




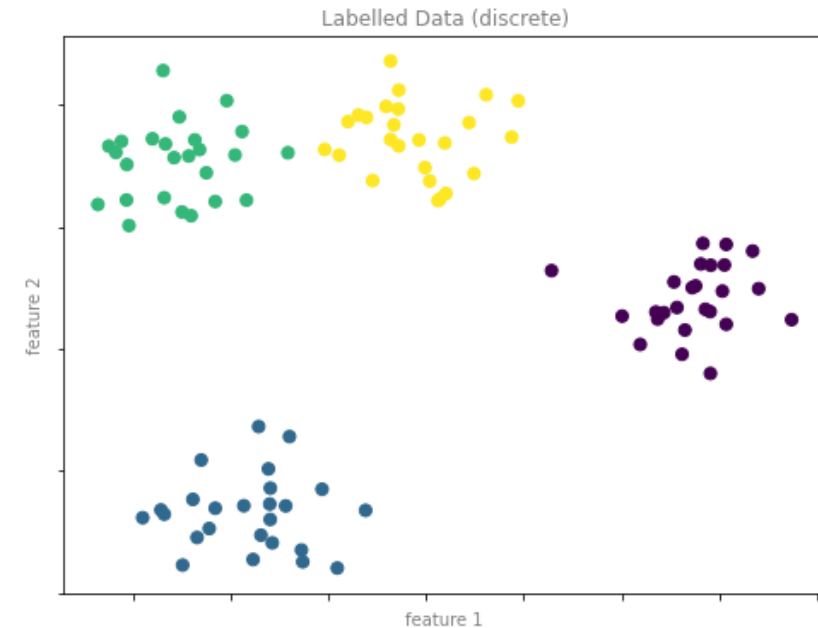
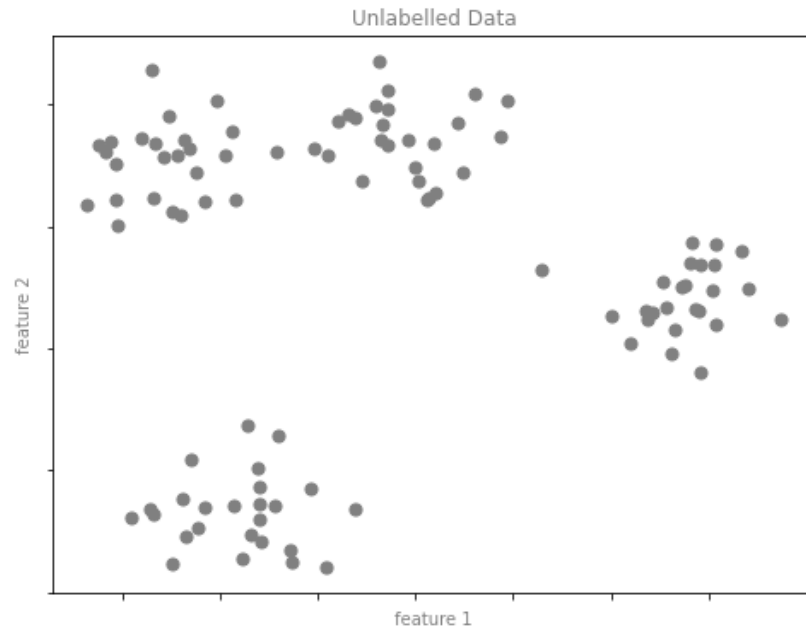
Modeling

Unsupervised Learning: Clustering

Agenda

- 
- Clustering im Allgemeinen
 - k-means Clustering: generell und in `sklearn`
 - Expectation-Maximization Algorithmus
 - k-means im Detail
 - Nachteile des EM-Algorithmus
 - Validierung von Clustering
 - Variationen k-means
 - Dichtebasiertes Clustering: DBSCAN

Was sehen Sie?

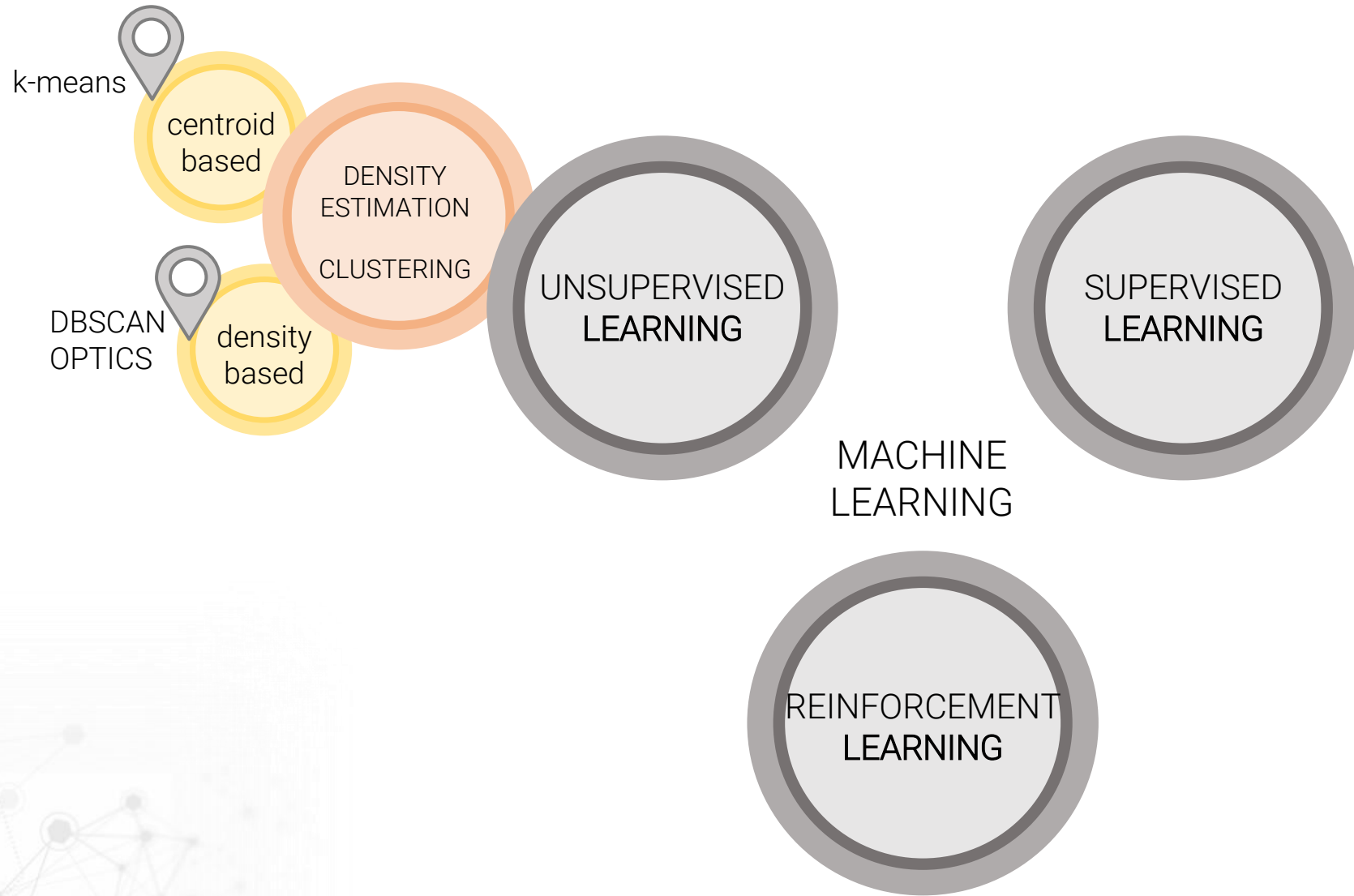


0

So what?

- Ihr visueller Apparat entdeckt automatisch und ohne Mühe Gruppierungen bzw. Struktur in Daten
- Wie bringen wir das Algorithmen bei?¹

SVM: wo befinden wir uns?



Clustering im Allgemeinen



Was denken Sie?

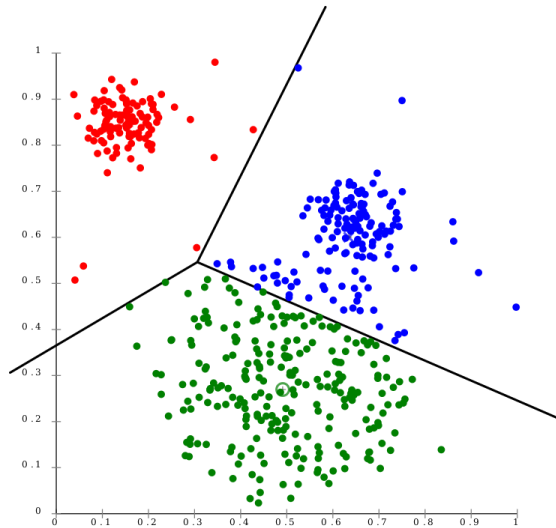
Wie würden Sie mit eigenen Worten beschreiben, was ein Clustering bewirkt bzw. durchführt?



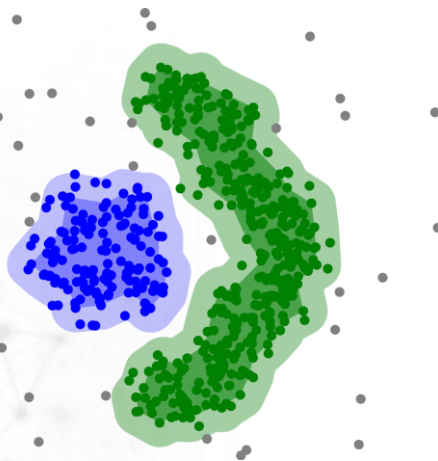
Was denken Sie?

Was heißt hier optimal?

Partitionierend



Dichtebasiert

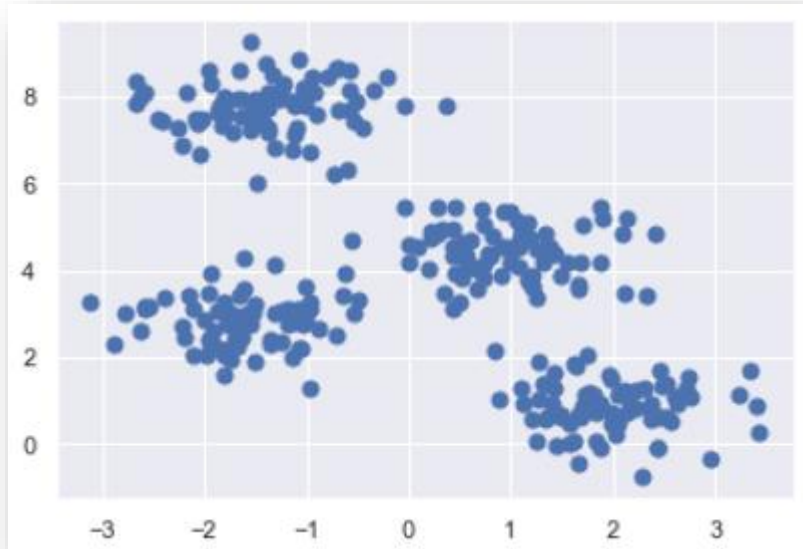


- Clustering Algorithmen versuchen aus den Eigenschaften der Daten eine **optimale Unterteilung in Gruppen** zu erreichen → diskrete Labels
- Es gibt **verschiedene Arten des Clusterings**:
 - Partitionierendes Clustering (z.B. k-means und Varianten)
 - Dichtebasiertes Clustering (z.B. DBSCAN, OPTICS)
 - Hierarchisches Clustering
 - Gitterbasierte Verfahren

k-means: im Allgemeinen

?

Was denken Sie?
Wie wird ein Cluster definiert?



```
1 X, y_true = make_blobs(n_samples=300, centers=4,  
2                       cluster_std=0.60, random_state=0)  
3 plt.scatter(X[:, 0], X[:, 1], s=50);
```

- Wir beginnen mit dem *k-means* Algorithmus
- Der k-means Algorithmus sucht nach einer **vorabbestimmten Anzahl** an Clustern in einem **nicht** gelabelten Datensatz
- **Heuristik** des k-means Algorithmus:
 - Der Cluster-Mittelpunkt (== **Centroid**) ist das arithmetische Mittel aller Punkte des jeweiligen Clusters
 - Jeder Datenpunkt ist **näher** an seinem eigenen Centroid als an anderen

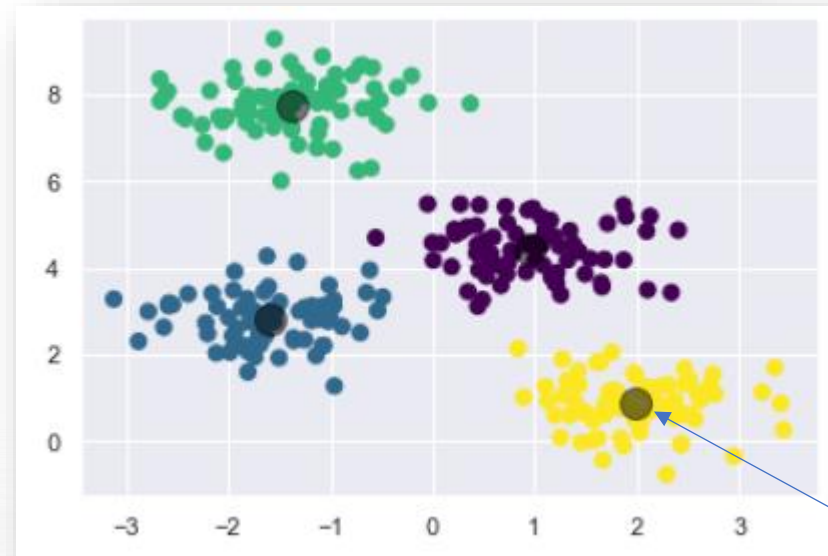
0

So what?

Unsere Leitfrage: Wie macht der k-means Algorithmus das, was wir „mit dem Auge“ mühelos können?

k-means: sklearn

```
1 from sklearn.cluster import KMeans
2 kmeans = KMeans(n_clusters=4)
3 kmeans.fit(X)
4 y_kmeans = kmeans.predict(X)
```



```
1 plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')
2
3 centers = kmeans.cluster_centers_
4 plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
```

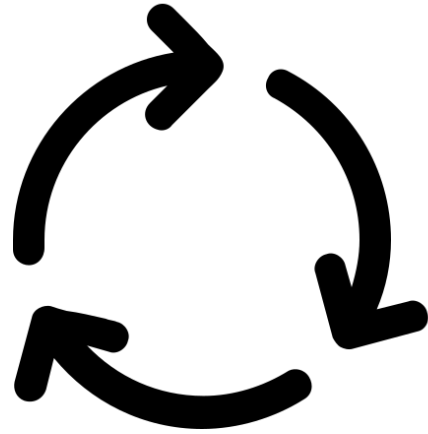
- In sklearn weist der k-means Algorithmus die gleiche Struktur wie andere Modelle auf
- Als wichtiger Zusatz bei der **Instanziierung** des Modells müssen wir die **Anzahl an Clustern** angeben
- Die **Zugehörigkeit** bzw. **Labels** der Datenpunkte (auch neuer bzw. ungesehener Datenpunkte) zu den jeweiligen Centroids erhalten wir über die `.predict()`-Methode
- Die erlernten Centroids stecken im `.cluster_centers_`-Attribut des Modells



Was denken Sie?

Wie findet der Algorithmus die Centroids ohne alle möglichen Kombinationen explizit zu evaluieren?

Einschub: Expectation-Maximization



- Die Heuristik der vorletzten Folie wird umgesetzt – ein „Rezept“ entsteht → die allgemeine Form dieses „Rezeptes“ ist der sog. **Expectation-Maximization-Algorithmus**
- Der EM-Algorithmus findet in vielen Machine Learning Modellen Anwendung
- Es handelt sich um einen **iterativen** Algorithmus
- **Generelle Idee:**
 1. Initialisiere eine **zufällige** Konfiguration des Modells
 2. Führe folgende zwei Schritte abwechselnd bis **Konvergenz** aus:
 - i. **E(xpectation)-Schritt:** Zuordnung der Daten zu den einzelnen Teilen des Modells
 - ii. **M(aximization)-Schritt:** Parameter an die neueste Zuordnung anpassen
- Konvergenz: findet keine **wesentliche** Verbesserung mehr statt → Abbruch

Einschub: Expectation-Maximization



Was denken Sie?

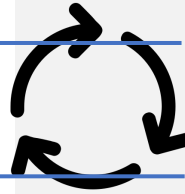
Was sind die analogen Schritte beim k-means?

Initialisiere Centroids zufällig

Weise Datenpunkte zum nächsten Centroid zu

Berechne die neuen Centroids als arithmetische Mittel der neuen Datenpunkte

- Die Heuristik der vorletzten Folie wird umgesetzt – ein „Rezept“ entsteht → die allgemeine Form dieses „Rezeptes“ ist der sog. *Expectation-Maximization-Algorithmus*
- Der EM-Algorithmus findet in vielen Machine Learning Modellen Anwendung
- Es handelt sich um einen **iterativen** Algorithmus
- **Generelle Idee:**
 1. Initialisiere eine **zufällige** Konfiguration des Modells
 2. Führe folgende zwei Schritte abwechselnd *bis Konvergenz* aus:
 - i. **E(xpectation)-Schritt:** Zuordnung der Daten zu den einzelnen Teilen des Modells
 - ii. **M(aximization)-Schritt:** Parameter an die neueste Zuordnung anpassen
- Konvergenz: findet keine **wesentliche** Verbesserung mehr statt → Abbruch



So what?

- *Expectation:* bedeutet beim k-means also „unsere Erwartung“ updaten, zu welchem Cluster die Datenpunkte gehören
- *Maximization:* ein Gütemaß wird maximiert (in unserem Fall durch die Berechnung des Mittelwertes)

k-means: Input und Output

Input

\mathbf{X} : Data matrix with dimensions $N \times T$

N : number of observations

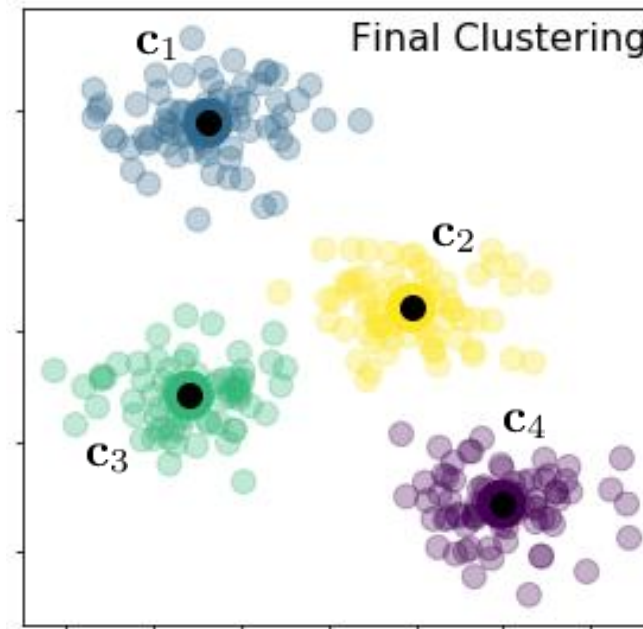
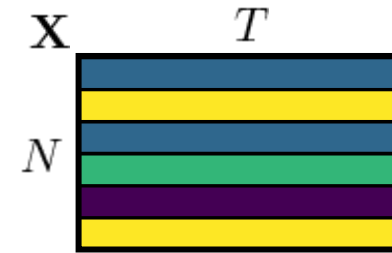
T : number of features (e.g. \bar{x} , σ^2 , ...)

k : number of clusters to be extracted

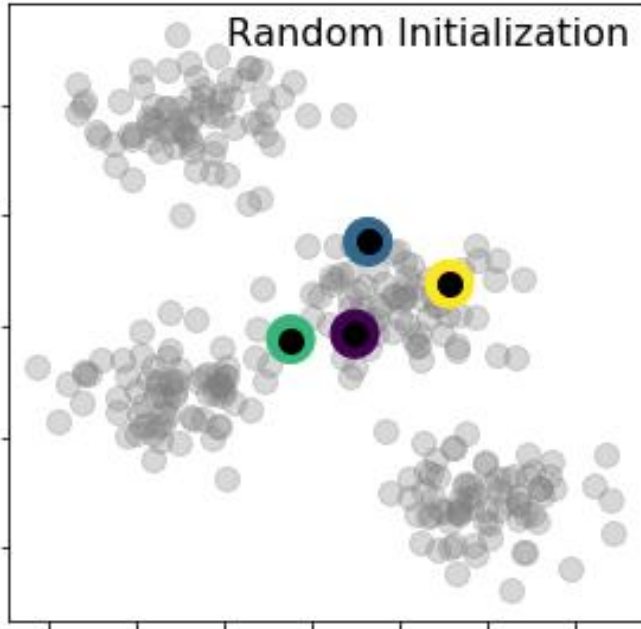
Output

\mathbf{c}_z : cluster centroids with $z = 1, \dots, k$

\hat{z}_t : assigned cluster index of data point \mathbf{x}_t



k-means: Ausführung



1. Initialization:
Initialize centroids \mathbf{c}_z
randomly

2. Expectation step:
Assign to nearest centroid

$$\hat{z}_t = \arg \min_z (d(\mathbf{c}_z, \mathbf{x}_t))$$

3. Maximization step:
Calculate new centroid

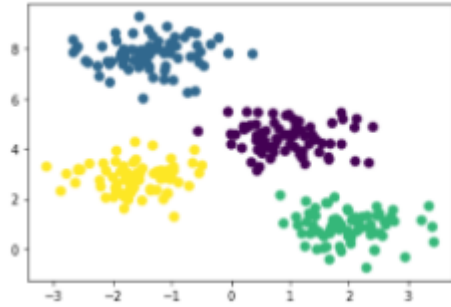
$$\mathbf{c}_z \leftarrow \frac{\sum_{t=1}^T u(t, z, \hat{z}_t) \mathbf{x}_t}{\sum_{t=1}^T u(t, z, \hat{z}_t)}$$

with

$$u(t, z, \hat{z}_t) = \begin{cases} 1, & \text{if } \hat{z}_t = z \\ 0, & \text{else.} \end{cases}$$

4. Repeat
2. and 3. until
convergence

Beispiel: k-means from Scratch



Der k-means Algorithmus ist so anschaulich, dass wir diesen Algorithmus "from Scratch" - also im Detail - nachprogrammieren können. Unsere Aufgabe in diesem Beispiel ist es also, den k-means Algorithmus Schritt für Schritt zu implementieren. Schreiben Sie also eine Funktion, die als Eingabeargumente die Daten und die Anzahl zu extrahierender Cluster aufnimmt - und uns die Centroids und Labels der Daten zurückgibt.