# Praktikum Data Mining

**Energieverbrauch und CO2-Emmisionen**

**Vorhersage und Clustering auf Finanzdaten**

Oliver Fesseler Maria Florusß Stefan Seibert

Daniel Grießhaber

21. Mai 2014

# Energieverbrauch und $CO_2$-Emmission

## Datenverwaltung und Statistik

### Einlesen der Daten, Hinzufügen der GPS Koordinaten, Abspeichern in neuer Datei

Bei der Umsetzung der Aufgaben haben wir mit verschiedenen Darstellungsformen der Daten experimentiert. Im ersten Plot fallen vor allem die Vielverbraucher einzelner Energieformen auf. Beim zweiten Plot können die verschiedenen Energiemixe pro Land direkt miteinander verglichen werden, da sie alle im selben Maßstab nebeneinander dargestellt werden.

**Ausgehend von der implementierten Visualisierung des Energieverbrauchs der Länder: Nennen Sie die 3 Ihrer Meinung nach interessantesten Beobachtungen.**

1. Durch die wenigen Industrieländer mit signifikant höherem Energieverbrauch, wie China oder die USA, wird der Plot so verzerrt, dass die Länder mit durchschnittlichem Verbrauch im Plot so gestaucht werden, dass sie kaum zu erkennen sind. Dies könnte behoben werden, wenn die Daten mit den Einwohnerzahlen aller Länder normalisiert werden würden. So könnte der Pro-Kopf-Verbrauch berechnet werden, was einen besseren Vergleich der einzelnen Länder bietet.

2. Dieses Prinzip wird klar bei der Betrachtung von China und Indien, die von der Einwohnerzahl her vergleichbar sind (China 1,3 Mrd., Indien 1,2 Mrd.[1]). China zeigt einen weitaus höheren Verbrauch als Indien an den in beiden Ländern häufigen Energieformen Kohle und Öl. Zusätzlich wären noch andere Normalisierungsfaktoren interessant:

    - Bruttoinlandsprodukt
    - Außenhandelsstatistik oder Export der Länder in US$

[1] Stand 2012, Quelle: Wikipedia

- Technologieindex

3. Die zwei Länder mit dem höchsten Energieverbrauch sind China und die USA. Dies fällt bei der Betrachtung des Gesamtenenergieverbrauchs auf. Bei der Betrachtung des Verbrauchs einzelner Energieformen wirkt es als sei China durch seinen hohen Kohleverbrauch weit vor den USA.

### Abgabe: Relevante Dateien

- `energy_consumption_per_country.py` und `energy_consumption_per_country_V2.py`
  - Implementierung Aufgabe 2.1.2: 1) - 2)
- `energy_consumption_per_country.pdf`
  - Ausgabe des Scripts `energy_consumption_per_country.py`
- `energy_consumption_per_country_V2.pdf`
  - Ausgabe des Scripts `energy_consumption_per_country_V2.py`
- `appendGeoCoordinates.py`
  - Implementierung Aufgabe 2.1.2: 3) - 5)
- `EnergyMixGeo.csv`
  - Ausgabe des Scripts `appendGeoCoordinates.py`

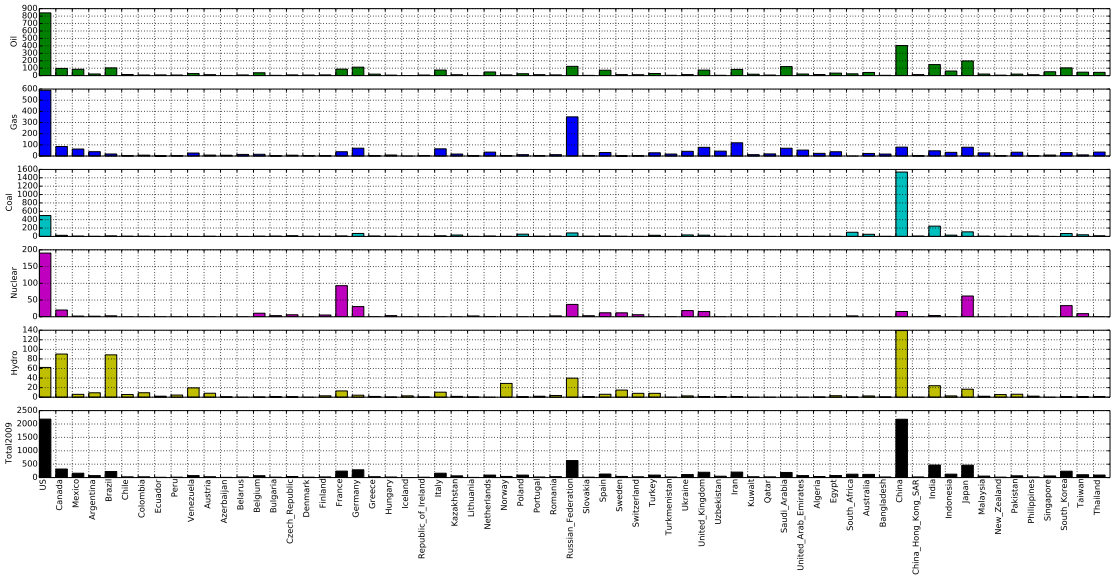



Abbildung 1: `energy_consumption_per_country.pdf`:

Ausgabe des Scripts `energy_consumption_per_country.py`

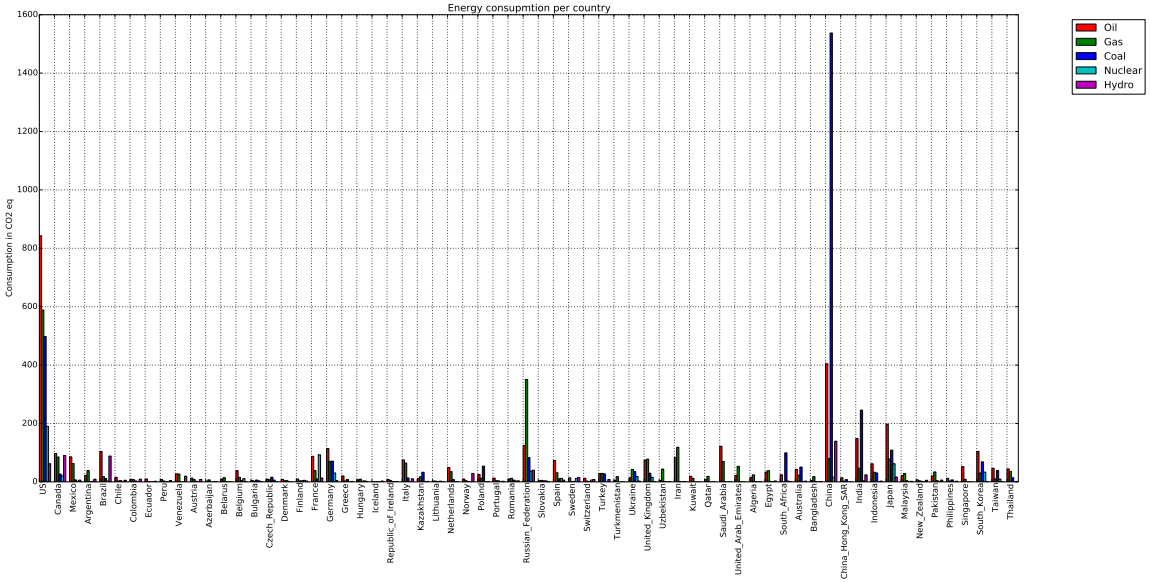
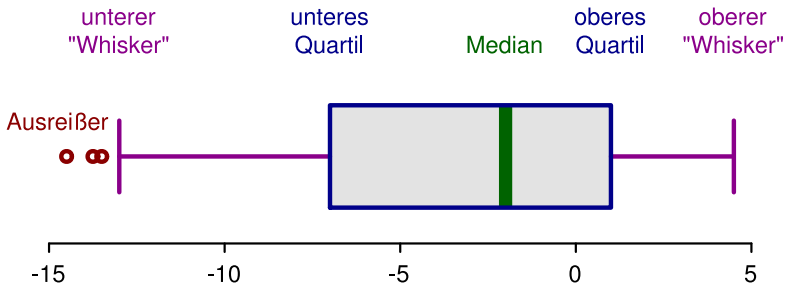



Abbildung 2: `energy_consumption_per_country_V2.pdf`:

Ausgabe des Scripts `energy_consumption_per_country_V2.py`

## Statistik der Daten

**Erklären Sie sämtliche Elemente eines Boxplot (allgemein).**




- Ausreißer ("Outliers"), in Python mit 'sym' zu setzen.
  Daten die außerhalb der Whisker liegen und somit als Ausreißer deklariert werden.
- Whisker, Länge in Python mit 'whis' zu setzen.
  Standartmäßig die 1,5-fache Länge des entsprechenden Quartils.
- Quartil
  Die Quartile sind Bestandteile der Box, welche 50 % aller Daten enthält. Dabei enthält das obere Quartil, die 25%, die über dem Median, das untere Quartil, die 25%, die unter dem Median liegen.
- Median
  Der mittlere Wert (nicht Mittelwert oder Durchschnitt), der aus dem gesamten Datensatz ermittelt wird. Er teilt den Boxplot in zwei Hälften, die wiederum jeweils in Whisker und Quartil unterteilt werden.

**Diskutieren Sie die im Boxplot angezeigte Statistik der Energieverbrauchdaten.**

- Durch den signifikant höheren Verbrauch von China und den USA wird beim Anzeigen der Ausreißer der restliche Boxplot soweit gestaucht, dass vernünftigen Daten mehr abgelesen werden können. Um die Verteilung innerhalb der verschiedenen Boxplots besser vergleichen zu können, entschlossen wir uns außerdem jede Energieform in einem einzelnen Subplot darzustellen.

- Im Gesamtplot lassen sich die einzelnen Energieformen gut miteinander vergleichen, in den einzelnen Plots können die Verteilungen innerhalb der Plots besser dargestellt werden.
- Beim Boxplot der nuklearen Energieform fällt auf, dass der Median samt unterem Quartil und unterem Whisker auf 0 liegt. Dies liegt daran, dass weniger als die Hälfte aller Länder nukleare Energie verwenden.
- Im Gesamtplot kann man erkennen, dass Öl die einzige Energieform ist, deren unterer Whisker nicht auf 0 liegt. Also verwendet jedes in Betracht gezogene Land Öl. Jede andere Energieform wird von mindestens einem Land nicht verwendet.

**Abgabe: Relevante Dateien**

- `enegryStatistics.py`
  - Implementierung Aufgabe 2.1.3: 1) - 3)
- `energyconsumption_by_energyform_in_seperate_subboxplots.pdf` und `energyconsumption_by_energyform_in_one_plot.pdf`
  - Ausgabe des Scripts `enegryStatistics.py`

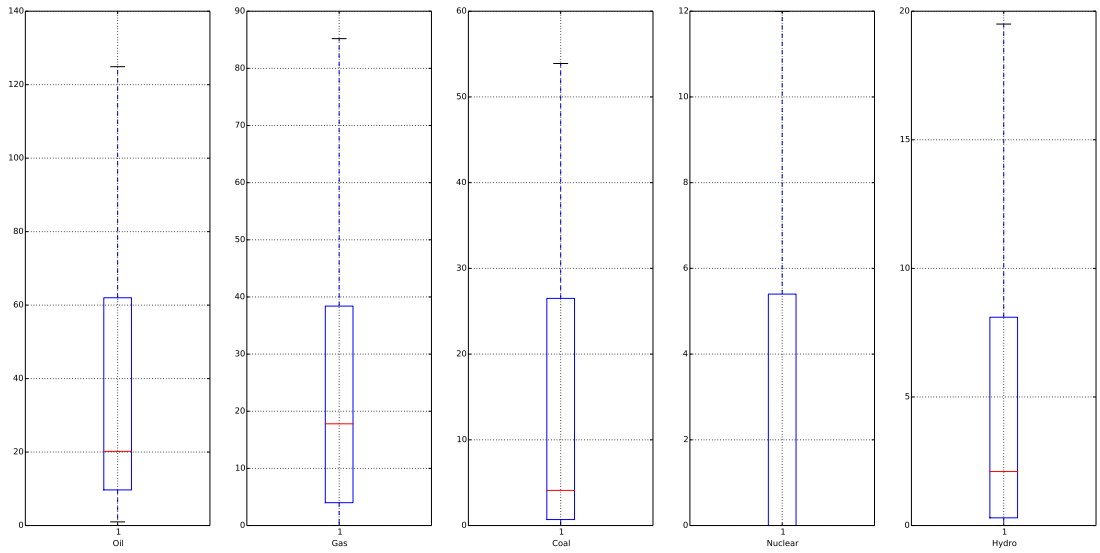



Abbildung 3: `energyconsumption_by_energyform_in_seperate_subboxplots.pdf`:
Ausgabe des Scripts `enegryStatistics.py`

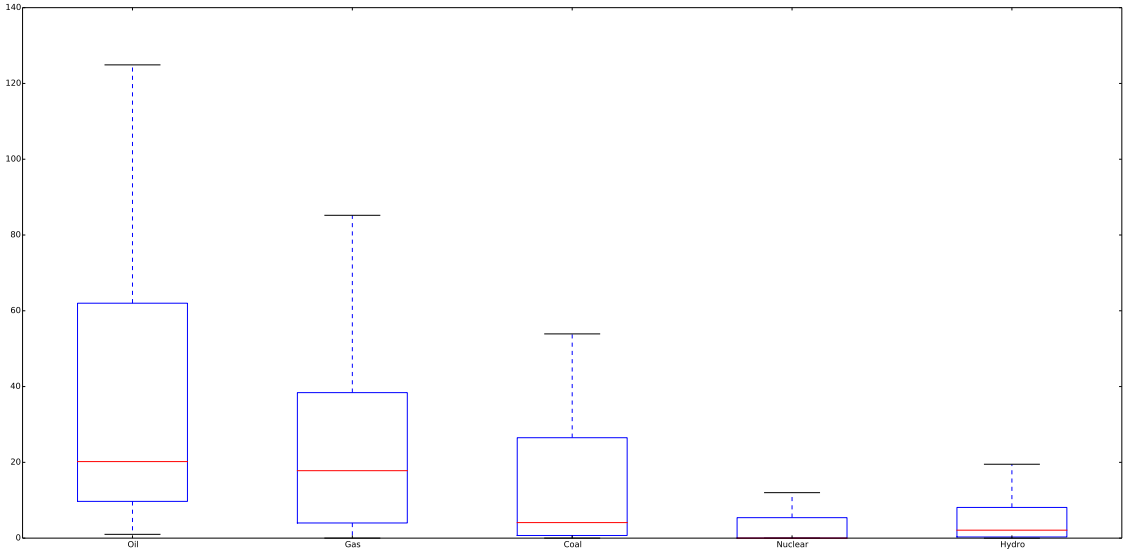



Abbildung 4: `energyconsumption_by_energyform_in_one_plot.pdf`:
Ausgabe des Scripts `enegryStatistics.py`

# Anwendung von Verfahren des unüberwachten Lernens auf Energieverbrauchsdaten

## Hierarchisches Clustering

**Was wird beim Standardisieren gemacht? Welcher Effekt könnte ohne Standardisieren beim Clustering eintreten (insbesondere wenn die euklidische Metrik verwendet wird)?**

Ohne Standardisieren ist die Ähnlichkeit des Gesamtverbrauchs ausschlaggebender als die Ähnlichkeit des Energiemixes. So werden eher Länder mit niedrigem Gesamtverbrauch gruppiert, als sie anhand ihres Energiemixes zu clustern. Dies kommt daher, dass die euklidische Metrik die geometrische Distanz zwischen zwei Punkten im Mehrdimensionalen Raum berücksichtigt. Zeigen zwei Vektoren in dieselbe Richtung, sind sich demnach vom Energiemix sehr ähnlich, haben aber unterschiedliche Längen, also einen unterschiedlich hohen Energieverbrauch, dann haben sie auch eine hohe euklidische Distanz und werden nicht demselben Cluster zugeordnet. Durch das Standardisieren werden die Längen der Vektoren normiert und so ein Vergleich erst möglich.

**Erkären Sie die beim hierarchischen Clustering einstellbaren Parameter `linkage-method` und `metric`. Welche Metrik ist Ihrer Meinung nach für diese Anwendung geeignet? Warum?**

Über die `linkage-method` wird festgelegt, wie die Cluster hierarchisch angeordnet werden. Dabei kann zwischen verschiedenen Methoden gewählt werden: Beispielsweise kann die mittlere (average), die kleinste (single) oder die größte (complete) Distanz zweier Punkte aus beiden Clustern, oder die Distanz der beiden Clusterschwerpunkte (weighted), gemessen werden. Die `metric` bestimmt die Ähnlichkeit zwischen zwei Punkten. Hierbei kann zwischen verschiedenen Metriken gewählt werden, wobei in unserem Fall Ähnlichkeitsmaße für boolsche Werte vernachlässigt werden können. Die Ähnlichkeit kann über die euklidische Distanz gemessen werden, allerdings wird hier nur die geometrische Distanz gewertet und nicht die Richtung. Weiterhin kann Cosinus-Distanz angewandt werden, bei der die Richtung der Vektoren mehr ins Gewicht fällt als die

Länge. Bei der Pearson-Ähnlichkeit wird zusätzlich die Durchschnittslänge miteinbezogen und mit der tatsächlichen Länge verrechnet. Dadurch erweist sich dieser Algorithmus als am besten geeignet.

**Welches Land ist bezüglich des Verbrauchs der hier betrachteten Energiequellen Deutschland am ähnlichsten, wenn für die linkage-method `average` und die Metrik `correlation` konfiguriert wird?**

**Antwort:** Belgien

**Charakterisieren Sie die 4 Cluster. Was ist typisch für die jeweiligen Cluster?**

- Cluster 0 ist einigermaßen gleichverteilt, wobei Wasserkraft den geringsten Anteil ausmacht. Im Vergleich zu den anderen Clustern ist der Anteil an aus Atomkraft gewonnener Energie sehr hoch.
- Cluster 1 zeichnet einen großen Verbrauch an fossilen Brennstoffen aus.
- Cluster 2 ist das kleinste Cluster und unterscheidet sich vor allem durch seinen hohen Kohleverbrauch von den anderen Clustern.
- Cluster 3 verzeichnet im Gegensatz zu allen anderen Clustern einen relativ hohen Wasserkraftanteil.

Fasst man den gesamten Energieverbrauch jedes jeweiligen Clusters zusammen (`individual_clusters_total.pdf`), erkennt man, dass die Charakteristik des Gesamtverbrauchs eines Clusters sehr stark von einzelnen Ländern abhängt. So diktieren China und die USA den Gesamtverbrauch in ihrem Cluster. Gleichzeitig zeigt sich allerdings auch die Tendenz des gesamten Clusters.

**Abgabe: Relevante Dateien**

- `energyClustering.py`
  - Implementierung Aufgabe 2.2.1: 1) - 5)
- `dendrogram.pdf`, `individual_clusters.pdf` und `individual_clusters_total.pdf`
  - Ausgabe des Scripts `energyClustering.py`

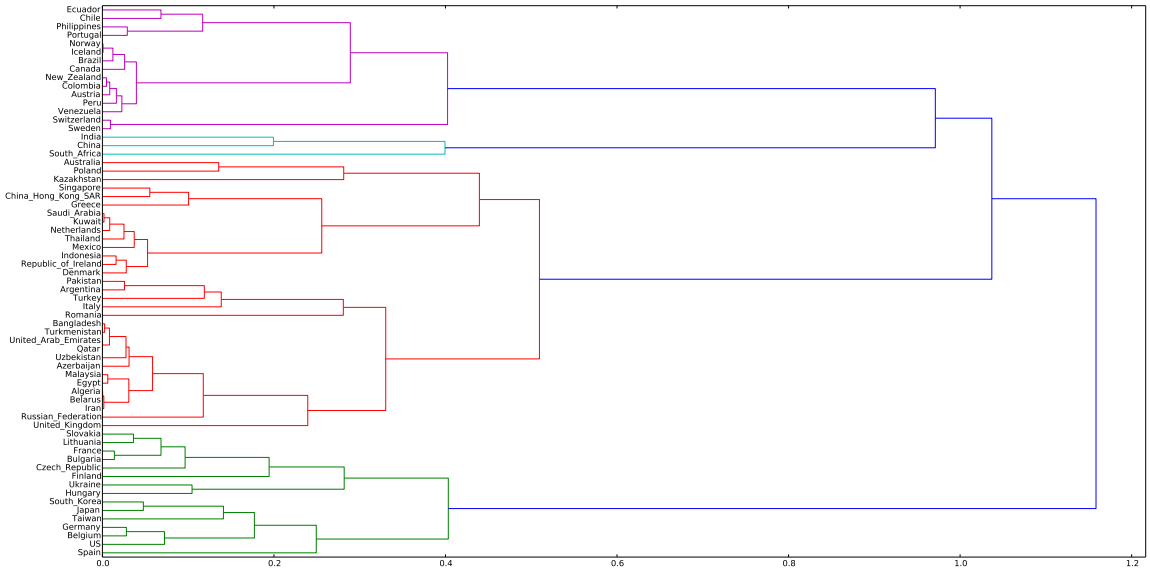



Abbildung 5: `dendrogram.pdf`:

Ausgabe des Scripts `energyClustering.py`

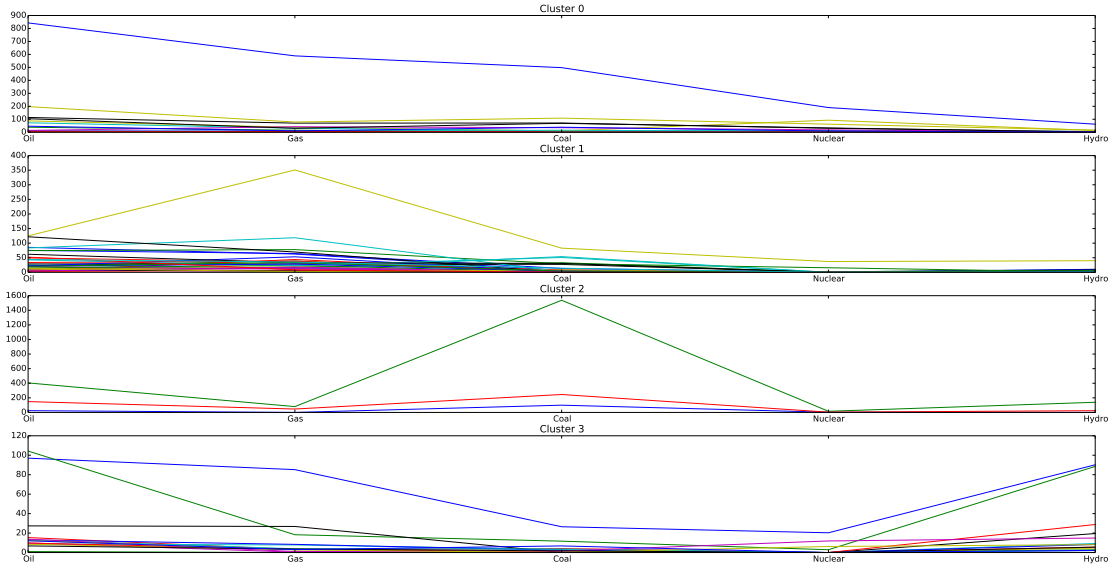



Abbildung 6: `individual_clusters.pdf`:

Ausgabe des Scripts `individual_clusters.py`

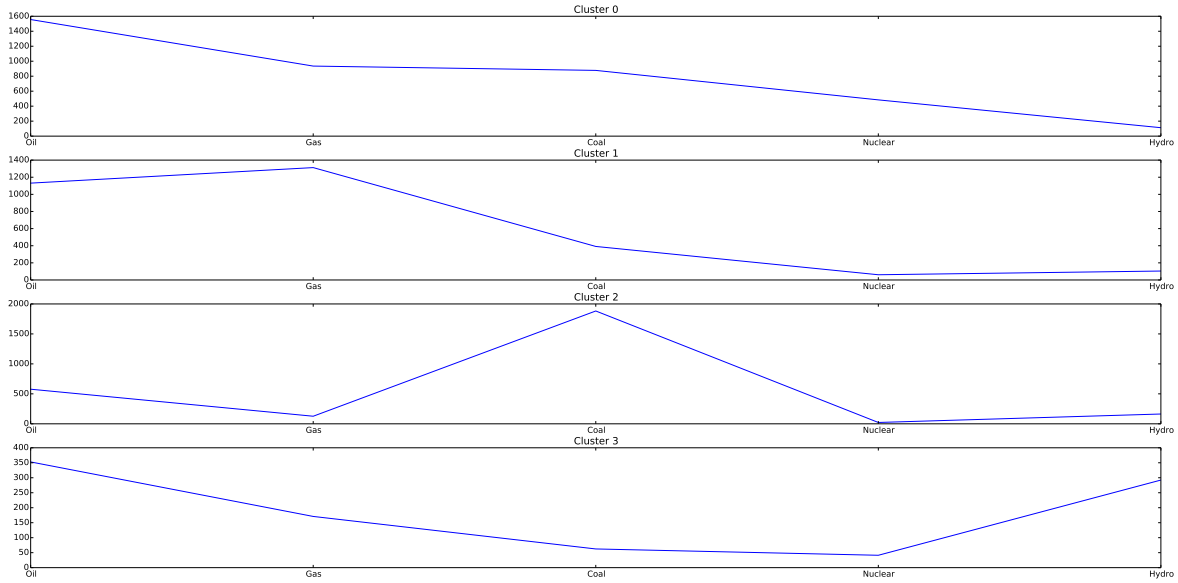



Abbildung 7: `individual_clusters_total.pdf`:
Ausgabe des Scripts `individual_clusters_total.py`

## Dimensionalitätsreduktion

**Welches Land ist nach dieser Darstellung Deutschland am ähnlichsten?**

**Antwort:** Südkorea

**Warum entspricht die hier dargestellte Ähnlichkeit nicht der im oben erzeugten Dendrogram?**

1. Da wir die Darstellung selbst optisch interpretiert und die Distanz zwischen den Punkten als Ähnlichkeitsmaßverwendet haben, haben wir die Ähnlichkeit nach der euklidischen Metrik bestimmt.

2. Auch im Dendrogramm waren sich Deutschland und Südkorea relativ ähnlich. Durch die Reduktion der Dimensionen von fünf auf zwei gehen zwangsweise Informationen verloren.

**Abgabe: Relevante Dateien**

- `energyReduceDim.py` - Implementierung Aufgabe 2.2.2: 1) - 3)
- `energyReduceDim_total.pdf`- Ausgabe des Scripts `energyReduceDim.py`

- `energyReduceDim_section.`- Ausschnitt aus `energyReduceDim_total.pdf`

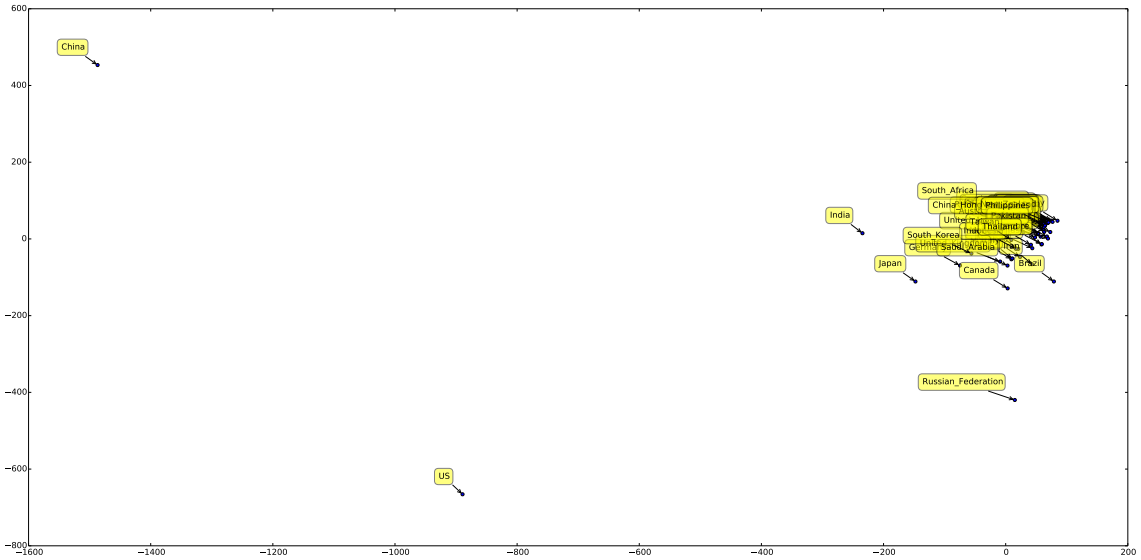



Abbildung 8: `energyReduceDim_total.pdf`:
Ausgabe des Scripts `energyReduceDim.py`

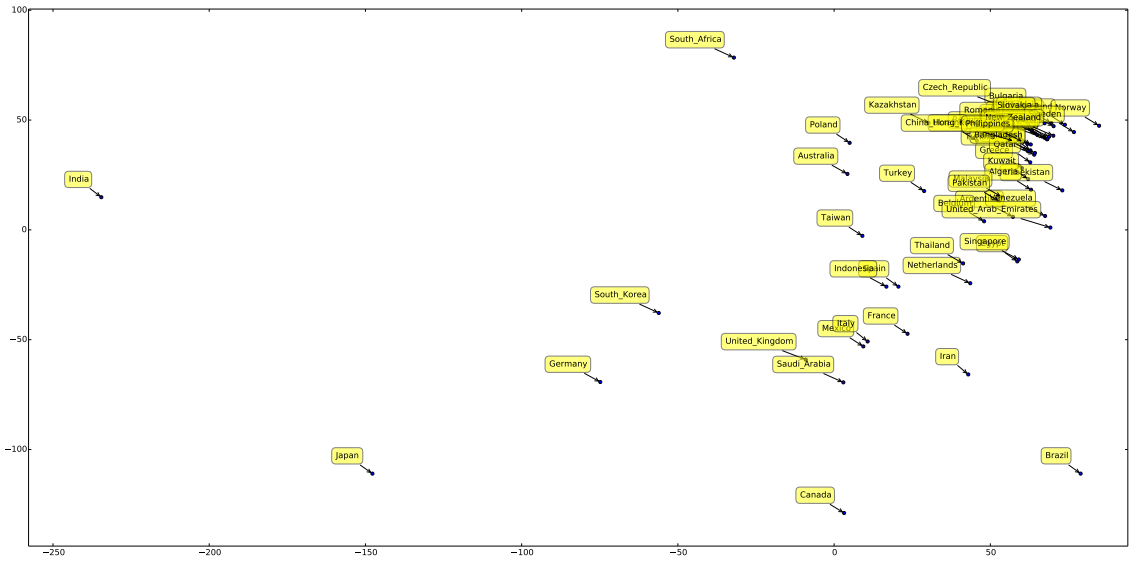



Abbildung 9: `energyReduceDim_section.pdf`:
Ausschnitt aus `energyReduceDim_total.pdf`

# Überwachtes Lernen: Schätzung der $CO_2$-Emmission

## Feature Selection

**Welche 3 Merkmale haben den stärksten Einfluß auf das Ausgabemerkmal $CO_2$-Emmission? Wie groß sind die vom Programm ausgegebenen Scores?**

| | |
|---|---|
| Kohle | 378.266881 |
| Öl | 220.010151 |
| Wasser | 79.045401 |
| Gas | 46.002230 |
| Nuklear | 34.572086 |

**Abgabe: Relevante Dateien**

- `energyFeatureSelection.py` - Implementierung Aufgabe 2.3.1: 1) - 3)

## Regression mit Epsilon-SVR

**Optimieren Sie die SVR-Parameter C und Epsilon so dass der Score in der Kreuzvalidierung minimal wird. Welche Werte für C und Epsilon liefern das beste Ergebnis?**

Die besten Ergebnisse erhielten wir für C = 0.01 und $\varepsilon = 0.001$

**Für das SVR-Objekt können die Koeffzienten der linearen Abbildung, welche durch die trainierte SVR realisiert wird, ausgegeben werden:** `meineSVR.coef_`**. Notieren Sie diese Koeffzienten für die beste SVR.**

| Öl | Gas | Kohle | Nuklear | Hydro |
|---|---|---|---|---|
| $-3.0690410$ | $-2.3485549$ | $-3.9608432$ | $4.1970815e-04$ | $4.1138445e-04$ |

**Welchen Aufschluss geben diese Koeffzienten über den Einfluss der einzelnen Eingangsmerkmale auf das Ausgangsmerkmal?**

Die Koeffizienten geben an, wie sehr die entsprechende Energieform Einfluss auf die $CO_2$-Emmission hat. Öl, Kohle und Gas haben demnach einen sehr viel größeren Einfluss als Energie aus Atom- und Wasserkraft.

**Wie groß ist die mittlere absolute Differenz zwischen Soll- und Ist-Ausgabe für die beste SVR? Diskutieren Sie dieses Ergebnis.**

Für die optimierten Parameter C = 0.01 und $\varepsilon = 0.001$ ergibt sich ein Mean Absolute Error (MAE) von 0.119259989310.

| C | $\varepsilon$ | MAE |
|---|---|---|
| 1 | 0.01 | 0.119938469138 |
| 1 | 0.001 | 0.119995514827 |
| 1 | 0.0001 | 0.119986240023 |
| 1 | 0.1 | 0.124915412379 |
| 0.1 | 0.01 | 0.119776387503 |
| 0.01 | 0.001 | 0.119259989310 |
| 0.001 | 0.0001 | 64.902638179800 |

Der erhaltene Wert für den MAE ist für realistische Daten viel zu klein, was ein eindeutiger Hinweis darauf ist, dass die verwendeten Ausgangsdaten selbst mit einem ähnlichen Algorithmus berechnet wurden. Im Diagramm (`energyPrediction.`) kann deshalb zwischen vorhergesagter und tatsächlicher Ausgabe nicht unterschieden werden, da beide Kurven genau übereinander liegen.

**Abgabe: Relevante Dateien**

- `energyPrediction.py` - Implementierung Aufgabe 2.3.2: 1) - 7)
- `energyPrediction.pdf` - Ausgabe des Scripts `energyPrediction.py`

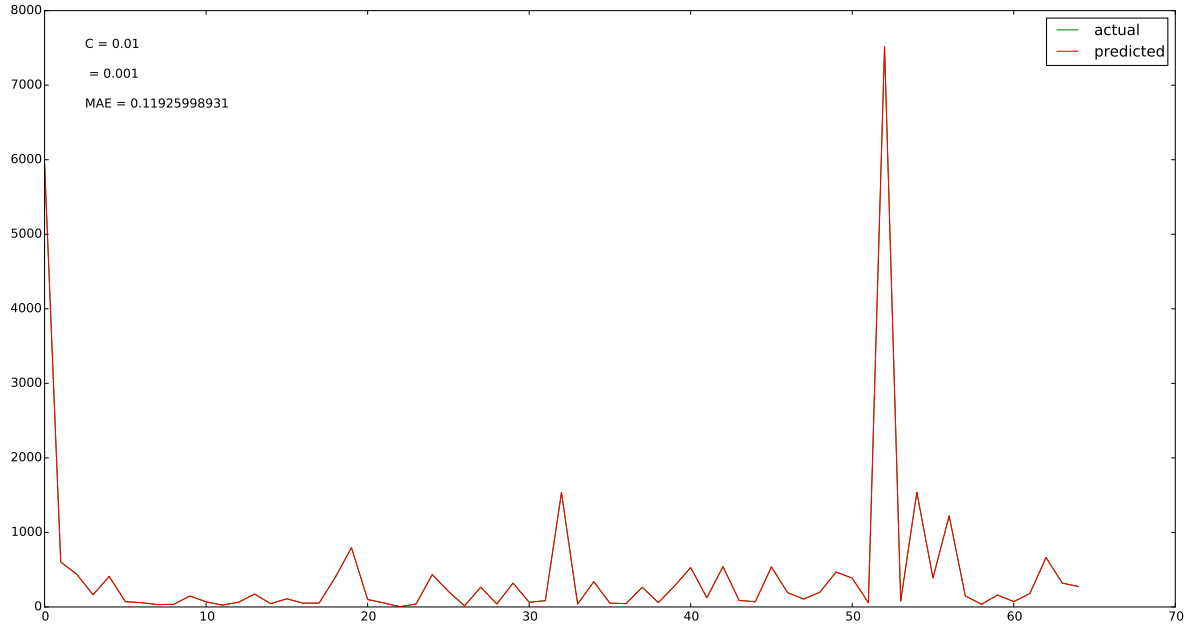



Abbildung 10: `energyPrediction.pdf`:
Ausgabe des Scripts `energyPrediction.py`

# Visualisierung des Clusterings in Google Maps

### Abgabe: Relevante Dateien

- `cluster2GoogleMaps.py` - Implementierung Aufgabe 2.4
- `clusterMap.html` - Ausgabe des Scripts `cluster2GoogleMaps.py`

# Vorhersage und Clustering auf Finanzdaten

## Vorhersage des Aktienkurses

### Datenbeschaffung

In der Implementierung von b101_stockMarketFile.py wie sie im Ergebnis vorliegt wurden die Werte von Kraft Foods (KFT) und News Corporation (NWS) nicht verwendet, da diese nicht verfügbar waren und durch einen 404-Not Found Error zum Programmabbruch führten.
In der im Fragment bereits vorhandenen Schleife wurde für jedes Symbol eine Pandas Serie angelegt, die über die Datumsangaben der Werte indiziert wird. Diese wurde anschließend als neue Spalte in ein Pandas Dataframe gespeichert, welches anschließend auch über den Funktionsaufruf to_csv als csv gespeichert werden kann.

**Abgabe: Relevante Dateien**

- `b101_stockMarketFile.py` - Implementierung Aufgabe 3.1.1
- `effectiveRates.csv` - In 3.1.1 gespeicherte Daten

### Kursvorhersage mit SVR

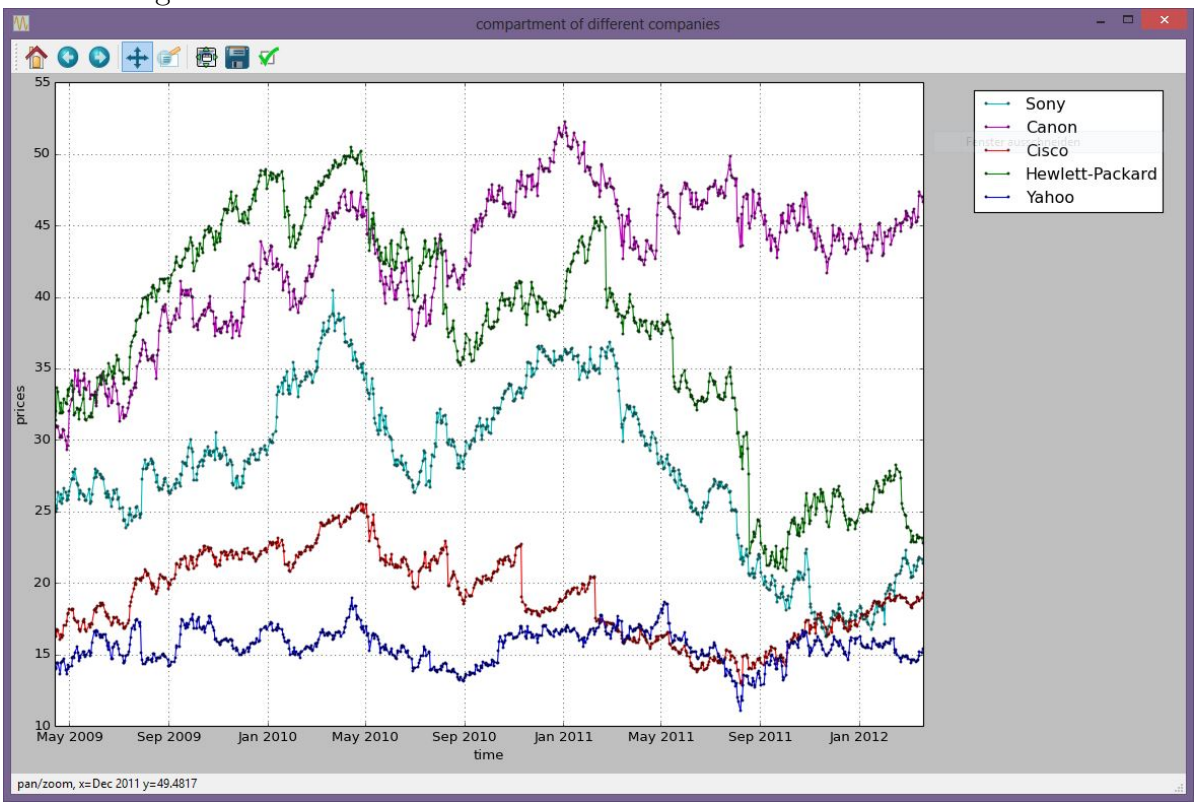
**1. Einlesen der Datei effectiveRates.csv**

Dies geschieht einfach über den Funktionsaufruf from_csv. Erfolgt dieser Aufruf auf einem leeren Pandas Dataframe liegen anschließend die Daten wieder im Speicher vor.

**2. Graphische Anzeige von 5 Aktienkursen Ihrer Wahl**

Es wurden 5 Technologie Unternehmen ausgesucht zum Vergleich. Sony, Canon, Cisco, Hewlett-Packard und Yahoo. Allen gemein ist bei größerer Betrachtung ein Aufwärtstrend, mehr oder weniger stark bis zum Anfang 2010, dann sind sinkende Kurse bis Herbst 2010 zu beobachten worauf die Kurse zum Teil relativ unterschiedlich verlaufen.

Sony und HP stürzen 2011 stark ab, die anderen Unternehmen können von lokalen Ausbrüchen abgesehen ihren Kurs in etwa halten.




### 3. Festlegen des Unternehmens

Wie vorgeschlagen wurde Yahoo (YHOO) für die Kursvorhersage ausgewählt.

### 4. Umwandlung der seriellen Daten in eine zyklische Form

Das geforderte Umwandeln der seriellen Daten in eine zyklische Form wurde durch eine getmodel Methode implementiert, der als Parameter der Time-Delay und die seriellen Daten übergeben werden können. Als Rückgabewert liefert diese daraufhin die zyklischen Daten in einem Pandas Dataframe und die Zieltage in einer Pandas Serie. Als Spaltenindex dienen hier ebenfalls die im Beispiel verwendeten Tage vor dem Zieltag, also bei 24 Lerntagen, sind die Spalten T-24 bis T-1. T-Class entspräche dann den Zieltagen in der Pandas Serie.

**5. Festlegen von Trainingszeitraum und Vorhersagezeitraum**

Alle Parameter, inklusive Trainingszeitraum und Vorhersagezeitraum werden an einer zentralen Stelle des Skripts eingestellt, so dass eine Änderung der Parameter leicht möglich ist. Es wurden folgende Variablen definiert:

1. TRAININGS_TIME - Der Trainingszeitraum
2. PREDICTION_TIME - Der Der Vorhersagezeitraum
3. TIME_DELAY - Der Time-Delay für die zyklischen Daten
4. C_VALUE - Der C-Parameter für die SVR
5. EPSILON_VALUE - Der Epsilon-Parameter für die SVR

**6. Anlegen der SVR und Vorhersagenberechnung**

Die SVR wurde durch den Funktionsaufruf svm.SVR angelegt. Anschließend wurde die fit Methode aufgerufen um die SVR mit den Trainingsdaten zu trainieren.

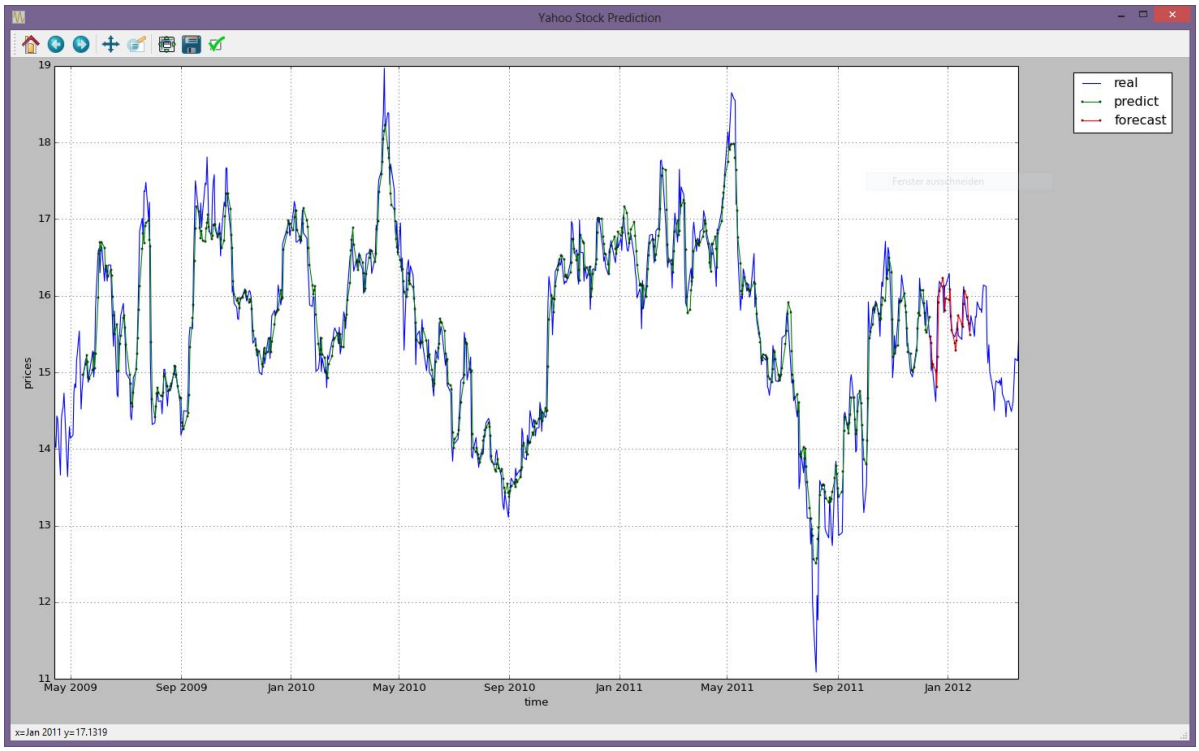
Im zweiten Schritt wurde die SVR angewiesen eine Vorhersage mit den Trainingsdaten auszuführen, um diese zum Vergleich mit in die Plotausgabe einfließen lassen zu können. Danach wurden die echten Vorhersagewerte berechnet und diese in eine Pandas Serie überführt.

**7. Berechnung und Vergleich des MAE**

Der Mean Absolute Error wird durch die bereits in Versuchsteil 1 verwendete getmae() Methode berechnet. In der folgenden Tabelle werden die MAE Werte für verschiedene Parameterkonfigurationen aufgelistet

| C | $\varepsilon$ | Time Delay | MAE |
|---|---|---|---|
| 500 | 0.6 | 24 | 0.312024245957 |
| 50 | 0.6 | 24 | 0.312024245957 |
| 5 | 0.6 | 24 | 0.305131498375 |
| .5 | 0.6 | 24 | 0.223760696196 |
| .05 | 0.6 | 24 | 0.341643714938 |
| .5 | 0.006 | 24 | 0.206443426583 |
| .5 | 0.06 | 24 | 0.206312883609 |
| .5 | 6 | 24 | 0.648 |
| .5 | 0.06 | 3 | 0.248084638701 |
| .5 | 0.06 | 6 | 0.247563624714 |
| .5 | 0.06 | 12 | 0.258172037051 |
| .5 | 0.06 | 24 | 0.206312883609 |
| .5 | 0.06 | 48 | 0.262666322932 |

**8. Darstellung der Kurswerte im zeitlichen Verlauf**




## Beantwortung der Fragen zu Kapitel 3.1 und Relevante Dateien

### Überlegung zum notwendigen Aufbau der Datenvektoren des Vorhersagezeitraums

**Antwort:** Die Vorhersage sollte immer nur für einen Tag durchgeführt werden, da hier die präzisesten Ergebnisse möglich sind. Anschließend wird der vorhergesagt Wert in die Datenvektoren übernommen, wodurch für den nächsten Tag die Datenvektoren bei einem Time-Delay von 24, die Datenvektoren aus 23 bekannten Werten und 1 vorhergesagten Wert entsprechen. Selbes Verfahren wird immer weiter fortgeschrieben, bis nach 24 Tagen die Vorhersagewerte nur noch auf Basis ebenfalls vorhergesagter Werte berechnet werden können.

**Für welche Werte von Time Delay, C und Epsilon wird die beste Vorhersage erreicht?**

**Antwort:** Für die Beantwortung dieser Frage kann man sich am MAE und der zugehörigen Tabelle aus Aufgabe 8 dieses Kapitels orientieren. Die besten Werte erreicht man bei einem möglichst niedrigen MAE, hier wurde ein Minimum von 0.206312883609 bei folgenden Werten gefunden: Time Delay = 24, C = 0.5, Epsilon = 0.06.

Interessant ist hierbei dass auch ein sehr niedriger Time Delay von 3 bereits relativ gute Vorhersagen bringt. Ein sehr hohes Epsilon sorgt für derart starkes Glätten der Vorhersagewerte dass keinerlei Wertveränderungen mehr sichtbar sind. Ein sehr hoher C Wert dagegen sorgt für zu starke Ausschläge in den Kursvorhersagen. Allgemein scheint der MAE verhältnismäßig niedrig für eine Vorhersage zu sein. Die Eingabedaten wurden jedoch überprüft und sind korrekt.

**Abgabe: Relevante Dateien**

- `b102_stockMarketPrediction.py` - Implementierung aller Aufgaben aus 3.1.2
- `stockPredict.pdf`- Der Plot mit den berechneten Vorhersagewerten bei minimalem MAE

# Clustering der Aktienkursverläufe

Aufgrund der zwei nicht verfügbaren Aktienkurse reduziert sich die Menge der betrachteten Firmen auf 59.

**1. Berechnung der Differenz**

Für die Differenzberechnung wurde ein Pandas Dataframe angelegt, dessen Spalten den Firmensymbolen entsprechen. Jede Reihe wiederum entspricht einem Datum und der Wert der Differenz aus Open und Close.

**2. Berechnung der Ähnlichkeitsmatrix**

Die Ähnlichkeitsmatrix wird durch die gegebene Funktion corrcoef() berechnet. Zu beachten war hier dass der Funktion in unserem Fall die transponierte Ähnlichkeitsma-

trix übergeben werden mußte, da sonst die Relationen zwischen den Tagen und nicht den Unternehmen zurückgeliefert wurde.

### 3. Berechnung der AffinityPropagation

Im dritten Schritt wurde ein entsprechend vorgegebenes AffinityPropagation Objekt mit dem Parameter pprecomputedërzeugt, um somit die Matrix übergeben zu können, ansonsten würde diese vom Objekt selbst berechnet. Anschließend wurde die Anzahl der gefundenen Cluster abgefragt.

### 4. Plot der zusammengehörenden Kurse

Die AffinityPropagation Klasse liefert eine Liste von 59 Werten, wobei jeder Wert der Clusternummer eines jeden Unternehmens steht. Über diese konnte iteriert werden um das Unternehmen dem entsprechenden Plot zuzuweisen und diese gemeinsam anzeigen zu können.

**Auszugsweise Darstellung des Clusters 10**

**Analyse der Clusterzuweisungen**

**Antwort:** Die Cluster wurden zuerst auf Zugehörigkeit zu Branchen untersucht:

1. Cluster 0 umfasst die Lebensmittelfirmen Coca-Cola und Pepsi sowie Kellogs.
2. Cluster 1 fasst die beiden Drogerieketten Walgreen und CVS zusammen.
3. Cluster 2 beinhaltet die drei großen Konzerne Apple, Amazon und Yahoo.
4. Cluster 3 fasst drei Pharmaziekonzerne zusammen.
5. Cluster 4 wird aus drei Multimediakonzernen gebildet.
6. Cluster 5 besteht ausschließlich aus Energiekonzernen.
7. Cluster 6 ist ein Mix aus Produzenten aus Motorfabriken, Konsumgüterherstellern und Büroartikelproduzenten.
8. Cluster 7 beinhaltet die Hersteller von Drogerieartikeln und Pflegeprodukten.
9. Cluster 8 ist ein Mix aus Banken, Warenhausketten und Fahrzeugherstellern.
10. Cluster 9 ist das Cluster der IT Technologiekonzerne wie IBM, Cisco, SAP, etc.
11. Cluster 10 beinhaltet Konzerne die unter anderem in der Rüstungsindustrie aktiv sind.

Insgesamt ergibt sich ein relativ gutes Clustering nach Branche, allein aufgrund der gemeinsamen Kursschwankungen innerhalb der Branchen. Cluster 6 und 8 enthalten jedoch verschiedene Märkte, was vermutlich an deren nicht branchenüblichen Kursverläufen liegt. Wenn die Kurswerte zusätzlich direkt verglichen werden, fällt auf dass in den Clustern die immer nur eine Branche enthalten oft ähnliche Kursverläufe sichtbar sind, nicht absolut jedoch relativ bezogen auf ähnliche lokale Maxima und Minima im zeitlichen Verlauf. Die Kurswerte im Cluster 8 sind größtenteils gleichbleibend auf einem sehr niedrigen Niveau, jedoch finden sich hier auch Unternehmen mit Ausreißern. Hier ist nicht ersichtlich wie eine Zuordnung zum Cluster erfolgt. Eine ähnliche Situation zeigt sich in Cluster 6. Insgesamt ist der größere Teil der Kurse ähnlich in seinem Verlauf, jedoch sind Ähnlichkeiten nicht direkt für das gesamte Cluster erkennbar sondern immer nur zwischen mehreren Einträgen.

### Abgabe: Relevante Dateien

- `b103_stockMarketClustering.py` - Implementierung aller Aufgaben aus 3.2