FH Vorarlberg
University of Applied Sciences

Datenanalyse mittels Python

Gross national income per capita Female

Projektbericht
Data Science

Fachhochschule Vorarlberg
Studiengang

Betreut von
Dr. Klaus Rheinberger

Vorgelegt von
Kilian Schatzmann

Dornbirn, 25.06.2017

Inhaltsverzeichnis

Darstellungsverzeichnis

1. Aufgabenstellung

Im Rahmen der Lehrveranstaltung Data Science des Studiengangs Energietechnik und Energiewirtschaft wurden wir mit einer Projektarbeit zur Datenanalyse mittels Python beauftragt. Der Projektauftrag setzt sich im Wesentlichen mit folgenden Schwerpunkten auseinander:

- selbständige Beschaffung von Daten sowie Aufarbeitung der Datensätze
- grobe Datenanalyse (bspw. sind ausreichend Daten vorhanden, korrelieren die Daten, Bestimmung der Zielgröße,…)
- detaillierte Datenanalyse und Datenoptimierung
- Programmiertool: *Python*
- Methodik: angelehnt an die Lehrveranstaltung sowie das Buch: *Introduction to Machine Learning with Python* (Müller Andreas; Guido Sarah 2017)

2. Herangehensweise an die Projektarbeit

Im Zuge der Projektarbeit habe ich mich für folgende Herangehensweise zur geordneten Abarbeitung der Arbeitspakte des Programmierprojekts entschieden:

1. Datenrecherche
2. Aufarbeitung der Daten
3. Feature Recherche
4. Optische Bewertung mit Hilfe von Plots
5. Datensätze bewerten mit Hilfe verschiedener Regressionsverfahren
6. Auswahl der Regressionsverfahren
7. Datensätze mittels verschiedenster Tools optimieren
8. Code auf Fehler und daraus falsch abgeleitete Schlussfolgerungen überprüfen
9. Bewertung der Verfahren mit eventuellen Handlungsempfehlungen

Um Komplexität zu vermeiden und zielgerichtet möglichst viele Tools von den Unterrichtsvorlesungen verwenden zu können, habe ich mich nach der ersten Datenanalyse für ein bestimmtes Regressionsmodell entschieden. Anhand dieses Modells wurden Optimierungen angestrebt. Die Herangehensweise könnte in gleicher Weise für andere Regressionsmodelle durchgeführt werden.

3. Datenrecherche

Anfänglich wollte ich Datensätze aus meinem beruflichen Umfeld verwenden, da es mir nicht möglich war diese Datensätze in einem zeitlich sinnvollen Rahmen aufzuarbeiten, habe ich weitere Datenrecherche betrieben. Im Zuge dessen bin auf das Data Science Webportal *kaggle („kaggle - Your Home for Data Science“ 2017*) aufmerksam geworden. *Kaggle* stellt eine große Anzahl von Datensätzen zu den unterschiedlichsten Themenbereichen öffentlich zur Verfügung. Ich habe meinen Datensatz nicht von *kaggle,* aber es hat mit sehr geholfen die Vielzahl von Projektmöglichkeiten zu erkennen. Da ich mich im Zuge des Energieprojekts *„Power for Benin“* mit *Human Development Index* Daten

(HDI) aus Afrika beschäftigt habe, habe ich beschlossen, landesspezifische Entwicklungsdaten für das Programmierprojekt zu verwenden.

3.1 HDI

Seit 1990 wird vom Entwicklungsprogramm der Vereinten Nation (UNDP) ein jährlicher Bericht über die menschliche Entwicklung veröffentlicht. Der Bericht umfasst länderspezifische Daten zur menschlichen Entwicklung und errechnet damit die Kennzahl: HDI. Die Kennzahl beinhaltet beispielsweise das Bildungsniveau, die Lebenserwartung und das Pro-Kopf-Einkommen, aus denen in weiterer Folge eine Rangliste erstellt wird. 2015 wurden insgesamt 188 Länder erfasst, von denen 83 in die Kategorie *geringe Entwicklung* eingestuft wurden („Das Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung (BMZ)“ 2010).

3.2 Human Development Data

Die Organisation *Entwicklung der vereinten Nationen* erhebt zu den HDI Daten eine Reihe weiterer Datensätze, welche sich mit Entwicklungsthemen auseinandersetzen. Beispiele dafür wären Arbeitslosigkeit, Armutskennzahlen, demografischer Wandel und vieles mehr. Diese Daten werden kostenlos zur Verfügung gestellt. Es handelt sich um einen bereits aufbereiteten Datensatz. Zudem werden auch alle Inhalte sehr ausführlich erklärt. Der Link zu den Datensätzen ist im Literaturverzeichnis vermerkt („United Nations Development Programme Human Development Reports“ 2017).

3.3 Verwendete Datensätze

Aus dem sehr umfangreichen Datensatz wurden jene Länder, welche unvollständige Datensätze aufweisen entfernt. Daraus resultiert ein Gesamtdatensatz von 134 Ländern mit 26 Features. Die unten angeführte Liste zeigt die verwendeten Features:

- *Human Development Index (HDI)*
- *Gross national income per capita Male*
- *Life expectancy at birth*
- *Expected years of schooling*
- *Mean years of schooling*
- *Gross national income (GNI) per capita*
- *GNI per capita rank minus HDI rank*
- *Average annual HDI growth 2010-2015*
- *Inequality in life expectancy*
- *Inequality-adjusted life expectancy index*
- *Gender Development Index*
- *Human Development Index (HDI) Female*
- *Human Development Index (HDI) Male*
- *Life expectancy at birth Female*
- *Life expectancy at birth Male*
- *Expected years of schooling Female*

- *Expected years of schooling Male*
- *Mean years of schooling Female*
- *Mean years of schooling Male*
- *Employment to population ratio (% ages 15 and older)*
- *Labour force participation rate (% ages 15 and older)*
- *Unemployment Total (% of labour force)*
- *Unemployment Youth (% ages 15-24)*
- *GDP Total (2011 PPP $ billions)*
- *GDP Per capita (2011 PPP $)*
- *Total unemployment rate*
- *Gross national income per capita Female*

Ich habe ausschließlich Daten verwendet, welche von der UNDP veröffentlicht wurden. Die Datensätze wurden von mir weder inhaltlich geändert noch auf Korrektheit geprüft.

3.4 Die Zielgröße

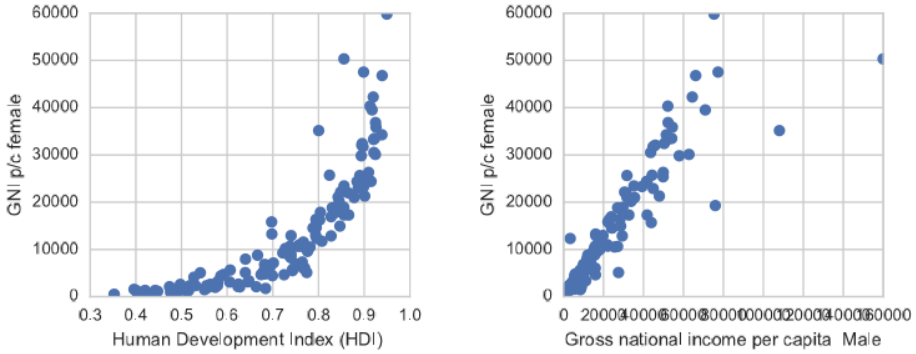
Im ersten Anlauf habe ich versucht die Arbeitslosigkeit (*Total unemployment rate)* über die gesammelten Informationen vorauszusagen. Leider hatten sich keine brauchbaren Korrelationen der Zielgröße: *Arbeitslosigkeit* mit den in Abschnitt 2.3 aufgezählten Features gezeigt. Erste Regressionsanalysen haben bestätigt, dass es nicht möglich ist, ohne zusätzliches Datenmaterial (bzw. besser korrelierende Features) die Arbeitslosigkeitszahlen vorauszusagen. Nach Abstimmung mit Klaus Rheinberger habe ich mich letztlich für eine besser korrelierende Zielgröße entschieden: *Gross national income per capita Female.*

4. Zum Programmierprojekt

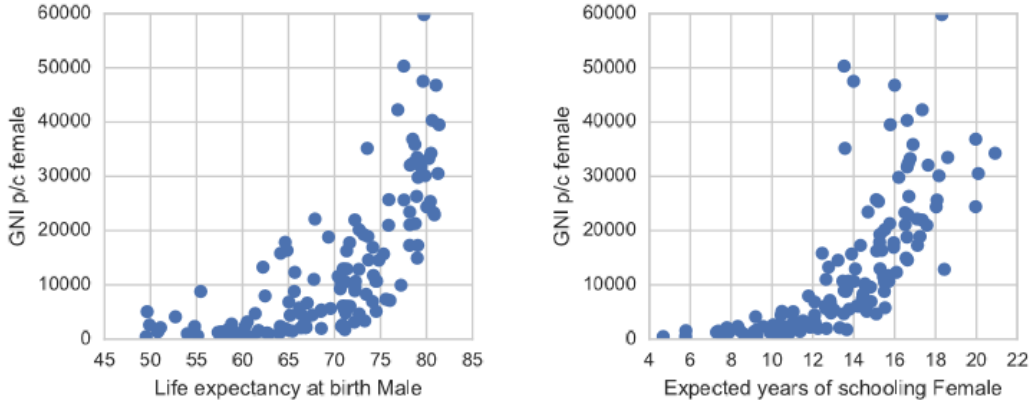
Ich habe mich dazu entschlossen die *Random State Funktion* nicht zu verwenden, damit bei jedem Split neu verteilt wird. Somit wird sich bei erneuter Simulation auch ein etwas anderes Ergebnis einstellen. Das File wurde mehrfach getestet und Abweichungen der Ergebnisdaten bleiben sehr gering.

4.1 Korrelationen von Features und Targets

Darstellung 1 und Darstellung 2 zeigen Ausschnitte des Feature-Target-Scatterplots. Wie zu erwarten korreliert das pro Kopf Bruttoeinkommen männlich (*Gross national income per capita Male*) sowie Bildungszeit in Jahre (*Expected years of schooling Female*) sehr gut mit der Zielgröße: Bruttoinlandsprodukt pro Kopf von Frauen (*Gross national income per capita Female).*



Darstellung 1: Target-Feature-Plot: *Human Development Index (HDI)* und Kopf *Gross national Income per capita Male*



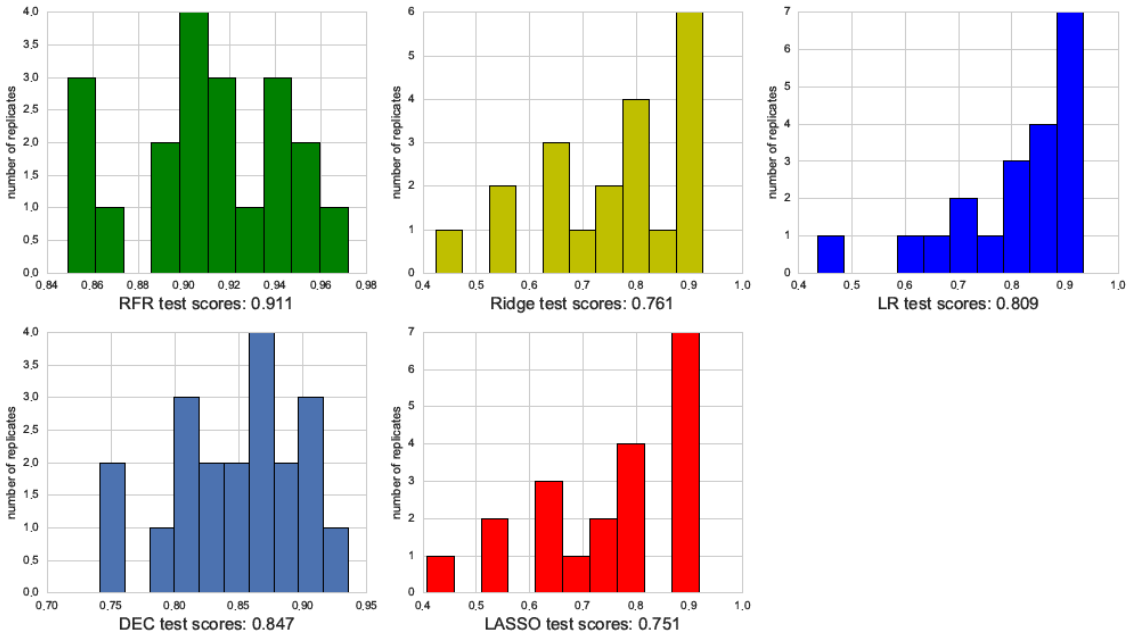
Darstellung 2: Target-Feature-Plot: Life expectancy at birth Male und Expected years of schooling Female

4.2 Status Quo

Um bessere Aussagekraft über die Korrelation der Feature Daten in Kontext auf die Target Daten zu erhalten, wurden folgende Regressionsvarianten angewendet:

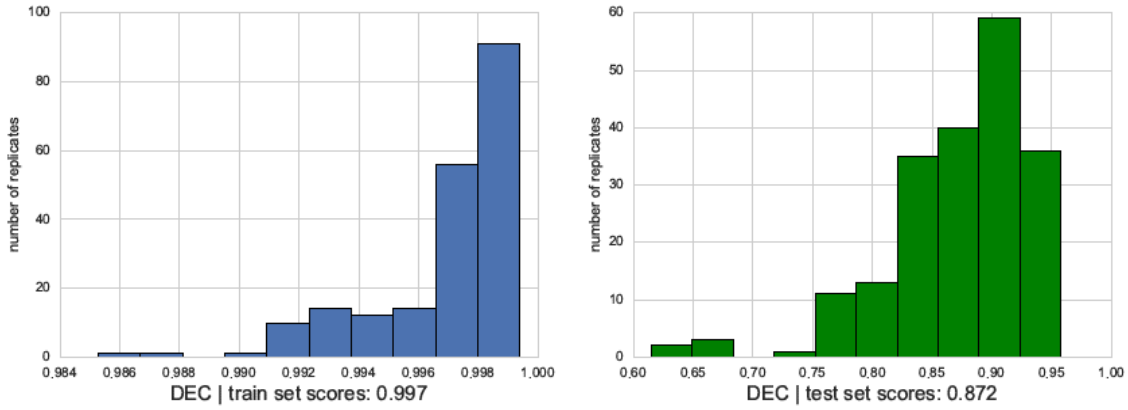
- *Random Forest Regression (RFR)*
- *Ridge Regression (Ridge)*
- *Linear Regression (LR)*
- *Decision Tree Regression (DEC)*
- *Lasso*

Dies wurde mittels Schleife zwanzigfach wiederholt und das Mittel daraus gebildet. Darstellung 3 zeigt die daraus resultierenden Ergebnisse.



Darstellung 3: Verifizieren der Fits (Mittelwert von 20 Fits)

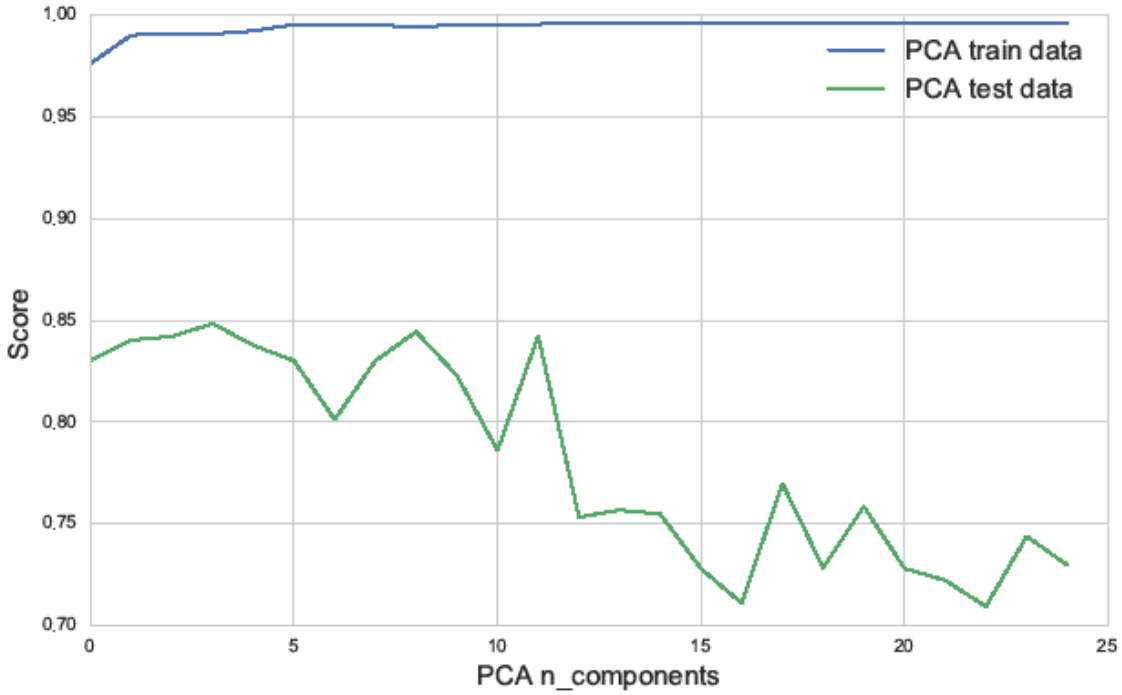
Wie bereits erwähnt, können Korrelationen verschiedener Daten schon vom Scatterplot erkannt werden. Die gemittelten Regressionsintervalle bestätigen diese Aussage, denn die Testscores der Vorhersage liegen bei Varianten über 80%. Für die weitere Bearbeitung habe ich mich entschlossen, die Regressionsvariante des *Decision Tree Regressors* zu optimieren. Darstellung 4 verdeutlicht nochmals die zu optimierende Ausgangslage anhand des Mittels aus 200 Simulationen.



Darstellung 4: Monte Carlo Simulation n=200, DEC

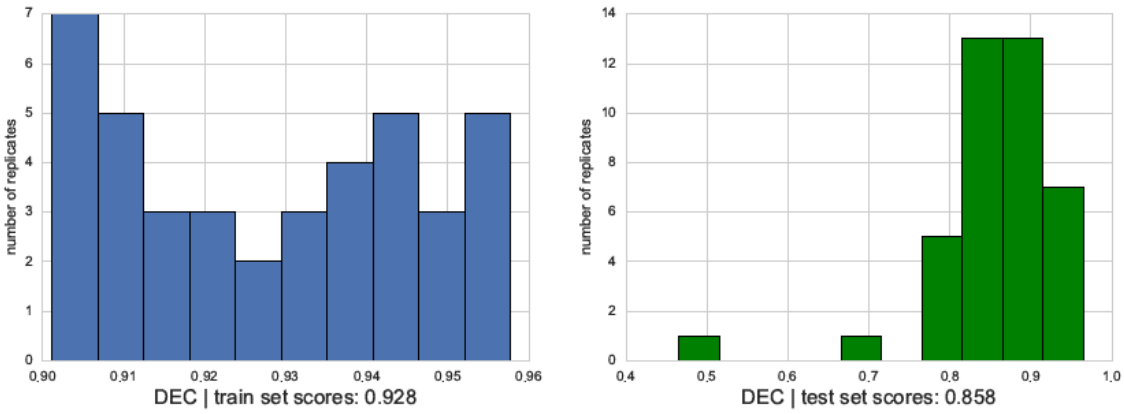
4.3 Principal Component Analyse (PCA)

Der erste Optimierungsansatz wurde mittels *Principal Components Analyse* durchgeführt. *PCA* stellte eine qualitative Analysemethode dar, welche mittels Koordinatentransformation multifunktionale Daten in einen niederdimensionalen Raum überführt. Dies soll zur Identifikation versteckter Muster in einem Datensatz verhelfen („ChemgaPedia“ o. J.). Darstellung 5 zeigt *DEC* Training- und Test-Score nach der *PCA*. Die abhängige Größe (aufgetragen auf der X-Achse) ist die Anzahl an Komponenten.



Darstellung 5: Trainings Score von DEC, PCA – Anzahl Parameter

Die hier angeführte Simulation ergab das beste Testergebnis bei zwei Komponenten. Die Auswertungen konnten nur geringfügig verbessert werden (in manchen Simulationen lag das Ergebnis auch unterhalb der Ausgangssituation). Die Auswertung hat sich zwar nur minimal verbessert, die Komplexität von 26 Datensätzen auf zwei konnte aber verringert werden. Das Balkendiagramm in Darstellung 6 zeigt stellt das Mittel von 40 Testläufen dar.



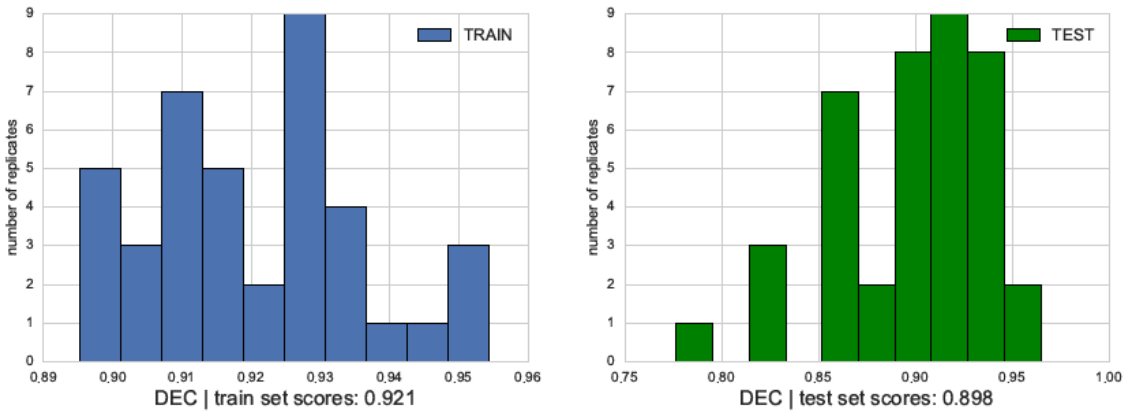
Darstellung 6: Mittel aus 40 Test-Scores nach *PCA*

4.4 Feature Selection

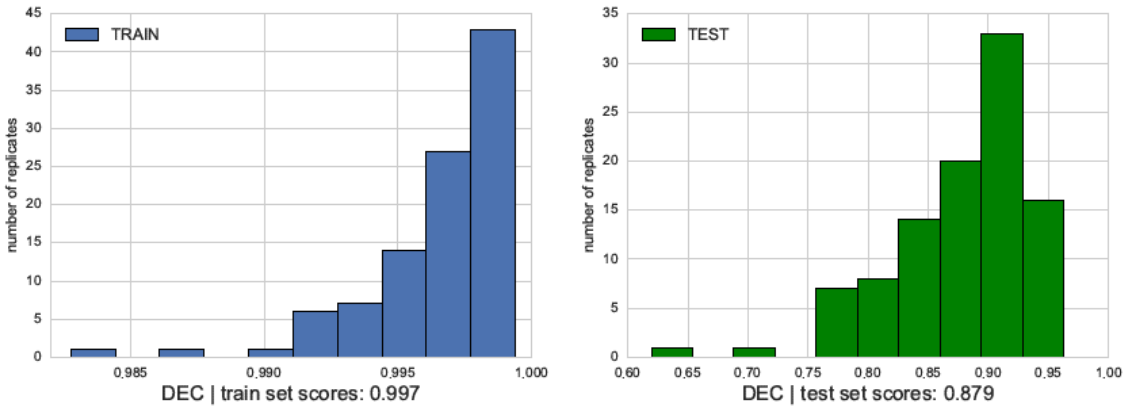
Der nächste Ansatz zur Reduktion von Komplexität und zur Optimierung der Analyse war *Feature Selection*. Dieses Verfahren zielt darauf ab Feature Datensätze auf Spaltenbasis zu reduzieren, die Anzahl der Datensätze bleibt aber konstant. Im Zuge der Projektarbeit wurden folgende Methoden verwendet:

- *Automatic Feature Selection - Univariate Statistics (AFS)*
- *Model-based Feature Selection (MbFS)*
- *Iterative Feature Selection (IbFS)*

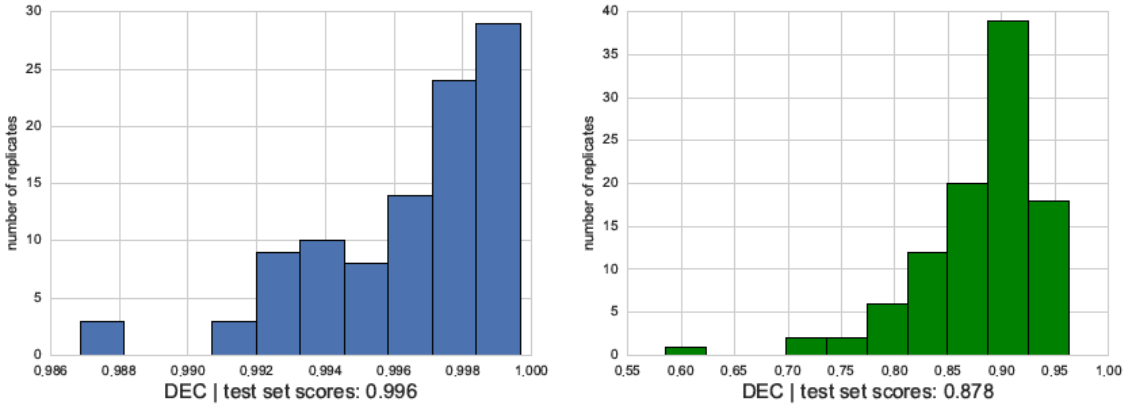
Es wurden alle Varianten mehrfach simuliert und geprüft. In den meisten Fällen wurden die Daten besser. Die Resultate sind in Darstellung 7 – Darstellung 9 abgebildet und beinhalten das Mittel aus 100 Wiederholungen.



Darstellung 7: *Automatic Feature Selection (AFS)*

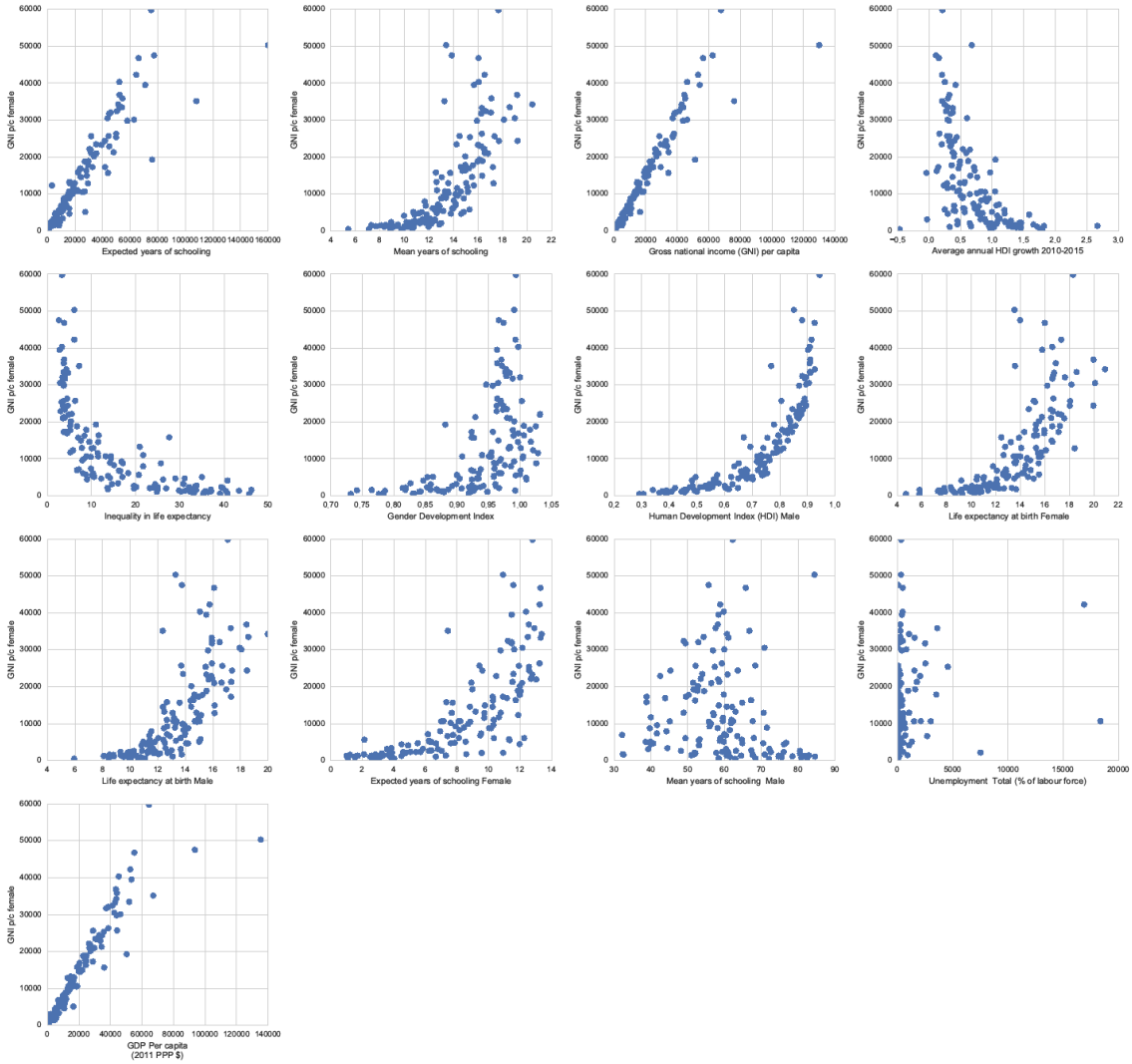


Darstellung 8: *Model-based Feature Selection (MbFS)*



Darstellung 9: *Iterative feature selection (IbFS)*

Die Auswahl der Feature Komponenten ist manchmal ähnlich, teilweise sehr ähnlich, aber auch unterschiedlich ausgefallen. An dieser Stelle habe ich entschieden die selektierten Datensätze nach *IbFS* weiter zu optimieren. Die Optimierung könnte aber auch mit allen Varianten (*MbFS*, *AFS)* weitergeführt werden. Abbildung 10 zeigt Korrelationen jener Features welche weiter verwendet werden. Eine ähnliche Selektion hätte auch manuell durchgeführt werden können, indem die Tragets einzeln mit der Zielgröße gegenübergestellt werden. Die besten Fits dienen dann als Bewertungskriterium zur Auswahl der Features.



Darstellung 10: Korrelationsmatrix nach *Iterative feature selection (IbFS)*

4.5 Model Evaluation

Mit dem reduzierten Feature Datensatz wurden zunächst verschiedene *Model Evaluationen* angewendet, um Parametereinstellungen zu verbessern. Die Parameteranalyse wurde mittels den Methoden *Simple Grid Search* und der sklearn Klasse *Grid Search CV*

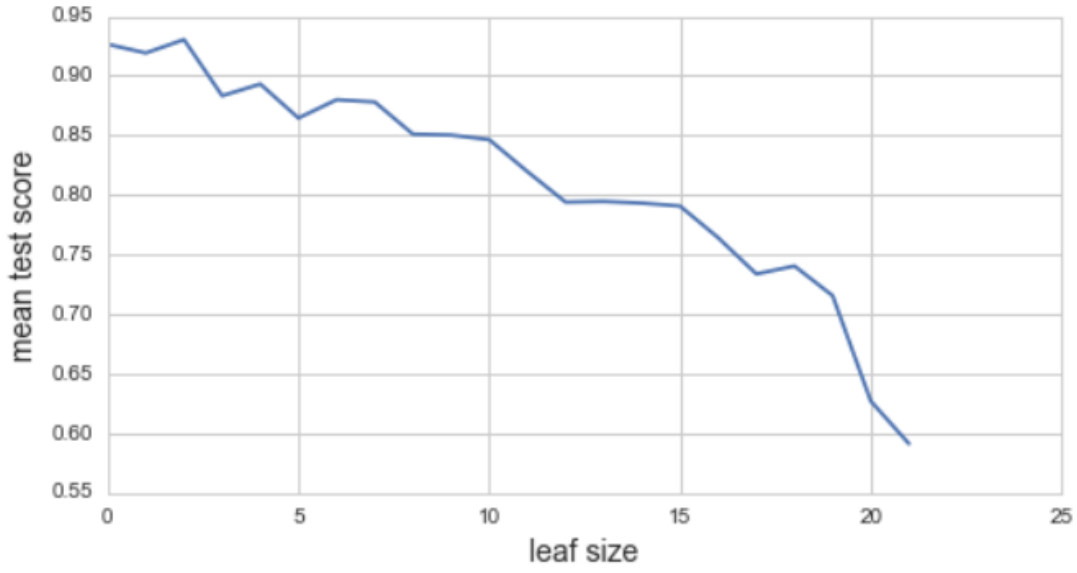
durchgeführt. Mit der Simple Grid Search Methode wird „manuell“ (innerhalb einer Schleife) nach dem bestgeeignetsten Parameter gesucht. Die sklearn Klasse GridSearchCV bietet eine komfortable Variante, um nach den richtigen Parameter zu suchen, ohne dass das Modell sich selbst beeinflusst. Daher können die Ergebnisse beider Varianten *(Grid Search CV und Simple Grid Search)* voneinander abweichen. Die folgenden Ergebnisse beziehen sich auf die bestmögliche Parametereinstellung des *Deciscion Tree* Parameter: *min_sample_leaf*. Die Auswertung nach *Simple Grid Search* ergab in allen Simulationen eine Bestparameterempfehlung von *min_sample_leaf: 3.* Die Parametervorgabe nach *Grid Search CV* ergab je nach Simulationsrunde einen Wert zwischen 1 und 3. Der Wert muss manuell im Python Skript übergeben werden.

```
          Best score: 0.904
          Best min_samples_leaf: 3

Out[116]: array([ 0.        ,  0.8600411 ,  0.89520374,  0.90400721,  0.89505735,
                  0.89707021,  0.84562549,  0.89138957,  0.88151397,  0.8815422 ,
                  0.87971453,  0.87971453,  0.87738555,  0.72818726,  0.72818726,
                  0.72818726,  0.72818726,  0.72710344,  0.72710344,  0.727265  ,
                  0.72737737,  0.72629964,  0.72651375,  0.72666791,  0.72678676,
                  0.72686024,  0.71614685,  0.71614685,  0.71614685,  0.71614685])
```

Darstellung 11: *Simple Grid Serach: min_samle_leaf: 3*

```
Test set score: 0.92
Best parameters: {'min_samples_leaf': 3}
Best cross-validation score: 0.93
```



Darstellung 12: *Grid Serach CV: min_samle_leaf: 2*

Nach der Feineinstellung des Parameters wurden Evaluationsmodelle verwendet, um die Test-Scores in möglichst vielen Varianten der Zeilen Reihenfolge zu testen. Alle Ergebnisse weisen einen verbesserten Score im direkten Vergleich zur Ausgangsituation mit dem gemittelten Testscore von 87.6% (siehe Abschnitt 4.2) auf. Die Ergebnis-Ausgabe der

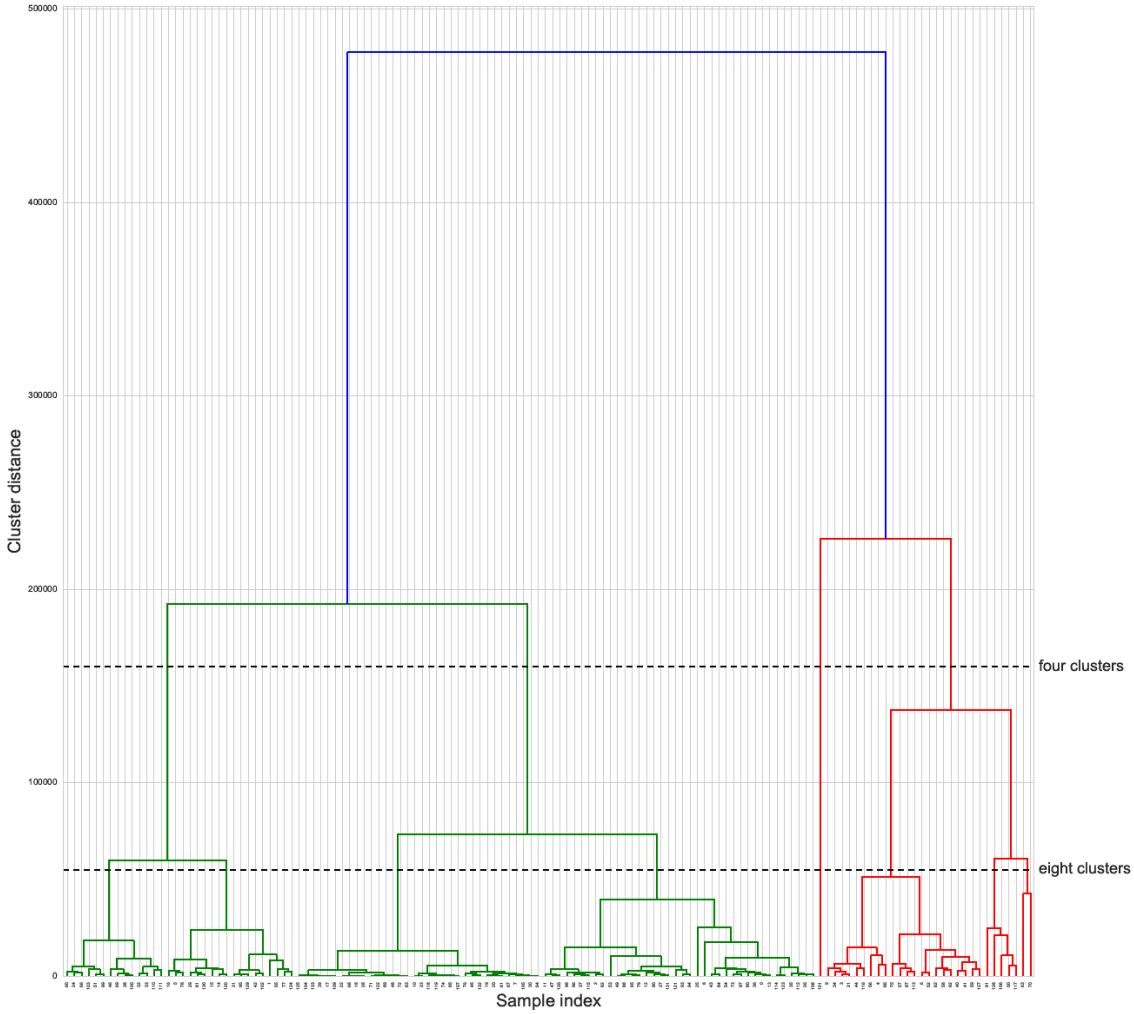
Evaluierung mittels *No Stratification cross-validation , shuffle-split cross-validation und Monte Carlo* wird kurz in der unten angeführten Tabelle zusammengefasst.

Model	Testscore
No Stratification cross-validation	**Mean accurancy:0.942**
Shuffle-Split cross-validation	**Mean accuracy: 0.885**
Iterative feature selection	**Mean accuracy: 0.889**

Darstellung 13: Finale Ergebnisse nach der Optimierung

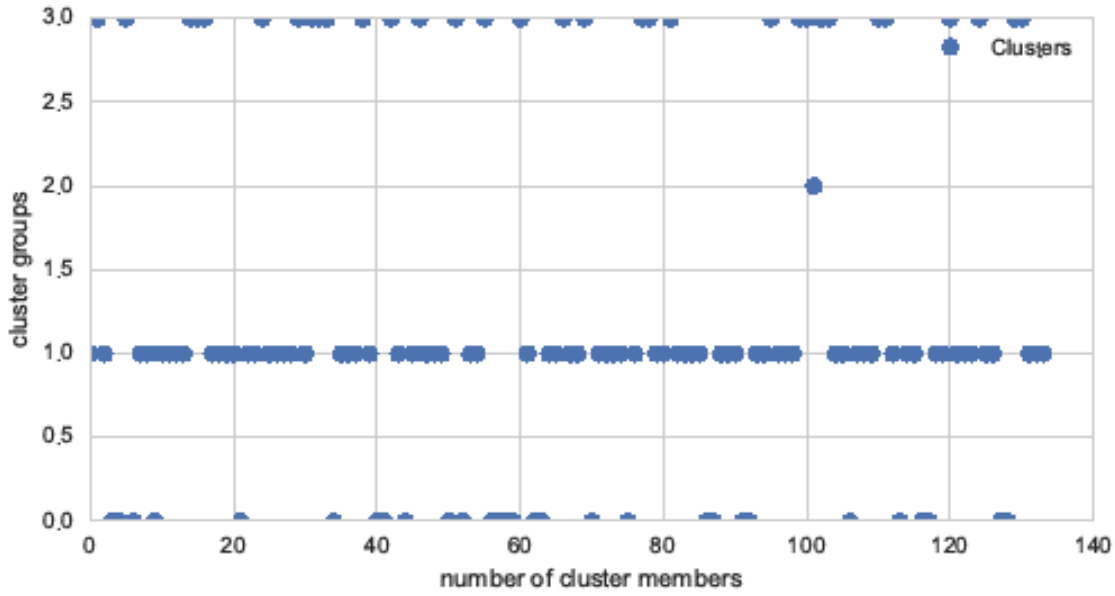
4.6 Clustering

Mit Hilfe der Clustering Methoden können Datensätze gruppiert werden. Ziel dieses *Clustering* Ansatzes ist es die Daten mit Hilfe der Clustermethode in Untergruppen zu unterteilen um anschließend die Regressionsvarianten an den Untergruppen anzuwenden. Ziel ist es Untergruppen zu finden, die besser korrelieren als der Gesamtdatensatz. Darstellung 14 zeigt zwei Gruppierungsansätze, welche weiter untersucht wurden (Variante 1: vier Cluster, Variante 2: acht Cluster). Auf der horizontalen Achse sind die einzelnen Datensätze aufgetragen. Die senkrechten Linien betrachten die Gruppenstärke. Das erste Cluster, welches ich verwendet habe, besteht aus 4 Gruppen und ist in Abbildung 14 strichliert gekennzeichnet. Die Gruppenstärke des Vierer Cluster kann aus der Grafik anhand Verzweigungen visuell entnommen werden.



Darstellung 14: *Cluster* Übersicht

Innerhalb des Vierer Clusters gibt es eine Spalte, welche viele Datensätze beinhaltet. Verschiedene Fits haben gezeigt, dass die Aussagekraft in dieser Ebene des Clusters die höchste Vorhersagekraft innerhalb der Clusterbildungen und im Vergleich zum Anfangswert aufweist. Der Validierte-Score aus Abschnitt 4.6 konnte nicht übertroffen werden. Darstellung 15 zeigt die Größe des Clusters. Interessant zu bemerken ist, dass bis zu einer hohen Cluster-Distanz ein Single Cluster entstanden ist.



Darstellung 15: Verteilung der Clustergrößen

5. Persönliches Fazit

Innerhalb des Projekts wurden viele Methoden angewandt um den Score dauerhaft über 90% anzuheben. Sofern dies gelungen ist, hatte es meistens zwei Gründe: Entweder hat sich der Wert bei mehrfachen Tests wieder unter 90% eingestellt, oder mir ist ein Programmierfehler aufgefallen, welcher das Ergebnis positiv verfälscht. Das war auch einer der Gründe, warum ich den Parameter *Random State* beim *Split* nicht fixiert habe. Ich wollte sicherstellen, dass ich mich nicht von einem guten Ergebnis kurzfristig beeinflussen lasse.

Neben der Optimierung des Ergebnisses war mein persönliches Ziel, mich mit den Methoden aus dem Unterricht und dem Buch auseinanderzusetzen. Das abgegebene Python-File zeigt bei Weitem nicht die Vielzahl von Anwendungsversuchen der verschiedenen Methoden aus dem Unterricht. Die Übung war mit einem großen Zeitaufwand verbunden, aber sie war hilfreich die Anwendungskompetenz der Unterrichtsschwerpunkte zu erhöhen und routinierter mit Python arbeiten zu können. Rückblickend würde ich die Abarbeitung eines neuen Projektes strukturierter und zielgerichteter gestalten.

Literaturverzeichnis

„ChemgaPedia“ (o. J.): ChemgaPedia. Online im Internet: URL: http://www.chemgapedia.de/vsengine/vlu/vsc/de/ch/13/vlu/daten/multivariate_datenanalyse_sensor/hauptkomponentenanalyse.vlu.html

„Das Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung (BMZ)“ (2010): Index der menschlichen Entwicklung (HDI). Das Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung (BMZ). Online im Internet: URL: https://www.bmz.de/de/service/glossar/I/index_hdi.html

„kaggle - Your Home for Data Science“ (2017): kaggle - Your Home for Data Science. Online im Internet: URL: https://www.kaggle.com/

Müller Andreas; Guido Sarah (2017): Introduction to Machine Learning with Python, A Guide for Data Scientists. O’Reilly Media.

„United Nations Development Programme Human Development Reports“ (2017): United Nations Development Programme Human Development Reports. Online im Internet: URL: http://hdr.undp.org/en/data