UNIVERSITÄT LEIPZIG

Recommendation & Machine Learning

Konzeptioneller Entwurf

---

1. Testat

Autor	Matthias Frei
	Simon Ganz
Matrikelnummer	3742806
	3741513
Studiengang	M.Sc. Informatik
Bearbeitungszeitraum	SS 2016
Veranstaltung	Big Data Praktikum
Betreuer	Dr. Eric Peukert

Inhaltsverzeichnis

Kapitel 1

Einleitung

In diesem Projekt soll ein „Movie Recommendation System“ mithilfe des Frameworks Spark[1] umgesetzt werden.
Das System soll dabei zunächst mit einer festen Anzahl an Nutzern, Filmen und Bewertungen trainiert werden, um anschließend Vorhersagen für Filme zu treffen, die einem bestimmten Nutzer gefallen könnten. Die hierfür verwendeten Daten werden von der Plattform Movielens[2] bezogen.
Die notwendigen Daten finden sich in einer CSV Datei mit dem Name `ratings.csv`. Jede Zeile besteht aus einer Nutzer-Id, einer Film-Id, einer Bewertung und einem Zeitstempel. Das System soll nun Bewertungen für die Nutzer-Film-Paare vorhersagen, die in dieser Datei nicht aufgelistet sind.
Diese Vorhersagen werden durch den Einsatz eines Machine-Learing-Algorithmus auf den gegebenen Daten generiert.

Im Folgenden werden das Framework Apache Spark und der in diesem Projekt verwendete Algorithmus kurz vorgestellt.

Kapitel 2

Apache Spark

Apache Spark ist ein „Cluster Computing Framework“ das aus mehreren einzelnen Komponenten besteht. Die Grundlage des Spark Systems wird dabei durch den „Spark Core“ gebildet. Er stellt die grundlegenden Funktionalitäten wie beispielsweise die Aufgabenverteilung, die Ein- und Ausgabe von Daten oder das Scheduling von Prozessen zur Verfügung.

Spark verwendet als Datenstruktur das sogenannten „Resilient Distributed Dataset“ (RDD) auf denen die entsprechenden Spark-Operationen ausgeführt werden. RDDs können dabei auf mehrere Rechner verteilt werden um parallele Berechnungen zu ermöglichen. Sie werden entweder aus externen Quellen gebildet oder als Ausgabe von verschiedenen Transformationsanwendungen erzeugt.

Neben dem „Spark Core“ stellt die Machine Learning Library (MLlib) einen wichtigen Bestandteil von Spark dar und ist für die Umsetzung des „Recommendation Systems“ essentiell. Bei der MLlib handelt es sich um eine Funktionsbibliothek, welche verschiedenen Machine-Learing-Algorithmen zur Verfügung stellt. Für das zu konstruierende System werden wir das „Collaborative Filtering“ verwenden, das im nächsten Abschnitt kurz erklärt wird.

Kapitel 3

Collaborative Filtering und ALS

„Collaborative Filtering“ ist eine weitverbreitete Technik für „Recommendation Systems“.
Diese Methode begründet sich durch die Annahme, dass falls zwei Personen A und B das selbe Interesse bezüglich einer Sache X haben, es wahrscheinlicher ist, dass Person A in einer anderen Sache Y mit B übereinstimmt, als mit einer anderen zufälligen Person. Im Fall des „Movie Recommendation System“ bedeutet das, dass einem Nutzer Filme eines anderen Nutzers gefallen könnten, wenn beide Nutzer bestimmte Filme ähnlich gut bewertet haben.

Um diese Filmvorschläge zu bestimmen müssen die leeren Einträge der Nutzer-Film-Matrix ausgefüllt werden. Diese Lücken entstehen wenn ein Nutzer noch keine Wertung für einen Film gesetzt hat. Die vorhergesagten Werte können dann als Empfehlungen an den Nutzer interpretiert werden. Die Umsetzung in Spark erfolgt durch den „Alternating Least Squares“ (ALS) Algorithmus. Dieser Algorithmus schätzt durch eine Faktorisierung der Nutzer-Film-Matix die fehlenden Einträge. Hierzu wird abwechselnd einer der beiden Matrizen festgehalten und das daraus resultierende quadratische Gleichungssystem gelöst. So können die fehlenden Einträge der Matrix iterativ berechnet werden. Als Eingabe fordert der Algorithmus die Nutzer-Film-Matrix, die zuvor aus einer CSV Datei eingelesen wird. Für die weiteren Parameter werden die default Werte verwendet. In Kapitel 5 wird ein kurzer Überblick über den Ablauf des Systems vom Einlesen bis zur Ausgabe der Ergebnisse gegeben.

Kapitel 4

Benutzerschnittstelle

Die Interaktion eines Anwenders mit dem Recommendation System soll über eine webbasierte Schnittstelle erfolgen.
Ein Webserver soll durch Drittanbieter-Bibliotheken direkt in Java implementiert und bereitgestellt werden. Dies ermöglicht eine einfache und direkte Verknüpfung von Weboberfläche und Seiten- bzw. URL-Aufrufen, die eine direkte Ansprache von API-Schnittstellen des Spark-Frameworks ermöglichen.
Für die Entwicklung der Weboberfläche sollen HTML5 und JavaScript eingesetzt werden. Die Funktionalität der Seite soll durch jQuery und dem Webframework „Bootstrap[3]“ interaktiv gestaltet werden können. Ein vorgefertigtes Bootstrap-Template (siehe Abbildung 4.1) dient hierbei als Grundlage[4].

Ein Besucher der Weboberfläche soll eine Liste mit allen Filmen sowie automatisch generierte Teillisten, z.B. die 100 am besten bewerteten Filme, einsehen können.
Der Nutzer hat weiterhin die Möglichkeit, selbst Filme zu bewerten. Basierend auf seinen Bewertungen wird ihm eine Liste mit Filmen angezeigt, die ebenfalls seinen Vorstellungen entsprechen könnten.

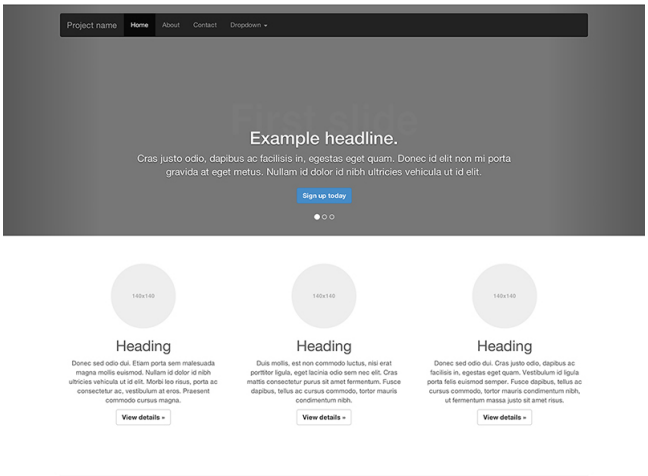

Abbildung 4.1: Bildschirmfoto des Carousel-Templates für Bootstrap

Kapitel 5

Systemaufbau

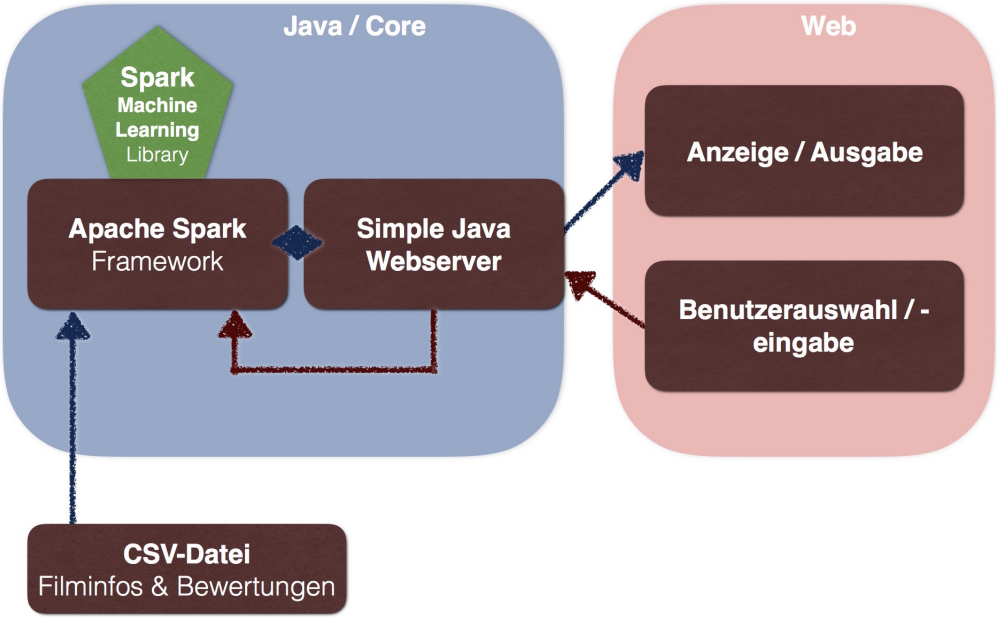

Abbildung 5.1: Aufbau des Recommendation Systems

Abbildung 5.1 zeigt den geplanten Aufbau des Systems.
Grundlage ist ein auf Java basierender Kern, der den Zugriff auf Apache Spark und den Webserver ermöglicht. Die Machine Learning Bibliothek von Spark wird zu Beginn mit bestehenden Filmbewertungdsaten (XX) trainiert. Diese werden aus einer CSV (Comma Seperated Value) Datei, welche aus dem Dateisystem gelesen wird, extrahiert.

Filminformationen und Bewertungen werden anschließend über einen in Java implementierten Webserver bereitgestellt. Der Nutzer kann sich diese somit in seinem Browser anzeigen lassen. Tätigt er eine Auswahl oder bewertet selber Filme, werden diese Informationen an den Webserver übertragen, welcher sie wiederum an Spark weiterreicht. Hier werden nun anhand der Nutzerdaten passende Vorschläge für ihn generiert und wieder über den Webserver an den Nutzer ausgeliefert.

Literaturverzeichnis

[1] The Apache Software Foundation: *Spark Overview*. https://spark.apache.org/docs/latest/index.html, Abruf: 24. Mai 2016 um 20:07

[2] GroupLens: *MovieLens*. http://grouplens.org/datasets/movielens/, Abruf: 24. Mai 2016 um 20:15

[3] The Bootstrap Core Team: *Bootstrap*. http://getbootstrap.com, Abruf: 24. Mai 2016 um 20:12

[4] The Bootstrap Core Team: *Carousel Template for Bootstrap*. http://getbootstrap.com/examples/carousel/, Abruf: 24. Mai 2016 um 20:19

[5] The Apache Software Foundation: *Machine Learning Library (MLlib) Guide*. http://spark.apache.org/docs/latest/mllib-guide.html, Abruf: 24. Mai 2016 um 20:09