Machine Learning Übungsblatt 2

Ramon Leiser Tobias Hahn

November 16, 2016

1 Begriffsdefinitionen

1.1 Klassifikation - Clustering

Klassifikation ist eine Methode um verschiedene Messungen einer jeweiligen Klasse zuzuordnen. Das ist eine Aufgabe für supervised Learning, d.h. wir haben Testbeispiele mit Daten und Klassenzuordnung die wir trainieren und daraus abstrakte Aussagen über Klassenzugehörigkeit treffen wollen, also Algorithmen trainieren die uns für beliebige Daten möglichst die korrekte Klasse liefern.

Clustering ist eine Methode um aus einer Menge aus Daten ähnliche Gruppen von Punkten zu extrahieren. Das ist eine Aufgabe für unsupervised Learning, d.h. wir haben nur die Daten ohne irgendeine Zuordnung. Das ist der eine große Unterschied zwischen Clustering und Klassifikation, dass wir Punkte nicht in Klassen einordnen sondern nur aufgrund von Strukturen in den Daten Cluster bilden. Außerdem ist es beim Clustering kein Problem einem Punkt mehreren Clustern zuzuordnen, so z.B. beim hierarchischem Clustering. Bei der Klassifikation wollen wir immer nur eine Klasse herausfinden für einen Datenpunkt.

1.2 Kernobjekt - Dichteerreichbarkeit - Dichteverbunden

1.2.1 Kernobjekt

Bei DBSCAN hat man für den Clustering-Algorithmus zwei Parameter - den Radius, welcher definiert was eine Nachbarschaft eines Punktes ist, sowie eine Anzahl an Nachbarn, also eine Anzahl von Punkten die mindestens in der Nachbarschaft eines Punktes sind damit dieser als dicht gilt. Hat ein Punkt mindestens die Anzahl der Nachbarn in seiner Nachbarschaft, so gilt er als dich und somit als Kernobjekt.

1.2.2 Dichterreichbar

Wie wir vorher festgestellt haben gibt es bei DBSCAN Kernobjekte. Ein Punkt gilt als dichteerreichbar falls er in der Nachbarschaft eines Kernobjekts liegt, aber selber kein Kernobjekt ist, also zuwenig Nachbarn hat.

1.2.3 Dichteverbunden

Zwei Punkte A und B sind dichteverbunden, wenn es einen Pfad bestehend aus Kernobjekten zwischen ihnen gibt. Das heißt dass Punkte A von einem Kernobjekt dichteerreichbar sein muss, welches wiederum mit anderen Kernobjekten verbunden ist, bis wir beim Punkt B sind. Alle Punkte die untereinander dichteverbunden sind bilden einen Cluster.

1.3 Lazy learning

Lazy learning bedeutet dass wir kein Modell unserer Trainingsdaten erstellen und dieses dann auf Testdaten anwenden, sondern erst ein Modell erstellen sobald wir eine Testaufgabe bekommen. Ein Beispiel welches wir schon hatten war k-Means, wo die Centroide und Cluster erst angepasst wurden als wir einen

Punkt einem Cluster zuordnen mussten, und nicht Cluster mit allen Punkten erstellt wurden und diese dann angepasst. Lazy learning beschreibt ein Paradigma in Machine Learning, hat also nicht unbedingt mit Clustering und Klassifikation zu tun.

2 DBSCAN

2.1 k-Distanzdiagramm

Für die initiale Berechnung von k und MinPts brauchen wir als Parameter die Dimensionalität unserer Daten. In unserem Fall geht es um Punkte auf der zweidimensionalen Ebene (Punkte auf der Karte), darum ist unsere Dimensionalität 2, bzw. d = 2. Die Gleichungen für die Anfangeswerte sind dann:

$$k = (2 * d) - 1 = 4 - 1 = 3$$
$$MinPts = k + 1 = 3 + 1 = 4$$

2.1.1 Graphen

Im folgenden die Graphen für die gegebene Anzahl von k:

Figure 1: k = 1

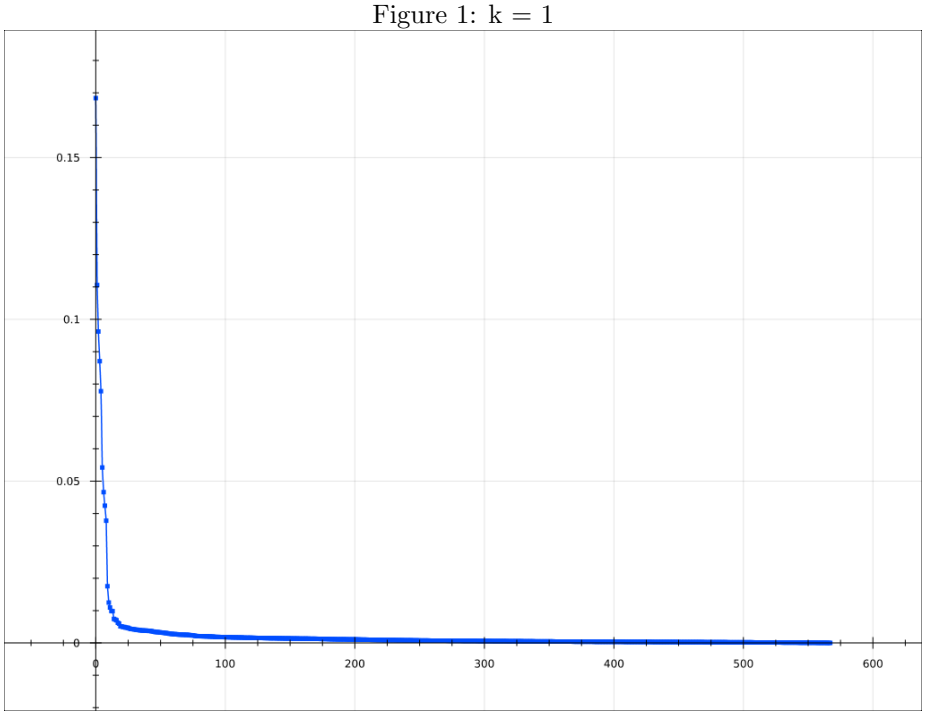


Figure 2: k = 2

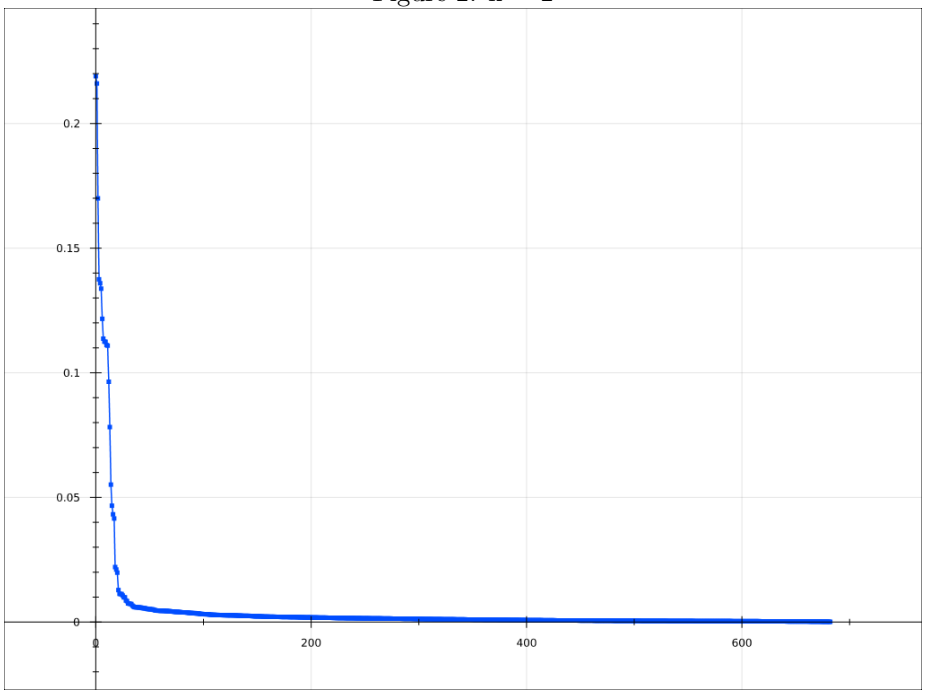


Figure 3: k = 3

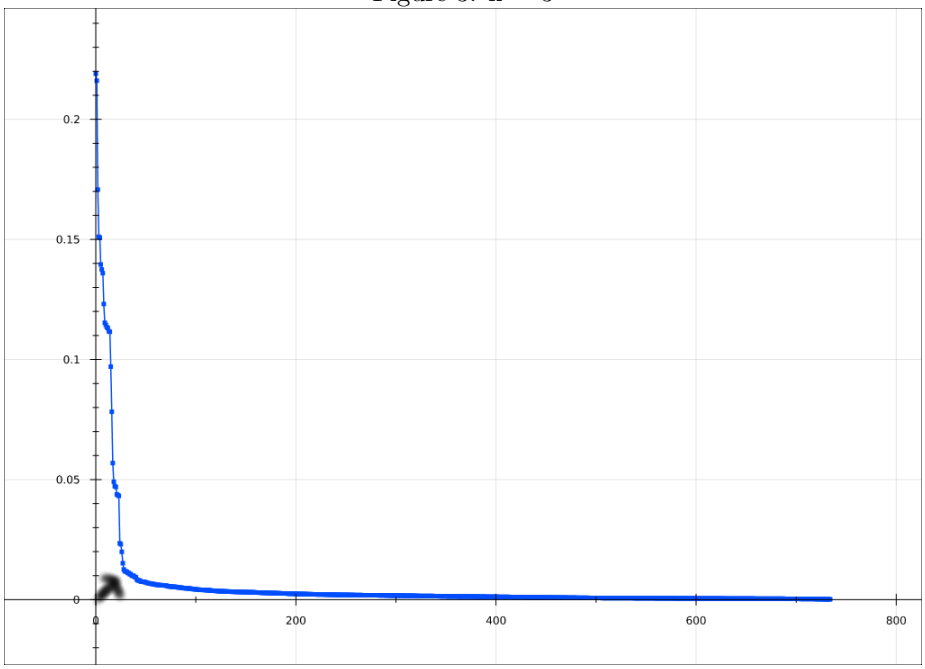


Figure 4: k = 4

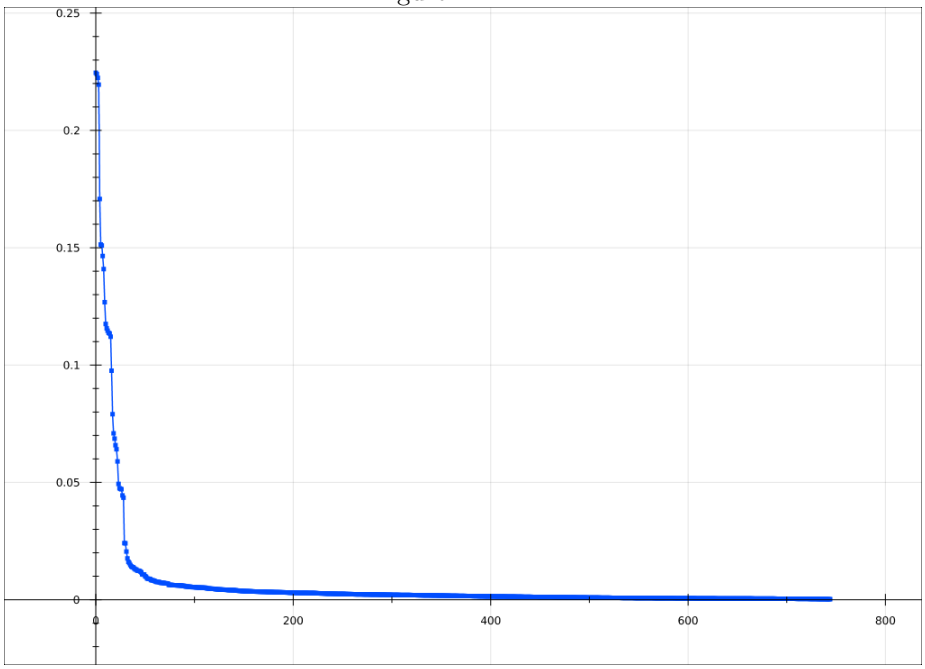


Figure 5: k = 5

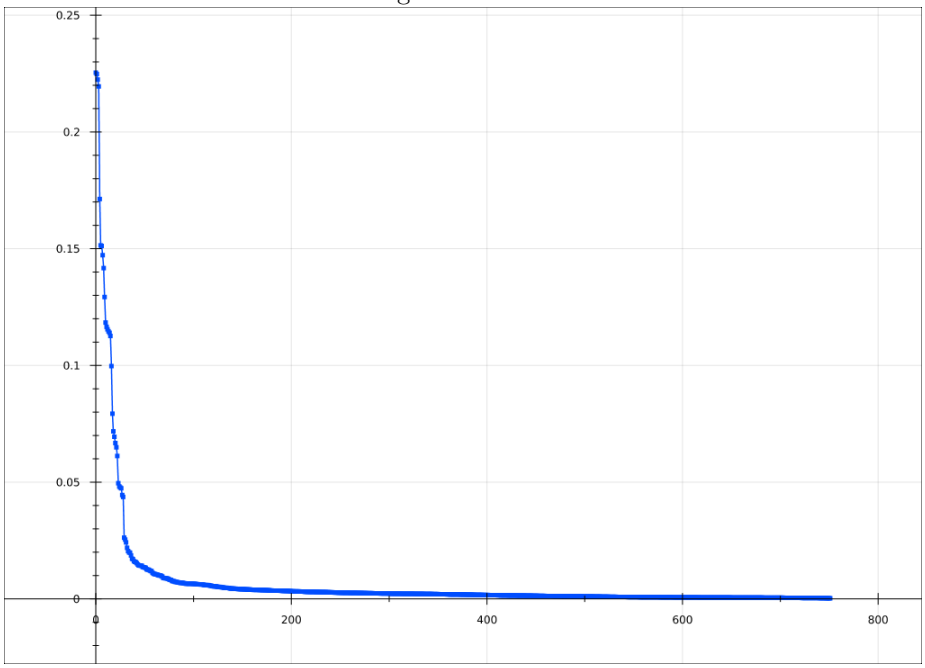
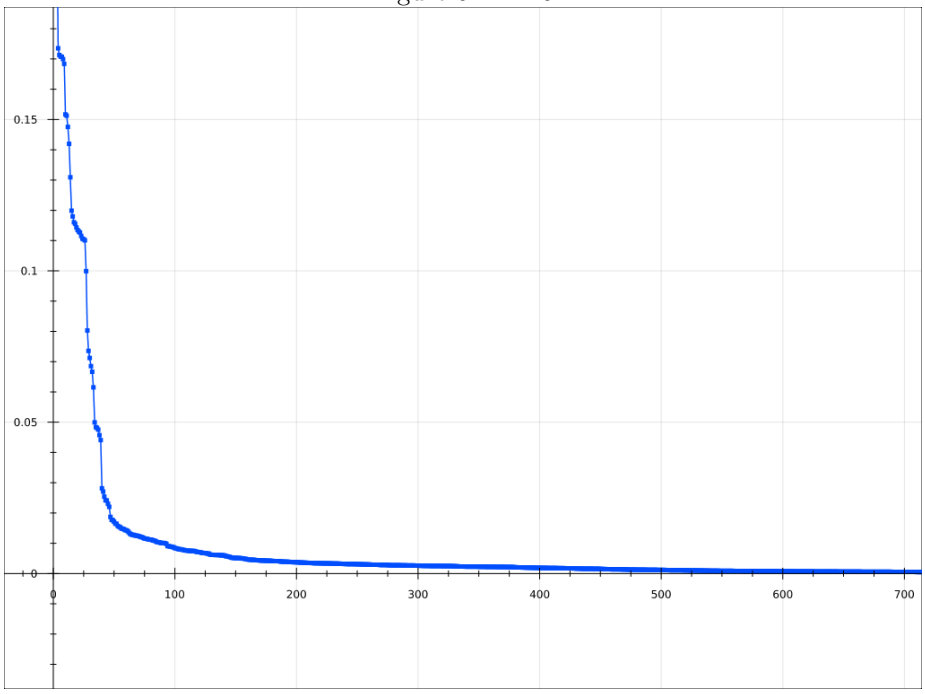


Figure 6: k = 6



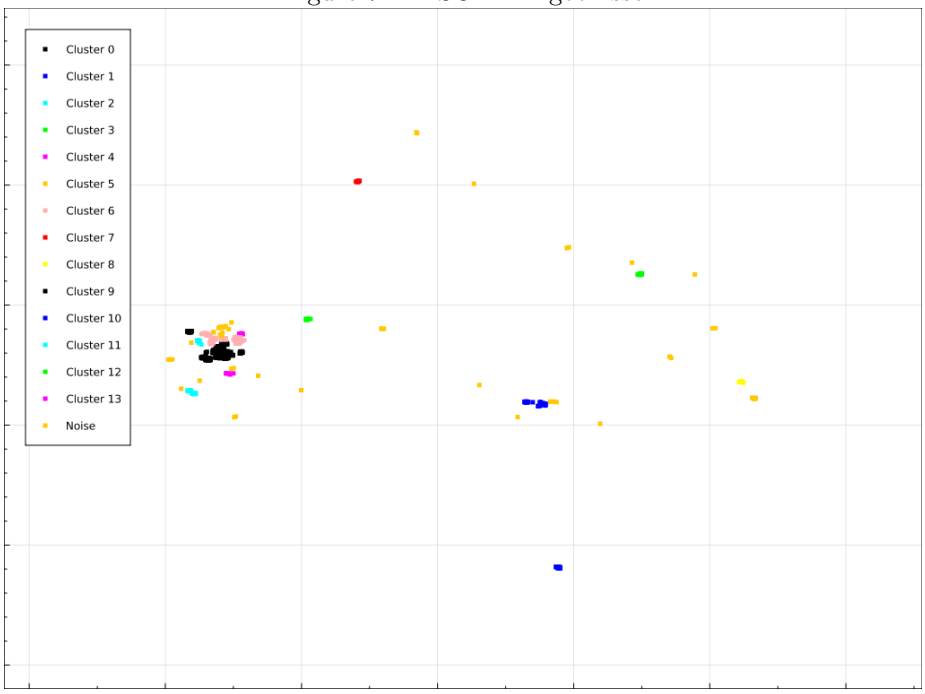
2.1.2 Auswahl Parameter

Wie im Graphen 3 markiert, wählen wir das Grenzobjekt mit den Werten MinPts = 4 und Epsilon = 0.012563822348311689 aus, wobei wir Epsilon der Einfachheit halber auf 0.012 abrunden.

2.2 DBSCAN

Mit den vorhin bestimmten Parametern wurde der DBSCAN-Algorithmus auf unsere Daten angewandt. Dabei kam folgendes Clustering zustande:

Figure 7: DBSCAN-Ergebnisse



Die verlangten Kennziffern sind im folgenden aufgelistet:

KENNZAHLEN
Anzahl Cluster: 14
Kosten fuer el Jefe: 50.861321
Anzahl der Noise-Punkte: 43
Silhouttenkoeffizient: 0.702970

2.3 Vergleich DBSCAN und kMeans

Um die beiden Algorithmen vergleichen zu können, im folgenden nocheinmal die Ergebnisse der beiden Verfahren als Plot:

2.3.1 kMeans Clustering

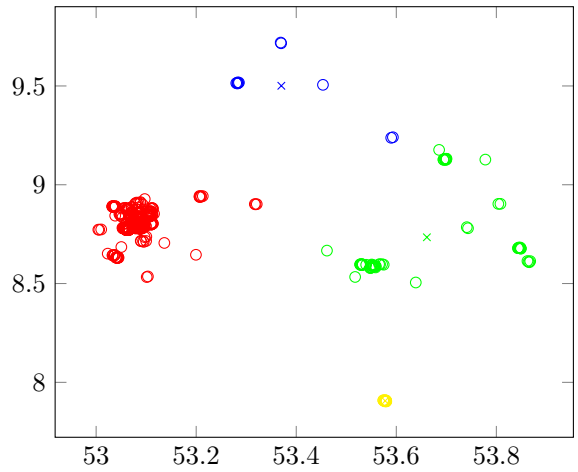
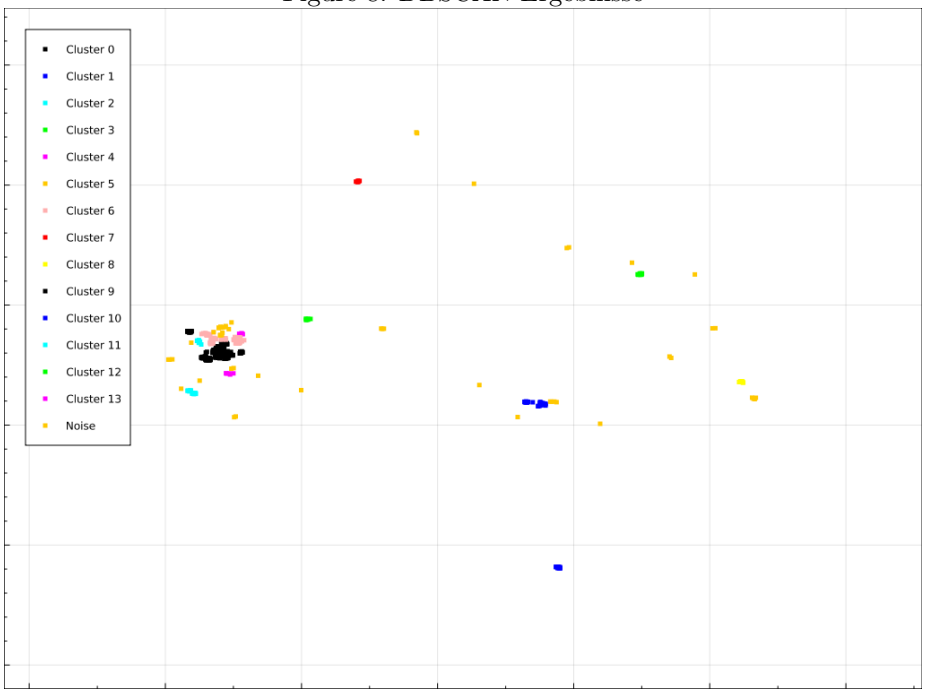


Figure 8: DBSCAN-Ergebnisse



Zum Vergleich: Die Kosten in Gummimünzen sind beim Dichteclustering viel höher, und es werden mehr Fahrer benötigt. Das ist natürlich für den Chef ein Nachteil, weswegen dieses Clustering wahrscheinlich nicht gewählt werden wird. Für die Kunden ist außerdem von Nachteil, dass beim DBSCAN Clustering manche von ihnen nicht mehr angefahren werden, da manche Punkte Noise sind und damit keinem Cluster zufallen. Dafür haben sie allerdings den Vorteil, schneller zu beliefert werden, da ein Fahrer weniger von ihnen betreut.

3 Descision Tree Learning

3.1 Mathematik

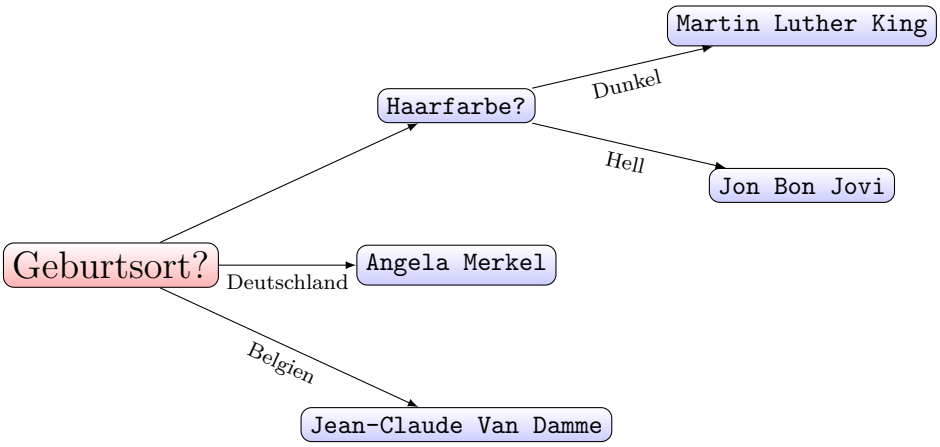
Der erste Schritt im lernen eines neuen Entscheidungsbaumes besteht darin den Informationsgehalt der zu klassifizierenden Fälle zu bestimmen um später den `InformationGain()` zu berechnen. Da jedes Ergebnis in der Tabelle ein unterschiedliches ist ergibt sich für $I(T) = I(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}) = 2$.
Für die jeweiligen Entscheidungskategorien ergibt sich:
$Info(Geburtsort, T) = \frac{1}{4} * I(\frac{1}{1}) + \frac{1}{4} * I(\frac{1}{1}) + \frac{1}{2} * I(\frac{1}{2}, \frac{1}{2}) = \frac{1}{4} * 0 + \frac{1}{4} * 0 + \frac{1}{2} * 1 = 0.5$
$Gain(Geschlecht, T) = Info(T) - Info(Geschlecht, T) = 2 - 0.5 = 1.5$

$Info(Geschlecht, T) = \frac{1}{4} * I(\frac{1}{1}) + \frac{3}{4} * I(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) = \frac{1}{4} * 0 + \frac{3}{4} * 1.585 = 1.187$
$Gain(Geschlecht, T) = Info(T) - Info(Geschlecht, T) = 2 - 1.187 = 0.813$

$Info(Haarfarbe, T) = \frac{1}{2} * I(\frac{1}{2}, \frac{1}{2}) + \frac{1}{2} * I(\frac{1}{2}, \frac{1}{2}) = \frac{1}{2} * 1 + \frac{1}{2} * 1 = 1$
$Gain(Haarfarbe, T) = Info(T) - Info(Haarfarbe, T) = 2 - 1 = 1$
Demnach ist der Informationgain am höchsten bei der Unterscheidung nach Geburtsort. Bei dieser Unterscheidung entstehen direkt zwei Blattknoten und am dritten Knoten muss nur noch zwischen zwei Fällen unterschieden werden: $I(T_{USA}) = I(\frac{1}{2}, \frac{1}{2}) = 1$.

$Info(Haarfarbe, T_{USA}) = \frac{1}{2} * I(\frac{1}{2}) + \frac{1}{2} * I(\frac{1}{2}) == 0$
$Gain(Haarfarbe, T_{USA}) = 1 - 0 = 1$
$Info(Geschlecht, T_{USA}) = \frac{1}{1} * I(\frac{1}{2}, \frac{1}{2}) == 1$
$Gain(Geschlecht, T_{USA}) = 1 - 1 = 0$
Nachdem Geschlecht überhaupt keine Information hinzufügen würde unterscheidet der Baum an dieser Stelle zwischen Haarfarbe und erzeugt so die letzten zwei Blattknoten.

3.2 Entscheidungsgraph



3.2.1 Frage A

Alle Personen deutscher Herkunft werden von dem Entscheidungsbaum als Angela Merkel klassifiziert.

3.2.2 Frage B

Donald Trump wird vom Entscheidungsbaum folgendermaßen klassifiziert:

$$Geburtsort \rightarrow USA$$

$$Haarfarbe \rightarrow (Hell)BonJovi$$

3.2.3 Frage C

Da Arnold Schwarzenegger weder in USA noch in Deutschland oder Belgien geboren ist fällt er in keine der Trainierten Kategorien und der Baum kann ihn nicht klassifizieren. Bei Pipi Langstrumpf ist der Geburtsort nicht bekannt was zum selben Problem führt.

4 Regression

4.1 Mathematik

Um einen minimalen Fehler bei der Abweichung der Messwerte von der Approximation zu finden leiten wir die Funktion nach m und t ab um dann Nullstellen der Ableitung zu suchen.

$$Err(x_{1\cdots n}, t_{1\cdots n}, m, b) = \sum_{i=1}^{n}(x_i - (b + m * t_i))^2$$

$$\frac{d(m,t)}{d(m)} Err(x_{1\cdots n}, t_{1\cdots n}, m, b) = 2\sum_{i=1}^{n}(x_i - b - m * t_i) * t_i$$

$$\frac{d(m,t)}{d(t)} Err(x_{1\cdots n}, t_{1\cdots n}, m, b) = 2\sum_{i=1}^{n}(x_i - b - m * t_i)$$

Jeweils nach b beziehungsweise m aufgelöst ergibt sich:

$$2\sum_{i=1}^{n}(x_i - b - m * t_i) = 0 \ (/2)$$
$$\sum_{i=1}^{n}(x_i - b - m * t_i) = 0 \ (+\text{n*b})$$
$$\sum_{i=1}^{n}(x_i - m * t_i) = n * b \ (/\text{n})$$
$$x_{avg} - m * t_{avg} = b$$

$$2\sum_{i=1}^{n}(x_i - b - m * t_i) * t_i = 0 \ (/2)$$
$$\sum_{i=1}^{n}(x_i - b - m * t_i) * t_i = 0$$
$$\sum_{i=1}^{n}(x_i * t_i - b * t_i - m * t_i^2) = 0 \ (\text{b ersetzen})$$
$$\sum_{i=1}^{n}(x_i * t_i - (x_{avg} - m * t_{avg}) * t_i - m * t_i^2) = 0$$
$$\sum_{i=1}^{n}(x_i - x_{avg} - m * t_{avg} - m * t_i) * t_i = 0$$
$$=> \sum_{i=1}^{n}(x_i - x_{avg} - m * t_{avg} - m * t_i) = 0$$
$$\sum_{i=1}^{n}(x_i - x_{avg} - m * (t_{avg} - t_i)) = 0 \ (\text{nach m*x auflösen, durch x teilen})$$
$$m = \frac{\sum_{i=1}^{n}(x_i - x_{avg})}{-\sum_{i=1}^{n}(t_{avg} - t_i))}$$
$$=> m = \frac{\sum_{i=1}^{n}(x_i - x_{avg}) * (t_i - t_{avg})}{\sum_{i=1}^{n}((t_i - t_{avg})^2)}$$

4.2 Fragen

Wir entschieden uns die Lineare Regression jeweils über zwei einzelne Jahre zu legen, ein Jahr von dem wir einen großen Fehler erwarten und eines bei dem die Regression bessere Ergebnisse liefert. Die Formeln von oben wurden jeweils in einem Spreadsheet umgesetzt wodurch sehr viele Zwischenergebnisse sichtbar sind. Dabei wird bei der Gleichung für das m der Teil oberhalb vom Bruchstrich separat ausgerechnet vom unteren Teil. Die jeweiligen Ergebnisse sieht mann bei *Sum* welche dann das m ergeben. Als Daten wurden die Jahre 1996 und 1999 verwendet.
Bei dem Jahr 1996 ergab sich eine eine negative Steigung für die Regression-

sgerad um fast 2 Einheiten pro Monat. Auch zu erkennen war wie $x - x_{avg}$ zunehmen stärker negative Werte annahm wodurch sich die fallende Regressionsgerade erklärt. Bei diesem Jahr ist jedoch der quadratische Fehler mit über 500 recht hoch.
Beim Jahr 1999 ergab sich für die Regressionsgerade eine sehr schwache Steigung aber auch ein viel kleinerer Fehler. In diesem Jahr verlaufen die Daten fast konstant entlang einer horizontalen Linie mit dem y-Offset 110.
Für den gesamten Datensatz erscheint eine Regression eher weniger sinnvoll da man den Daten einen wellenartigen Verlauf ansieht und sich ein sehr großer quadratischer Fehler ergeben würde bei einer sehr schwachen Steigung/Gefälle der Regressionsgerade.