Data Mining

18.06.2021

Fakultät für Ingenieurwissenschaften
Bereich Elektrotechnik und Informatik
C. Werner, J. Prothmann
www.hs-wismar.de

Gliederung

Vorverarbeitung

Rohdatensatz

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1	female	group B	bachelor's degree	standard	none	72	72	74
2	female	group C	some college	standard	completed	69	90	88
3	female	group B	master's degree	standard	none	90	95	93
4	male	group A	associate's degree	free/reduced	none	47	57	44

Bild 1: Rohdatensatz

Datenvorverarbeitung

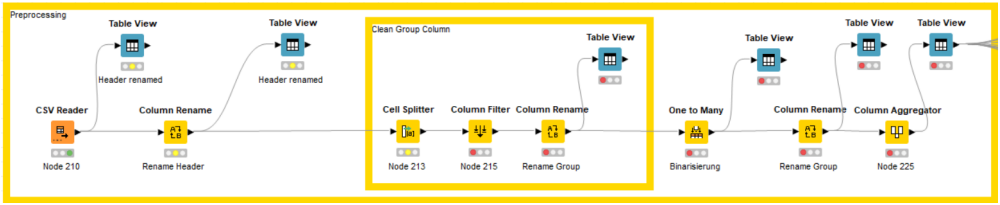


Bild 2: Knime Workflow zur Vorverarbeitung

gender	parental education	lunch	preparation	math	reading	writing	group	bachelor	college	master	associate	high school	some high school	standard	nonstandard	none	completed	B	C	A	D	E	Mean
female	bachelor's degree	standard	none	72	72	74	B	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	72.66666666666666
female	some college	standard	completed	69	90	88	C	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	82.33333333333333
female	master's degree	standard	none	90	95	93	B	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	92.66666666666666
male	associate's degree	free/reduced	none	47	57	44	A	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	49.33333333333333
male	some college	standard	none	76	78	75	C	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	76.33333333333333
female	associate's degree	standard	none	71	83	78	B	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	77.33333333333333

Bild 3: Vorverarbeitete Daten

Entscheidungsbäume

Decision Tree Learner

- Standardknoten von Knime
- Zielattribut: nominal
- Entscheidungsfindungsattribute: nominal, numerisch
- Qualitätsmaße für Splitberechnung:
  - Gini-Index
  - Gain-Ratio
- Pruning möglich

SimpleCart

- Weka-Knoten
- Erzeugung von Binärbäumen
- Pruning möglich
- Je höher der Informationsgehalt eines Attributs in Bezug auf die Zielgröße, desto weiter oben im Baum findet sich dieses Attribut.

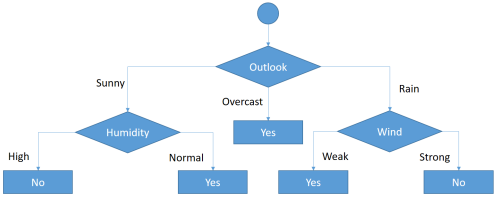


Bild 4: CART Tree Beispiel

J48

- Weka-Knoten
- C4.5 Algorithmus von J. Ross Quinlan
- Ähnlich zu CART, jedoch kein Binärbaum
- Deutlich breiter und weniger tief als CART
- Pruning möglich

NBTree

- Weka-Knoten
- Hybridalgorithmus aus Entscheidungsbaum- und Naive-Bayes-Klassifikatoren
- „klassische“ Knoten
- Blätter enthalten Naive-Bayes’sche Klassifikatoren

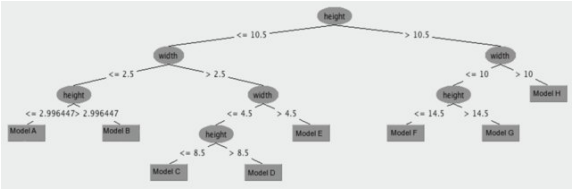


Bild 5: NB Tree Beispiel

REPTree

- Weka-Knoten
- basiert auf C4.5 Algorithmus
- Generierung unter Berücksichtigung von:
  - Informationsgewinn
  - Varianz

LMT

- Weka-Knoten
- Blätter: lineare Regressionsfunktionen
- stufenweiser Anpassungsprozess
- Automatische Auswahl relevanter Attribute

DecisionStump

- Weka-Knoten
- einstufiger Entscheidungsbaum
- Vorhersage anhand des Wertes eines Eingabe-Features
- Knoten: Schwellenwert
- Blätter: Werte unterhalb und oberhalb des Schwellenwerts
- Einsatz als „schwache Lerner“ (z.B. Gesichtserkennung)

J48Graft

- Weka-Knoten
- nutzt den C4.5++ Algortihmus
- Verbesserung durch „all-tests-but-one-partition“ (ATBOP)
- Reduzierte Rechenzeit
- Reduzierte Komplexität des Baums

BFTree

- Weka-Knoten
- Best-First-Entscheidungsbaum
- „beste“ Knoten zuerst expandieren
- „beste“ Knoten: maximalen Reduktion der Unreinheit (z.B. Gini-Index)
- resultierende Baum nur in Reihenfolge unterschiedlich

RandomTree

- Weka-Knoten
- zufällig ausgewählte Attribute an den Knoten
- kein Pruning

RandomForest

- Weka-Knoten
- Kombination von Baumprädiktoren
- Abhängigkeit jedes Baumes von Werten eines Zufallsvektors
- Zufallsvektor: unabhängig und besitzt gleiche Verteilung für alle Bäume im ‘Wald’

Cluster

Cluster

- kMeans
- Dichtebasiertes Clustern
- Hierarchisches Clustern

kMeans Algortihmus

- 3 Schritte: 1. Initialisierung, 2. Zuordnung, 3. Aktualisierung
- Wiederholen von Schritt 2 und 3 bis Abbruchbedingun erreicht
- kMeans Knoten ist in Auslieferungsversion von KNIME enthalten
- Keine dynamische Anzahl an Cluster
- Abbruchbedingung entweder max Iterationen oder Schritt 2 und 3 bringen keine Änderungen mehr
- Distanzberechnung mit euklidischer Distanz (Lineare Distanz von 2 Punkten im Raum)

Dichtebasiertes Clustern

- DBSCAN Knoten in KNIME
- Density-Based Spatial Clustering of Applications with Noise
- Unterteilung der Daten in 3 Kategorien: Core Punkte, Border Punkte, Noise Punkte
- Clusterbildung durch verbinden von Core Punkten
- Punkte innerhalb der Core Punkte - Distanz zählen zum Cluster, alle außerhalb sind Noise

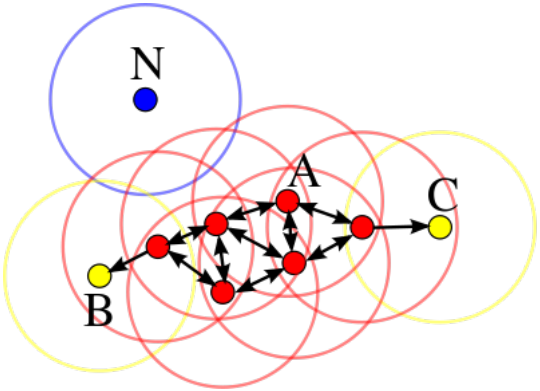


Bild 6: DBSCAN Algorithmus

Hierarchisches Clustern

- Sowohl Build-In Knoten als auch WEKA Extension
- Berechnen der Punktdistanzen durch diverse Distanzmaße (Euklidische-, Manhatten-, ...-Distanz)
- Beide Knoten sind agglomerativ (bottom-up): Iterative Bildung von großen Clustern aus bestehenden
- Darstellung in Dendrogrammen

Silhouettenkoeffizient

- Berechnet die Qualität von Clustern
- Berechnent für jede Zeile wie gut das ausgewählte Cluster passt
- Reichweite von -1 bis 1 $\rightarrow$ je höher der Wert, desto besser die Clusterung

$$S(o) = \begin{cases} 0 & \text{wenn } o \text{ einziges Element von } A \text{ ist} \\ \frac{\text{dist}(B,o) - \text{dist}(A,o)}{\max\{\text{dist}(A,o), \text{dist}(B,o)\}} & \text{sonst} \end{cases}$$

Bild 7: Formel Silhouettenkoeffizient

Implementierung

Gesamtworkflow Entscheidungsbäume

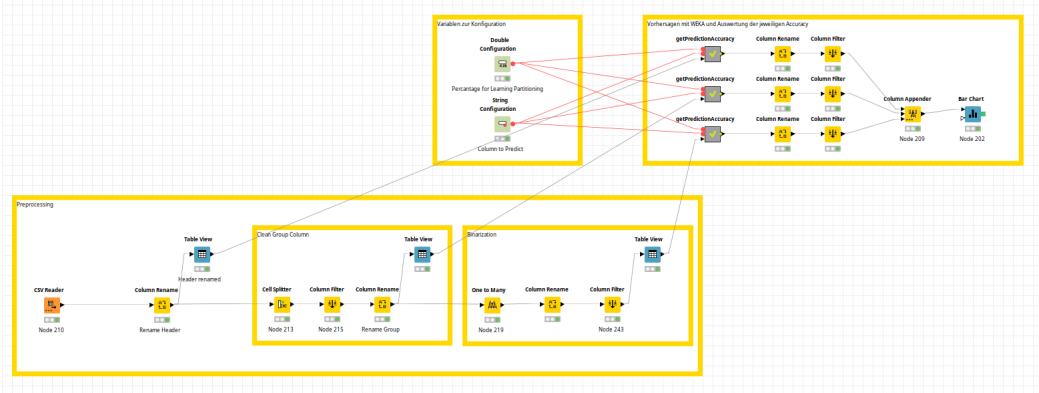


Bild 8: Gesamtworkflow

Ermittlung der Accuracys

- Knoten zur Eingabe von:
  - vorherzusagender Spalte
  - Trainingsdatenaufteilung
- Extrahierung der Accuracys und Anzeige in Balkendiagramm

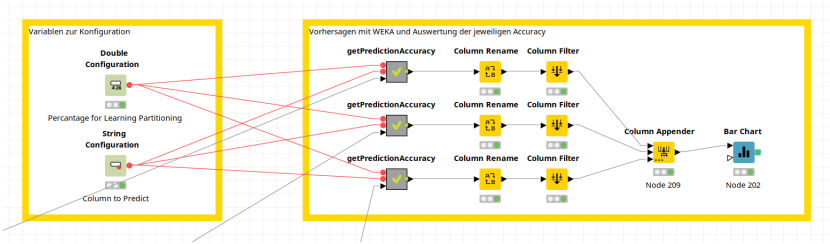


Bild 9: Ermittlung der Accuracys

Metaknoten ‘getPredictionAccuracy’

- Aufteilung in Trainings und Testdaten
- Extrahierung der Accuracys

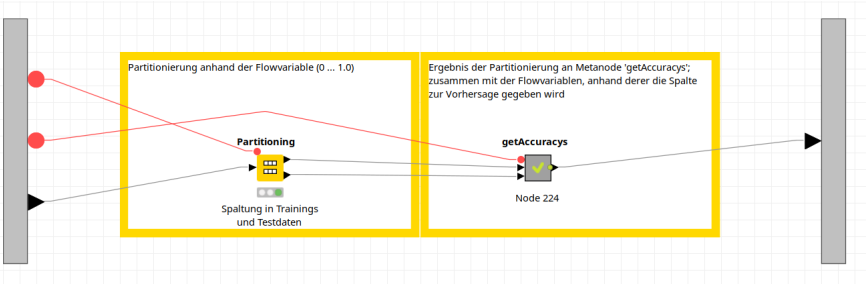


Bild 10: Inhalt des Metaknoten ‘getPredictionAccuracy’

Metaknoten ‘getAccuracys’

- Metaknoten für verschiedene Entscheidungsbäume
- Gleiche Trainings- und Testmenge
- Zusammenführung der Accuracys in eine Tabelle

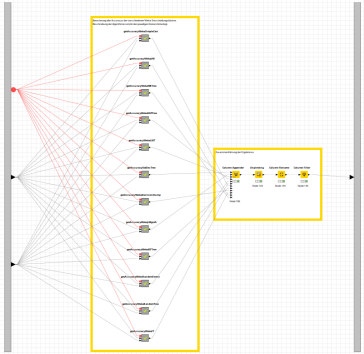


Bild 11: Inhalt des Metaknoten ‘getAccuracys’

Beispielhafter ‘getAccuracyWeka*’ Knoten

- Lerner
- Vorhersage
- Scoring
- Extahierung der Accuracy

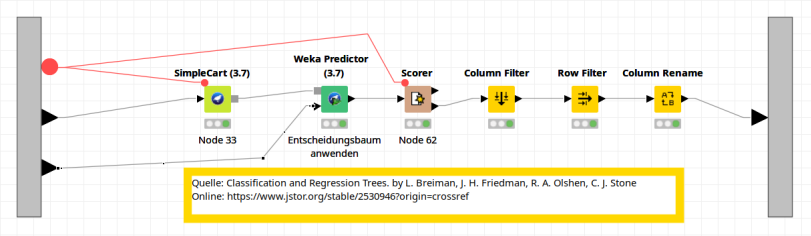


Bild 12: Inhalt des Metaknoten ‘getAccuracyWekaSimpleCart’

Accuracy Chart

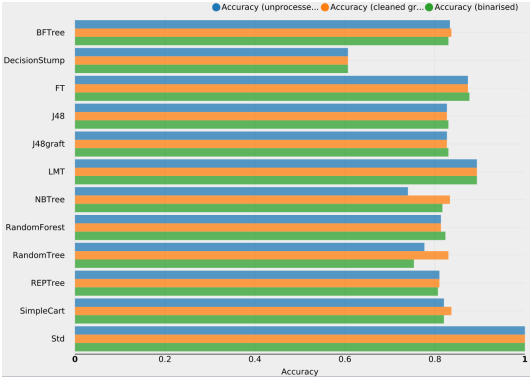


Bild 13: Balkendiagramm für ‘gender’ und Trainingssatz von 70%

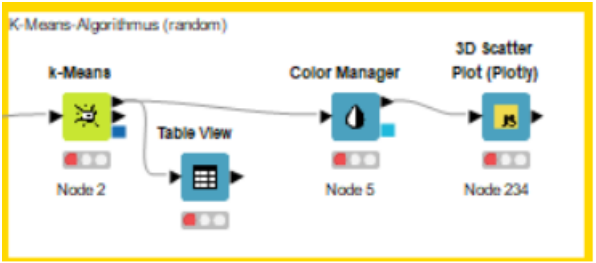


Bild 14: Implementierung des kMeans Algorithmus in KNIME

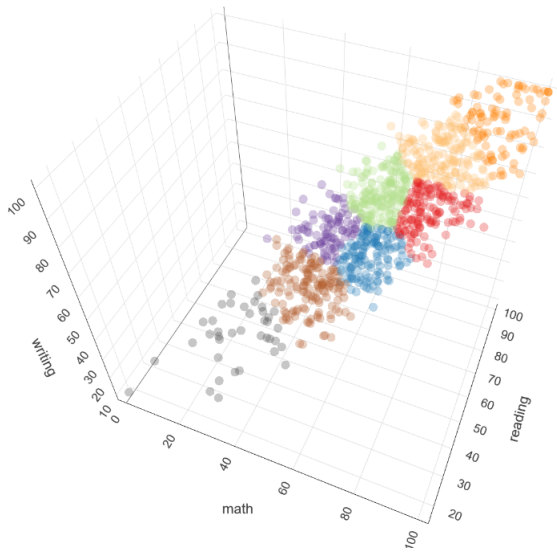


Bild 15: Ergebnisse der kMeans Clusterung in KNIME

RowID	Mean Silhouette Coefficient
cluster_1	0.23250713131400805
cluster_7	0.3003979971435271
cluster_5	0.3647558863935451
cluster_2	0.33663805036620026
cluster_0	0.3491567522765088
cluster_4	0.29479048752917436
cluster_6	0.29271663786378105
cluster_3	0.2868838307286652
Overall	0.30065149235523864

Bild 16: Random Cluster Initialisierung

RowID	Mean Silhouette Coefficient
cluster_4	0.285934358318074
cluster_1	0.27698431365825305
cluster_2	0.3747220515647596
cluster_3	0.3362331798056654
cluster_7	0.3519141075987184
cluster_5	0.30796418885837923
cluster_0	0.2716289533585836
cluster_6	0.2845174544293571
Overall	0.30354359857262203

Bild 17: First k Rows CLuster Initialisierung

Overall	0.37111894938059187

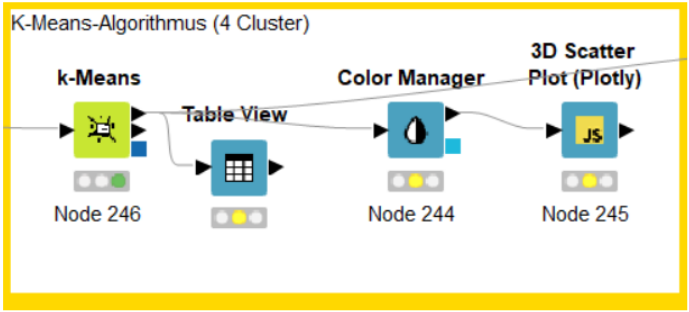


Bild 18: Beste Clusterleistung und first-k-rows

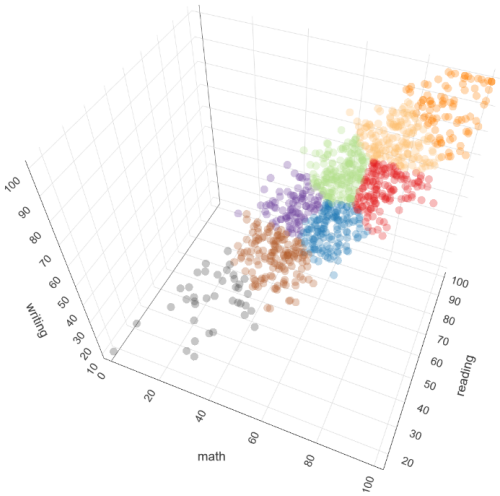


Bild 19: Erster Clusterung

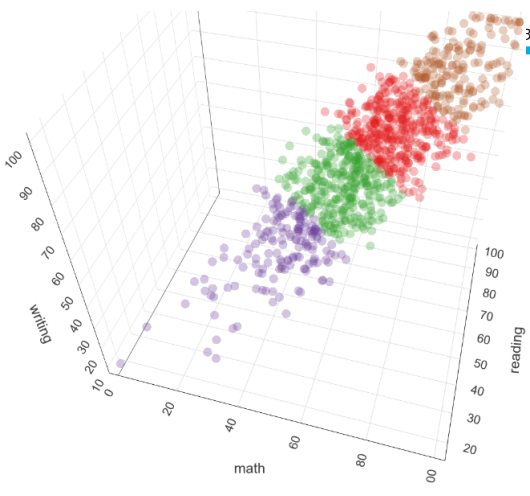


Bild 20: Beste Clusterung

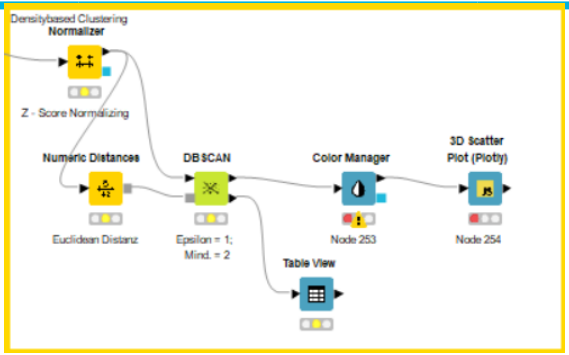


Bild 21: Implementierung des DBSCAN Algorithmus in KNIME

RowID	Mean Silhouette Coefficient
Cluster_0	0.6915721844630638
Noise	0
Overall	0.6908806122786022

Bild 22: Clusterbewertung DBSCAN

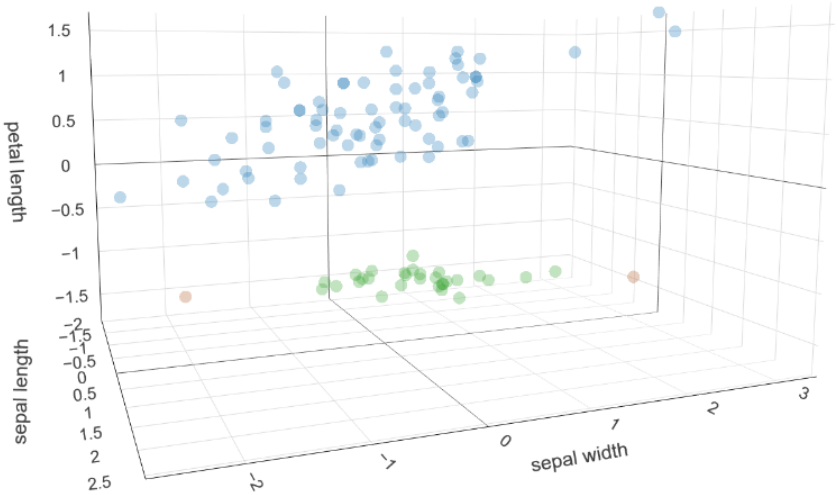


Bild 23: Iris Datensatz DBSCAN

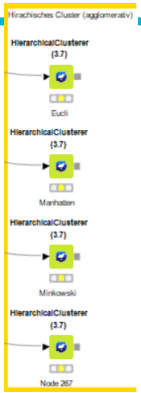


Bild 24: Implementierung des WEKA Hierarchichen Clusterer in KNIME

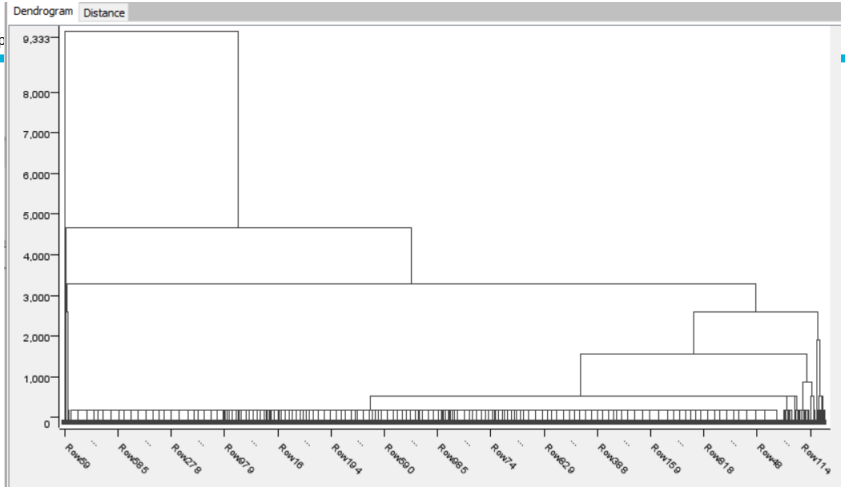
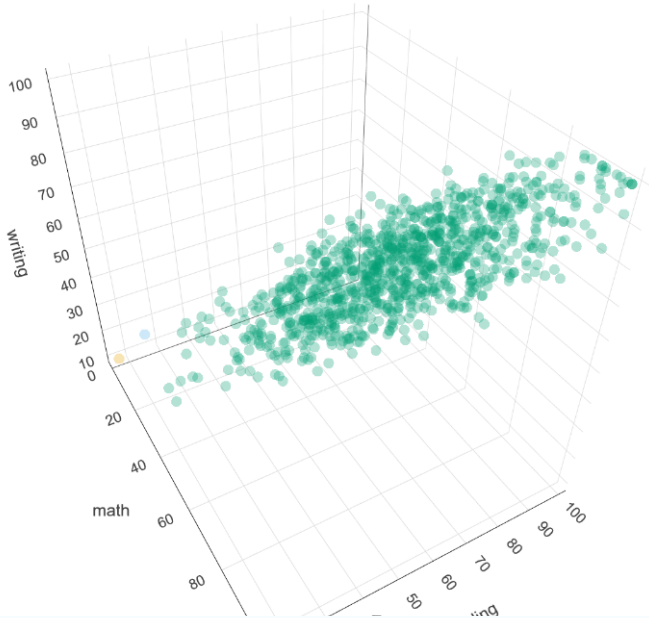


Bild 25: Dendrogramm Hierarchischer Clusterer


writing
100
90
80
70
60
50
40
30
20
10
0
math
0
20
40
60
80
100
90
80
70
60
50