**Ludwig-Maximilians-Universität München**
**Institut für Informatik**
Prof. Dr. Matthias Schubert
Julian Busch, Daniyal Kazempour

Munich, 22.11.2017

# Big Data Management and Analytics

WS 2017/18

## Tutorial 4: Introduction to Spark using PySpark

### Assignment 4-1 *Spark & PySpark*

In this assignment we are going to become a bit more familiar with Spark

(a) First make sure that Java ($\geq 1.8$) is installed. Download Spark from http://spark.apache.org/downloads.html and install it on your local machine. It already includes the Spark Python API PySpark.

(b) Implement the word count example using PySpark.

### Assignment 4-2 *MapReduce using PySpark*

The aim of this assignment is to solve various problems on a given data set using MapReduce.

Given a RDD *dataset* which consists of the following data rows:

| id | product | spent | rating |
|---|---|---|---|
| 1 | 'chocolate cookies' | 2 | 'awesome' |
| 2 | 'peanut butter cookies' | 4 | 'satisfactory' |
| 3 | 'chocolate cookies' | 1 | 'awesome' |
| 4 | 'chocolate cookies' | 1.5 | 'awful' |
| 5 | 'macadamia cookies' | 8 | 'awesome' |
| 6 | 'peanut butter cookies' | 1 | 'awesome' |
| 7 | 'macadamia cookies' | 4 | 'satisfactory' |
| 8 | 'chocolate cookies' | 2 | 'satisfactory' |
| 9 | 'peanut butter cookies' | 3 | 'awesome' |
| 10 | 'chocolate cookies' | 1 | 'awful' |

The dataset has been extracted from a customer analysis database. Each row describes a customer. The column name id represents the customer id, product represents the sort of cookies which has been acquired, spent stands for the amount of money which has been spent and rating reflects how the customer has rated the acquired cookies.

(a) The following expression below is being executed (please note that *reduce()* is an action):

```
result = dataset.map(lambda (id, product, spent, rating):  (rating, spent)).\
    filter(lambda (rating, spent):  rating == 'awesome').\
    map(lambda (rating, spent):  spent).\
    reduce(lambda first, second:  max(first, second))
```

What is the value of *result*? Describe within one sentence the meaning of the result in context of customer evaluation.

For the following tasks you can either create the RDD entries of the table above manually, or you can first import the csv file *customercookies.csv* by using the following commands:
*rdd = session.read.csv('customercookies.csv', header=False).rdd*

(b) Create a PySpark query in which for each product type the average money that has been spent is computed where the customer rating is **'awesome'**. For this purpose solve the sub-tasks below:

(i) Write a PySpark query which returns the number 'awesome' ratings per product.

(ii) Write a PySpark query which returns how much money has been spent per product only by customers who gave an 'awesome' rating.

(iii) Write a PySpark query which computes the average of money spent by customers for each product, considering only the awesome ratings. Hint: use (i) and (ii) to solve this assignment. E.g.: for chocolate cookies we have 2 entries in the dataset which have been rated with 'awesome'. The total amount of money spent for the two awesome ratings of chocolate cookies is $2 + 1 = 3$. The average money spent for chocolate cookies by customers who gave an awesome rating is therefore $\frac{3}{2} = 1.5$.

## **Assignment 4-3** *k-Means with Spark*

In this exercise, we are going to implement the $k$-Means algorithm with PySpark. In order to achieve this proceed as follows:

1. Download the code template *kMeans_template.py* and become familiar with it.
2. Implement the two functions *assign_to_centroid*, *calculate_new_centroid* and fill in the sections marked as *TODO* with your code.
3. Test your implementation by running $k$-Means on some test datasets. You can use the datasets provided for Exercise 2-6. In the Figure, you can navigate between iterations using the left and right arrow keys.