffende Parameter zwischen 1 und 5 000 variiert wurde. Die Ergebnisse zeigen, dass die Verschlechterung der Accuracy im Bereich zwischen 1 und 25 zu vernachlässigen ist. Erst bei einem Parameterwert von über 4 000 erreicht die Accuracy den Wert der hier vorliegenden „No Information Rate“, bei dem ein Modell keinen Mehrwert bietet im Vergleich zu einem Schätzer, der immer die Mehrheitsklasse vorhersagt. ↘ Grafik 5

4 Dieser Begriff bezeichnet die Verringerung der Komplexität eines Modells, um eine Überanpassung an die Daten zu verhindern.

Grafik 5
Einfluss des XGBoost-Parameters min_child_weight auf die Accuracy des Modells



4.2 Rückschluss von der Stichprobe auf den Gesamtdatensatz

Wie bereits erwähnt, handelt es sich bei der Verdienststrukturerhebung im Gegensatz zu den Daten der Bundesagentur für Arbeit um eine geschichtete Stichprobe. Dieser Stichprobe entsprechend sind den Datenpunkten Hochrechnungsfaktoren zugeordnet, mit denen auf die Gesamtdaten geschlossen werden kann, wie sie auch von der Bundesagentur für Arbeit erhoben werden. Da das entwickelte ML-Modell auf die BA-Daten angewendet werden soll, ist es sinnvoll, die Hochrechnungsfaktoren bei der Modellberechnung mit einzubeziehen. Hierfür boten sich zwei Möglichkeiten: Der erste Fall berücksichtigt die Hochrechnungsfaktoren bei der Berechnung der verwendeten Gütemaße, indem er die einzelnen Datensätze mit den Faktoren gewichtet. Dies erfolgt, um beim Testen eine Verteilung mit den VSE-Daten abzubilden, die möglichst ähnlich zu der von den bei der Bundesagentur für Arbeit vorliegenden Daten ist. So vorgegangen wurde sowohl bei der Kreuzvalidierung im Rahmen des Parameter-Tunings wie auch bei der finalen Bewertung des Modells mithilfe des Testdatensatzes. Eine zweite Möglichkeit bietet der verwendete Gradient-Boosting-Algorithmus XGBoost, indem er die Einbeziehung von datenpunktspezifischen Gewichten bei der Modellerstellung ermöglicht. Diese Variante wurde getestet, hatte aber keinen statistisch signifikanten Einfluss auf die Ergebnisse und wurde für die Berechnung der finalen Modelle nicht eingesetzt.

4.3 Imbalanced-Data-Problem

Es gibt verschiedene Möglichkeiten, dem Effekt des Imbalanced-Data-Problems entgegenzuwirken. Für das vorliegende Projekt wurden drei Methoden getestet. Die erste Methode ist die Variation verschiedener Parameter, die die Behandlung der verschiedenen Klassen bei der Modellberechnung mit XGBoost beeinflussen. Die zweite Methode ist das sogenannte Downsampling der entsprechenden Mehrheitsklasse, bei dem das Verhältnis der Klassen im Trainingsdatensatz zueinander durch Reduktion der Datenpunkte der Mehrheitsklasse angeglichen wird. Die dritte Methode ist die Wahl einer entsprechenden Metrik für die Modellbewertung.

Die Variation der beiden von XGBoost bereitgestellten Parameter „scale_pos_weight“ und „max_delta_step“ (Readthedocs Webseite von XGBoost, 2020) brachte keine Verbesserung der Ergebnisse im Vergleich zur Standardrechnung. Daher wurden für die finale Modellberechnung die Standardwerte dieser Parameter verwendet. Um den Einfluss von Downsampling zu untersuchen, wurde ein neuer Trainingsdatensatz erstellt, der sich aus den Teilzeitbeschäftigten und 50 % der Vollzeitbeschäftigten zusammensetzte. Das resultierende Modell wurde mit einem entsprechenden Modell, das auf dem Gesamttrainingsdatensatz trainiert wurde, verglichen.

Für die Modellbewertung wurden im Zuge der vorliegenden Testrechnungen die Metriken Balanced Accuracy, F1score und auch die Accuracy bestimmt. Um die Auswirkungen des Downsampling zu illustrieren und die berechneten Metriken zu vergleichen, sind im Folgenden die Werte für beide Modelle, mit und ohne Downsampling, dargestellt. ↘ Tabelle 1

Tabelle 1
Bewertung der ML-Modelle mit und ohne Downsampling

Metrik	Standard	Mit Downsampling
Sensitivity	0,5203	0,6060
Precision	0,8219	0,7209
Accuracy	0,9650	0,9629
Balanced Accuracy	0,7566	0,7956
F1score	0,6373	0,6585

Gemessen an Balanced Accuracy und F1score führt Downsampling im Vergleich zum Standardvorgehen zu besseren Ergebnissen. Das ist darauf zurückzuführen, dass Downsampling auf eine Steigerung der Sensitivität des Modells abzielt, was heißt, dass möglichst viele Teilzeitbeschäftigte unter den als Vollzeitbeschäftigte deklarierten Beschäftigten identifiziert werden. Wenn der Anteil der Teilzeitbeschäftigten im Trainingsmaterial also höher ist, klassifiziert das Modell einen Beschäftigten auch eher als Teilzeitbeschäftigten. Anhand von Sensitivity und Precision ist zu erkennen, dass im Fall des Downsamplings zwar wesentlich mehr Teilzeitbeschäftigte als solche erkannt werden, aber zugleich auch viel mehr tatsächlich Vollzeitbeschäftigte als teilzeitbeschäftigt klassifiziert werden, also die Falsch-Positiv-Rate steigt.

Dieser Effekt ist bei der Bewertung zu berücksichtigen, da bei der vorliegenden Fragestellung nicht im Vordergrund stand, möglichst viele Teilzeitbeschäftigte zu identifizieren, sondern die Anzahl der fehlerhaften Tätigkeitsschlüssel zu minimieren. Anders formuliert bestand das Ziel darin, möglichst viele Beschäftigte, ob Teilzeit- oder Vollzeitbeschäftigte, richtig zu klassifizieren, was gleichbedeutend mit der Maximierung der Accuracy ist. Folglich wurde die Accuracy für die Bewertung der berechneten Modelle eingesetzt. Für die Modellrechnungen, bei denen die Auswirkung des Einsatzes von Downsampling getestet wurde, ergibt sich somit, dass die Klassifikationsgüte des mit Downsampling berechneten Modells minimal schlechter ist als die des Modells, das basierend auf dem Gesamttrainingsdatensatz berechnet wurde. Folglich wurde Downsampling bei der Berechnung des finalen Modells nicht eingesetzt.

5 Ergebnisse

Das erstellte ML-Modell soll dazu eingesetzt werden, den fehlerhaften IEB-Datensatz zu korrigieren. Eine solche Korrektur ist dann erfolgreich, wenn der Fehler des Modells niedriger ist als der ursprüngliche Fehler im Datensatz oder – umgekehrt formuliert – wenn die Accuracy des Modells höher ist als die der Ausgangsdaten. In diesem Fall entspricht dies auch der „No Information Rate“. Die folgende Auswertung bezieht sich auf den vom Modelltraining ausgeschlossenen Testdatensatz mit 97 324 Beschäftigten, der nach Berücksichtigung der Hochrechnungsfaktoren 3 963 994 Personen entspricht. Die Accuracy der Daten liegt vor der Korrektur mit dem ML-Modell bei 0,9402; das ist der Anteil der richtig als Vollzeitbeschäftigte geführten Beschäftigten. Mit dem berechneten XGBoost-Modell kann eine Accuracy von 0,965 erzielt werden, was einer Verbesserung um 0,0248 Prozentpunkte entspricht. In absoluten Zahlen unter Berücksichtigung der Hochrechnungsfaktoren entsteht diese Verbesserung dadurch, dass bei 234 254 fälschlich als Vollzeitbeschäftigte geführten Teilzeitbeschäftigten 121 246 richtig als Teilzeitbeschäftigte klassifiziert werden, aber auch 25 753 Vollzeitbeschäftigte falsch als Teilzeitbeschäftigte klassifiziert werden. Dadurch verringert sich insgesamt die Zahl der falsch signierten Beschäftigten um 95 493 (siehe auch Tabelle 2).

Ausschluss kritischer Subgruppen

Als alternativer Ansatz wurde untersucht, ob es möglich ist, durch den Ausschluss von Teilgruppen, für die eine Vorhersage des Tätigkeitsschlüssels nur schwer möglich ist, das Ergebnis für die verbleibenden Datensätze zu verbessern. Hierzu erfolgten verschiedene Modellberechnungen, bei denen Beschäftigte beispielsweise abhängig von ihrem Bruttomonatsverdienst oder der Anzahl der Beschäftigten im Betrieb mit unterschiedlichen Schwellenwerten von der Berechnung ausgeschlossen wurden.

Um systematisch Gruppen zu identifizieren, für die die Klassifikationsgüte vergleichsweise schlecht ist, wurde letztlich ein Entscheidungsbaum eingesetzt, der auf dem Trainingsdatensatz mit dem Fehler des XGBoost-Modells als Zielvariable trainiert wurde. Die Idee war, mithilfe des Baums zu bestimmen, welche bekannten Merkmale innerhalb welcher Wertebereiche möglicherweise einen Hinweis darauf geben, in welchen Fällen der Vollzeit/Teilzeit-Schlüssel zuverlässig vorhergesagt werden kann und für welche Fälle dies nicht möglich ist, sodass letztere von der Berechnung ausgeschlossen werden können. ↘ Tabelle 2

Es wurden verschiedene Varianten getestet, die sich durch die Anzahl der berücksichtigten Knoten des Entscheidungsbaums unterscheiden. Die getesteten Vari-

anten führen für die Teilgruppe im Vergleich zur Gesamtgruppe zu einer deutlichen Erhöhung der Sensitivität bei fast gleichbleibender Precision. Ein Beispiel, bei dem nur ein Blatt des Baums, das einer Auswahl von Beschäftigten mit einem maximalen Bruttomonatsverdienst von 1 899 Euro entspricht, berücksichtigt wurde, wird im Folgenden diskutiert.

Tabelle 2
Bewertung der ML-Modelle

Gütemaß	Alle korrigiert	Nur Bruttomonatsverdienst unter 1 900 EUR korrigiert	
	Standard	Auswertung für Subgruppe	Auswertung für alle
Accuracy	0,9650	0,9127	0,9635
Accuracy vor der Korrektur	0,9402	0,8191	0,9402
Sensitivity	0,5176	0,6640	0,4915
Precision	0,8248	0,8192	0,8192
Echte Teilzeitbeschäftigte	234 254	173 395	234 254
Anzahl korrigierter Teilzeitbeschäftigter	95 493	89 724	89 724

Für die betrachtete Teilmenge, die sich durch den Ausschluss von Beschäftigten mit einem Bruttomonatsverdienst von mehr als 1 899 Euro ergibt, ist die Accuracy vor der Korrektur mit 0,8191 wesentlich geringer als für die Gesamtmenge (0,9402). Durch die Korrektur des ML-Modells wird die Accuracy für diese Teilmenge um etwa neun Prozentpunkte auf 0,9127 erhöht. Berechnet man analog zur Darstellung für die Gesamtmenge mithilfe der Hochrechnungsfaktoren die entsprechenden absoluten Zahlen, so würden bezogen auf die Testmenge die insgesamt 234 254 falsch klassifizierten Beschäftigten mit der Korrektur des ML-Verfahrens um 89 724 verringert. Für die Teilmenge ist das Potenzial zur Verbesserung also deutlich höher und es wird auch ein größerer Anteil von Beschäftigten korrigiert, die absolute Anzahl an korrigierten Datenpunkten bezogen auf den Gesamtdatensatz ist jedoch geringer.

Daher stellte sich die Frage, ob das auf der Teilmenge trainierte Modell besser an diese angepasst ist oder ob die beiden Modelle auf der Teilmenge ein vergleichbares Ergebnis liefern. Es zeigte sich, dass sich die Modelle selbst kaum unterscheiden. Die unterschiedliche Klassifikationsgüte, die auf der Teilmenge und der Gesamtmenge beobachtet wurde, lässt sich hauptsächlich durch die Unterschiede zwischen den Mengen, also die Merkmale der entsprechenden Beschäftigten, begründen.

Gemessen an der Accuracy bezogen auf den Gesamtdatensatz ist es möglich, ohne eine Subgruppenauswahl mit 0,965 im Vergleich zu 0,9635 (mit Ausschluss von Beschäftigen mit einem Bruttomonatsverdienst von mindestens 1 900 Euro) ein etwas besseres Ergebnis zu erzielen. In diesem Fall kann die Anzahl der falsch klassifizierten Teilzeitbeschäftigten insgesamt etwas weiter gesenkt werden. Es werden aber auch bewusst Korrekturen für eine Menge von Beschäftigten vorgenommen, für die das Modell nur sehr unzuverlässige Vorhersagen macht (Bruttomonatsverdienst von mindestens 1 900 Euro). Es kann nicht getestet werden, inwieweit sich die Güte der Vorhersagen für den VSE-Datensatz auf die Vorhersagen mit den Integrierten Erwerbsbiografien übertragen lässt, insbesondere da die Qualität anderer gemeinsamer Merkmale, auf denen die Vorhersage beruht, durch unterschiedliche Bearbeitung der beiden Datensätze leicht voneinander abweichen können. Daher besteht bei einer Subgruppe, für die die Vorhersage bei unserem Datenmaterial eher unzuverlässig ist, auch wenn sie den Fehleranteil minimal senkt, die Gefahr, dass bei Anwendung auf die Integrierten Erwerbsbiografien der Fehleranteil gleich bleibt oder sich sogar erhöht. Im letzteren Fall ist natürlich von einer Korrektur innerhalb dieser Subgruppe abzusehen, aber auch bei gleichbleibendem Fehleranteil ist die Originalkodierung vorzuziehen.

Zudem warf die Fehlsignierung der Teilzeitkräfte insbesondere Probleme bei Analysen des Niedriglohnbereichs auf. Im Jahr 2014 lag laut Verdienststrukturerhebung die Niedriglohnschwelle bei einem Bruttomonatsverdienst von 1 993 Euro. Somit liegt die Menge der Beschäftigten, für die nur eine eher unzuverlässige Vorhersage möglich ist, zu einem großen Teil außerhalb des für die Analyse der Daten besonders interessanten Bereichs. Damit könnten mit einer Einschränkung der zu korrigierenden Beschäftigten die genannten Probleme ohne starken Einfluss auf die Analyseergebnisse umgangen werden. Darüber hinaus ist anzumerken, dass zwar die von der Korrektur ausgeschlossene Menge 75 % der Beschäftigten ausmacht, aber nur 27 % der falsch klassifizierten Teilzeitbeschäftigen dazugehören. Für diese würde in der Folge des Ausschlusses keine Korrektur durch das Modell vorgenommen.

Zusammenfassend kann für die Anwendung auf die Integrierten Erwerbsbiografien festgehalten werden, dass das auf dem Gesamtdatensatz trainierte Modell eingesetzt werden sollte.

Sollen aber Korrekturen, die auf eher unzuverlässigen Vorhersagen beruhen, nicht vorgenommen werden, ist es trotzdem möglich, entsprechende Teilmengen des Datensatzes direkt bei der Korrektur auszuschließen.

6

Fazit

Mit dem vorgestellten Projekt sollte ein Korrekturverfahren für das im Tätigkeitsschlüssel der Integrierten Erwerbsbiografien kodierte Merkmal „Vollzeit/Teilzeit“ entwickelt werden. Um eine hohe Datenqualität zu gewährleisten, findet für die Erstellung des VSE-Datensatzes des Statistischen Bundesamtes mit der Plausibilisierung ein aufwendiges, manuelles Korrekturverfahren statt. Im Zuge dessen wurde das auch in der Verdienststrukturerhebung vorhandene Merkmal „Vollzeit/Teilzeit“ für die Erhebung von 2014 überprüft und korrigiert. Zur Korrektur wurden auch Angaben zu den bezahlten Arbeitsstunden verwendet, die in der Verdienststrukturerhebung im Gegensatz zu den Integrierten Erwerbsbiografien vorhanden sind. Für das vorliegende Problem war dies nutzbar, indem ein maschinelles Lernverfahren auf den manuell korrigierten VSE-Daten zur Vorhersage der korrekten Angabe über die Vollzeit- oder Teilzeitbeschäftigung trainiert wurde. Unter Berücksichtigung verschiedener projektspezifischer Anforderungen wurde so ein ML-Modell erstellt, das in der Lage ist, den Fehler in der Signierung des Tätigkeitsschlüssels bei einer aus dem VSE-Datensatz erzeugten Testmenge zu reduzieren, bei 234 254 falsch kodierten Einträgen um 95 493. Mit dem entwickelten Verfahren besteht somit auf die Integrierten Erwerbsbiografien angewendet die Möglichkeit, die dort festgestellte Untererfassung der Teilzeitbeschäftigten teilweise zu korrigieren.

LITERATURVERZEICHNIS

Bertat, Thomas/Dundler, Agnes/Grimm, Christopher/Kiewitt, Jochen/Schomaker, Christine/Schridde, Henning/Zemann, Christian. *Neue Erhebungsinhalte "Arbeitszeit", "ausgeübte Tätigkeit" sowie "Schul- und Berufsabschluss" in der Beschäftigungsstatistik*. Methodenbericht der Statistik der BA. 2013. [Zugriff am 12. November 2020]. Verfügbar unter: statistik.arbeitsagentur.de

Breiman, Leo. *Arcing classifier (with discussion and a rejoinder by the author)*. In: Annals of Statistics. Jahrgang 26. Ausgabe 3/1998, Seite 801 ff.

Breiman, Leo/Friedman, Jerome H./Olshen, Richard A./Stone, Charles J. *CART: Classification and Regression Trees*. Boca Raton 1984.

Breiman, Leo. *Random Forests*. In: Machine Learning. Ausgabe 45/2001, Seite 5 ff.

Chen, Tianqi/Guestrin, Carlos. *XGBoost: A Scalable Tree Boosting System*. In: arXiv, 2016, arXiv:1603.02754

Fitzenberger, Bernd/Seidlitz, Arnim. *The 2011 Break in the Part-Time Indicator and the Evolution of Wage Inequality in Germany*. ZEW Discussion Paper No. 19-029, Mannheim 2019. [Zugriff am 12. November 2020]. Verfügbar unter: ftp.zew.de

Fitzenberger, Bernd/Seidlitz, Arnim. *Die Lohnungleichheit von Vollzeitbeschäftigten in Deutschland: Rückblick und Überblick*. In: AStA Wirtschafts- und Sozialstatistisches Archiv. Jahrgang 14. Ausgabe 2/2020, Seite 125 ff. [Zugriff am 12. November 2020]. Verfügbar unter: link.springer.com

Freund, Yoav/Schapire, Robert E. *Experiments with a new boosting algorithm*. In: Machine Learning: Proceedings of the Thirteenth International Conference. 1996. [Zugriff am 12. November 2020]. Verfügbar unter: cseweb.ucsd.edu

HackerEarth. *Beginners Tutorial on XGBoost and Parameter Tuning in R*. [Zugriff am 12. November 2020]. Verfügbar unter: www.hackerearth.com

Jain, Aarshay. *Complete Guide to Parameter Tuning in XGBoost with codes in Python*. 2016. [Zugriff am 12. November 2020]. Verfügbar unter: www.analyticsvidhya.com

Möller, Joachim. *Lohnungleichheit – Gibt es eine Trendwende?* In: Wirtschaftsdienst. Jahrgang 96. Ausgabe 1/2016, Seite 38 ff. [Zugriff am 12. November 2020]. Verfügbar unter: www.wirtschaftsdienst.eu

Readthedocs Webseite von XGBoost. *XGBoost Parameters.* [Zugriff am 12. November 2020]. Verfügbar unter: xgboost.readthedocs.iol

Schapire, Robert E./Freund, Yoav. *Boosting: Foundations and Algorithms*. 2012.

Schapire, Robert E. *The Strength of Weak Learnability*. In: Machine Learning. Jahrgang 5. Ausgabe 2/1990, Seite 197 ff. [Zugriff am 12. November 2020]. Verfügbar unter: rob.schapire.net

Statistisches Bundesamt. *Qualitätsbericht Verdienststrukturerhebung 2014*. Wiesbaden 2016. Verfügbar unter: www.destatis.de

Herausgeber
Statistisches Bundesamt (Destatis), Wiesbaden

Schriftleitung
Dr. Daniel Vorgrimler
Redaktionsleitung: Juliane Gude
Redaktion: Ellen Römer

Ihr Kontakt zu uns
www.destatis.de/kontakt

Erscheinungsfolge
zweimonatlich, erschienen im Dezember 2020
Das Archiv älterer Ausgaben finden Sie unter www.destatis.de

Artikelnummer: 1010200-20006-4, ISSN 1619-2907