Praktikum 2 Clustering auf dem Million Song Dataset

SparkSQL ML zum Clustering, Tableau zur Visualisierung

Ziel des Praktikums

ist es, unterschiedliche Clusterings auf dem 10k Song Dataset und dem Million Song Dataset (1m) zu generieren, zu visualisieren und zu vergleichen.

Durchführung des Praktikums

Loggen Sie sich auf Zeppelin ein und kopieren Sie das Notebook praktikum2/_istaccount

Zum Generieren der k-Means-Modelle sollten Sie sich mit der API und den Erläuterungen zur KMeans-Klasse vertraut machen:

- https://spark.apache.org/docs/2.4.0/ml-clustering.html#k-means
- http://spark.apache.org/docs/2.4.0/api/python/pyspark.ml.html#pyspark.ml.clustering.KMeans

Als **Datenbasis** sollte die im ersten Praktikum von Ihrer Gruppe erstellte Tabelle mit den Profile-Vektoren der Lautstärkeprofile für das 10k-Song Dataset zur Verfügung stehen.

Eine entsprechende Tabelle für das 1m-Song Dataset auf der Basis von 10 Bins gleicher Breite wird von uns zur Verfügung gestellt → vgl. Aufgabe 2.

Führen Sie die gesamte Aufgabe 1 zunächst auf den Daten der 10k-Stichprobe durch!

Laden Sie zunächst die für das Clustering benötigten Prädiktoren Ihrer Tabelle z.B. mit einem `select`-Statement in eine geeignete Variable.

Aufgabe 1 – Clustering und Visualisierung auf dem 10k Song Dataset

a) Generieren Sie ein **Clustering** auf Ihren vorbereiteten Profile-Vektoren der **10k-Stichprobe** unter Verwendung der Klasse **KMeans**.

   Informieren Sie sich über die default-Werte der zu setzenden Parameter und wählen Sie ein erstes k für die Anzahl der Cluster.

b) Lassen Sie sich die **Kosten** (quadratischer Fehler) für das Clustering ausgeben.

c) Generieren Sie für weitere k erneut Modelle (auf den Profile-Vektoren der 10k-Stichprobe!) und skizzieren Sie mit Hilfe von matplotlib in einem Diagramm den Verlauf des entsprechenden Charts zur Anwendung der **Elbow-Methode** (vgl. Kapitel 2, Teil 1).

d) Wenden Sie ein aus Ihrer Sicht geeignetes Modell auf Ihre Profile-Vektoren an und speichern Sie die geclusterten Daten in einer neuen Tabelle.

   Das generierte **Modell speichern und laden** Sie wahlweise wie folgt:

```
model.write().overwrite()
.save("hdfs://141.100.62.85:9000/user/ISTACCOUNT/modelname")
newModel = KMeansModel
.load("hdfs://141.100.62.85:9000/user/ISTACCOUNT/modelname")
```


Bearbeitungsdatum: 24.04.19 (Mi6y), 25.04.19 (Do5y) und 08.05.19 (Mi6x)

e) Öffnen Sie **Tableau**. Verbinden Sie sich mit Ihrer Datenquelle auf dem Cluster, die das Ergebnis des Clusterings enthält.

f) Führen Sie in Tableau einen Join durch mit der Tabelle der zugehörigen Metadaten.

g) Visualisieren Sie zum Beispiel
   - die durchschnittlichen Werte pro Bin und Cluster,
   - die Anzahl der Titel pro Künstler, um anschließend
   - für einen ausgewählten Künstler alle Titel mit zugehöriger Cluster_Id anzeigen zu lassen.
   - Ziehen Sie die Cluster_Id im Bereich Markierungen auf "Farbe" → die Anzahl der Titel pro Künstler wird mit der Anzahl der Titel pro Cluster_Id farblich überlagert.
   - ***Hinweis zur Normierung von Balkendiagrammen auf den Bereich [0,1] (siehe auch Dokument „Tableau Introduction"):***
     Rechte Maus Spalte unten bzw. Zeile links (je nach Orientierung) → "Achse bearbeiten …", "Bereich": fixiert, "fester Anfang": 0.0, "festes Ende": 1.0

h) Führen Sie „Audio-Stichproben" als Tests für die „akustische Güte" der Clusterzugehörigkeit durch. Vergeben Sie darauf (und auf den Profilen) basierend für einige Cluster geeignete Labels.

Hinweis: Übertragung von Tableau-Visualisierungen auf neue Datenquellen s. u.

Vermeiden Sie Visualisierungen von Millionen von Datensätzen in Tableau!
Lassen Sie sich in Tableau nur Aggregation und/oder Filter der 1m-Datensätze anzeigen.

Aufgabe 2 – Clustering und Visualisierung auf dem 1m Song Dataset

Auf dem Cluster steht eine Tabelle zur Verfügung, die den Profilevektor per Track für das 1m Song Dataset enthält – generiert auf der Basis eines Binnings mit 10 Bins gleicher Breite.

Der Wertebereich von *timbre_0* beträgt hier im Vergleich zur 10k Stichprobe [0.0, 64.841]:

profilevectorpertrack_1m

Sie sollten diese bereits generierte Tabelle für das Clustering verwenden, es sei denn, Sie möchten eine andere Binning-Strategie auf *timbre_0* für das 1m Song Dataset anwenden.

Führen Sie zunächst die *Teilaufgaben a) - c) von Aufgabe 1* auf den vorbereiteten Profile-Vektoren aus dem 1 Million-Dataset zu *timbre_0* – loudness durch.

Übertragen Sie die Daten des generierten Clusterings wie folgt in Tableau:

→ **Übertragung von Tableau Visualisierungen für 10k-Datenquelle auf 1m-Datenquelle**
Wenn Sie in Tableau Visualisierungen für eine 10k-Datenquelle angelegt haben, können Sie diese einfach auf eine entsprechende 1m-Datenquelle übertragen, indem Sie im Datenquellen-Bereich die 10k Tabelle durch die 1m Tabelle ersetzen:

Hierzu ziehen Sie die 1m Tabelle genau auf die 10k Tabelle im Join-Bereich bzw. vice versa.

Tipp: Um die Ergebnisse vergleichen zu können, speichern Sie Ihr Tableau-Buch vorher ab und laden Sie es parallel in einer zweiten Tableau-Instanz.

Analysieren Sie die geänderten Grafiken auf den Arbeitsblättern aus Teilaufgabe g) von Aufgabe 1. Lassen sich Charakteristiken, die Sie beim 10k-Clustering bereits mit Tableau festgestellt haben, auch im 1m-Clustering wieder finden?

Testat: Protokollieren Sie spätestens 1 Woche nach dem Praktikumstermin alle Anweisungen aus Aufgabe 1 und 2 in Ihrem Notebook und geben Sie dieses der Gruppe „Dozent“ frei. Laden Sie weiterhin Ihr erstelltes Tableaubuch als Dokument (pdf) in Moodle hoch.

Bonusaufgabe – Skalierung der Elbow-Method auf AWS Elastic MapReduce

Aufgabenstellung: Berechnung der Cluster-Kosten für 20 verschiedene k zur feingranularen Ermittlung des „Elbows“

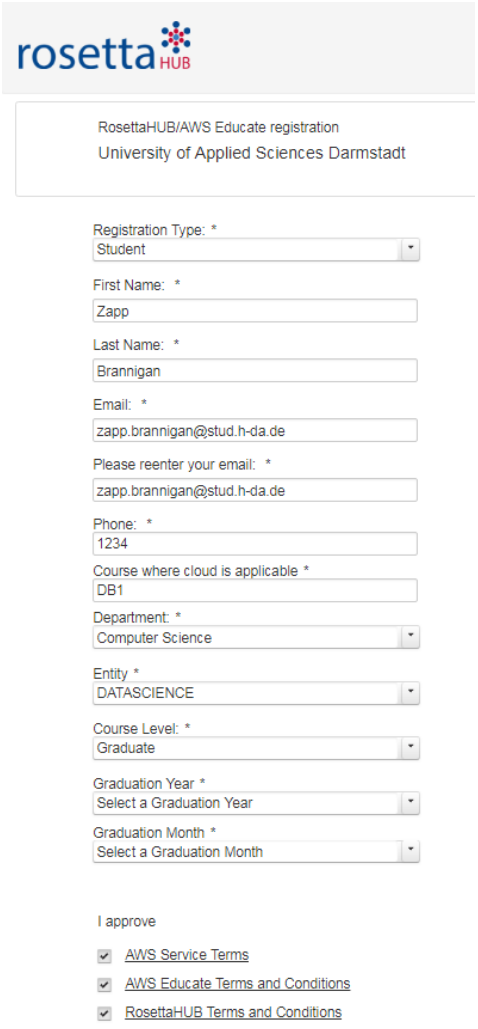
1. Fügen Sie (falls nicht bereits getan) Ihre Notebooks aus Praktikum 1 und Praktikum 2 zu einer sequentiellen Pipeline in einem Notebook zusammen und kommentieren Sie alle Stellen aus, an denen Daten persistiert werden. Modifizieren Sie dieses so, dass je ein Clustering für 20 verschiedene k (z.B. 1 bis 100 in 5er Schritten) für dieselben Ausgangsdaten erstellt, die Kosten pro k ermittelt und (optional) die so entstandene Kurve mit matplotlib visualisiert wird.
2. Testen Sie die Laufzeit für den MSD1m Datensatz (ggf. mehrere Stunden über Nacht) auf dem h_da Cluster und schauen Sie sich den Grad der Parallelisierung an (siehe Praktikum 1).

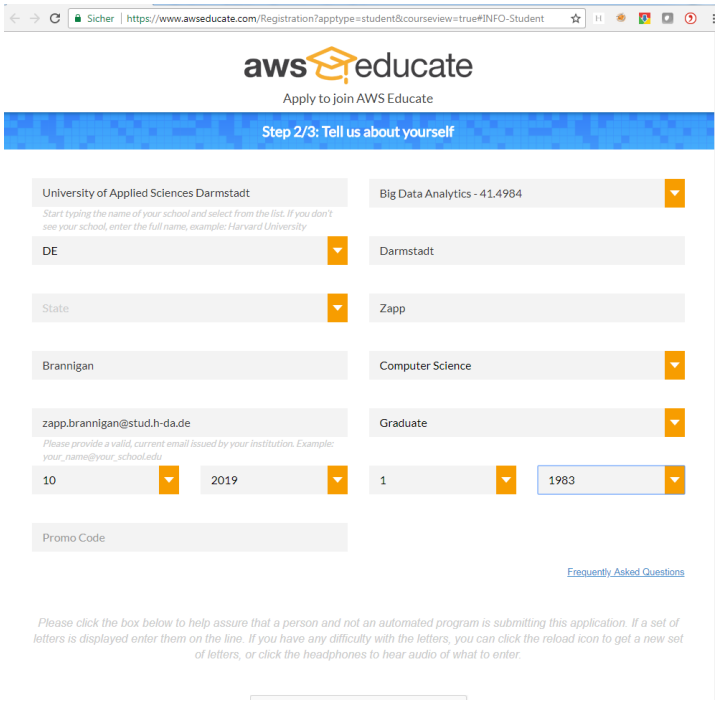
Registrierung bei AWS und Rosettahub

Alternativ zur Nutzung des zentral bereitgestellten „on-premise“ Clusters am FBI kann eine dediziert (persönlich) zur Verfügung gestellte Cluster Umgebung in der Amazon Cloud im Rahmen der Big Data Analyics Veranstaltung genutzt werden. Die Nutzung ist vollkommen freiwillig! Es wird jedoch empfohlen, hier eigene Erfahrungen zu sammeln, da viele Unternehmen in Zukunft mehr und mehr auf ähnliche Cloud-Technologien setzen werden. Die Vorteile sind:

- Verfügbarkeit von überall, auch außerhalb des Hochschulnetzes
- Einfache Administration durch vorkonfigurierte, ständig aktualisierte Releases inklusive Hadoop, Spark, Zeppelin, Hive, u.v.m.
- Einfache vergleichbare Bereitstellung einer flexibel skalierbaren (wachsenden) Cluster-Umgebung zum professionellen Einsatz, z.B. in eigenen Projekten oder beim eigenen Arbeitgeber.
- Börsenähnliches Preismodell (bei geringer Nachfrage sinkt der Preis); Berechnung nur nach Nutzungsintensität.
- Rosettahub ist aktuell in einer Pilotphase als „Cloud Broker“, der die Nutzerverwaltung, Budgetvergabe und den Ressourcenverbrauch bei Amazon für die Studenten der h-da steuert. Bitte registrieren Sie sich zunächst mit Ihrer stud.h-da.de Email-Adresse auf:

- https://h-da.rosettahub.com (Entity: „Datascience“)
- https://www.awseducate.com/Registration?apptype=student&courseview=true#INFO-Student (Institution Name = “University of Applied Sciences Darmstadt”)





- Ihre Telefonnummer müssen Sie nicht angeben. Bei „Graduation Year“ und „Graduation Month“ tragen Sie bitte Jahr und Monat ein, wann Sie voraussichtlich Ihren Master-Abschluss erhalten.
- Die Genehmigung Ihrer Registrierung kann 2-3 Tage dauern; Sie erhalten dann eine Bestätigungsmail.

Anmelden bei Rosettahub, Starten und Konfigurieren des Clusters

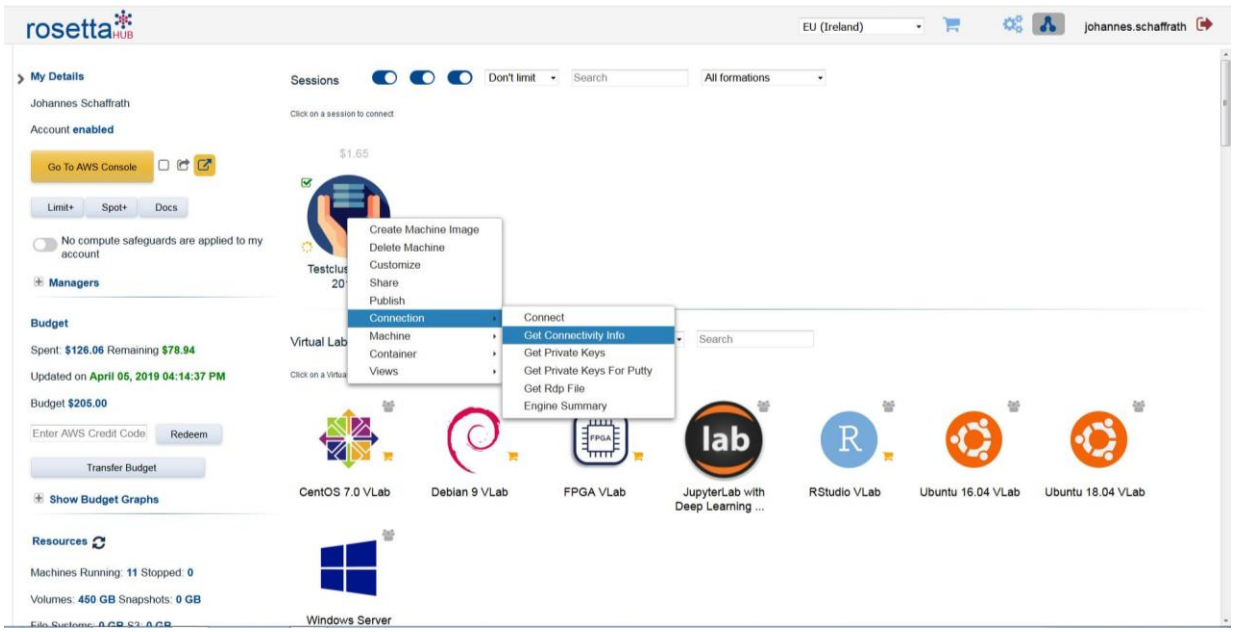
Sobald Sie Ihre Zugangsdaten per Email erhalten haben, können Sie sich einloggen auf:

https://www.rosettahub.com/console/Logon.aspx

Links unten finden Sie Ihr zugeteiltes Budget, und wie viel Sie davon bereits verbraucht haben. Klicken Sie auf die Formation „Testcluster BDA 2019“, um ein eigenes Cluster hochzufahren.

Ihr Cluster hat 10 Knoten und kostet Sie pro Stunde etwa 1.50$. Sobald Sie Ihr Budget überschritten haben, werden alle Ressourcen deaktiviert. Wenn Sie Ihr Cluster nicht nutzen, fahren Sie die Session also mit Rechtsklick-Delete wieder herunter (Cluster bei Amazon sind nicht persistent, d.h. ein Shutdown in diesem Sinn gibt es nicht).

Sobald oben links über der Session ein grüner Haken zu sehen ist, rechtsklicken Sie bitte darauf und dann auf „Connection“ -> „Get Connectivity Info“.



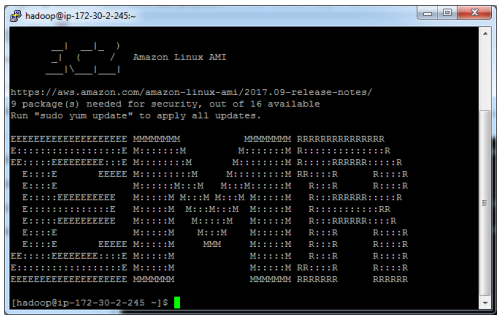
Verbinden Sie sich über die Kommandozeile Ihres Rechners mit

- "Putty Command for Windows" oder
- "Ssh Command for Linux“

Führen Sie dann folgendes Kommando aus:

```
sudo /mnt/config/cluster/connect.sh
```

Sie sollten dann folgende Eingabeaufforderung sehen:



Führen Sie dann folgende Kommandos aus und bestätigen Sie falls nötig mit “yes”:

```
sudo yum install update
sudo yum groupinstall "Development Tools"
sudo yum install python-devel libpng-devel freetype-devel
sudo pip install matplotlib pandas
```

Schließen Sie danach die Eingabeaufforderung wieder.

Klicken Sie dann mit links auf die gestartete Session in RosettaHub, um Ihre Data-Science Workbench aufzurufen.

Skalierung der Kostenermittlung pro k

Laden Sie nun Ihr Zeppelin Notebook vom h_da Cluster herunter und laden Sie es auf das AWS Cluster hoch (via Zeppelin). Dieselben Million-Song Daten, die auch innerhalb des lokalen Clusters verfügbar sind, können aus einem Amazon-Repository mit folgenden PySpark Befehlen geladen werden:

```
segments = sqlContext.\
read.parquet("s3://hda-rosettahub-millionsong/msd1m_timbre_parquet/").\
select(["track_id", "timbre_0"])
segments.rdd.cache()
```

Testen Sie nun die Laufzeit für Ihre k-Kostenermittlung und schauen Sie sich den Grad der Parallelisierung an. Dies tun Sie wie folgt:

1. Aus Rosettahub klicken Sie den gelben Button „Go to AWS Console“
2. Klicken Sie auf "EMR“ oder suchen Sie ggf. nach diesem Service

3. Klicken Sie auf die aktuell laufende Cluster-Instanz
4. Klicken Sie auf den "Application History“ Tab, dann auf den Tab „Stages“
5. Klicken Sie eine beliebige Stage mit „KMeans“ in der Beschreibung an
6. Klicken (und erweitern Sie damit) die „Event Timeline“

Wie stark wird Ihre k-Means Berechnung im Vergleich zum h_da Cluster parallelisiert? Welchen Laufzeitvorteil würden Sie erwarten? Welche Gründe könnte es geben, dass der Laufzeitvorteil ggf. nicht voll erreicht wird?

Wichtig: Fahren Sie das Cluster nach der Benutzung mit „Delete“ wieder herunter!