Big Data

Management and Analytics

Assignment 2

TU Clausthal, Institut für Informatik

!!! Due date: 20. May 2020, 1pm !!!

In this assigment you will further practise your programming skills with the programming language Python and become more familiar with the popular data science platform Anaconda (`https://www.anaconda.com/`. In addition, you will get some first experience with the Python library Pandas (`https://pandas.pydata.org/`) for big data analysis.

If you have not already installed Anaconda on your computer, then please download it from the website `https://www.anaconda.com/download/` with Python version 3.7.

Look through the Jupyter Python notebooks which are included in Anaconda. If you like, you can also install an IDE/editor of your choice for writing your Python code. Please make sure to have Python (version 2.7) setup up and running for the future tutorials.

Read the following notes in detail for your submission of solutions.

- You can work in teams of up to 2.
- You can send us your solutions via the GATE-System `https://si.in.tu-clausthal.de/`
- Solutions sent after the deadline will automatically be marked with 0 points and treated as not participated in the respective assignment!
- Copying solutions from other students will be treated as cheating and will lead to exclusion from the course!

Here are some useful links for learning more about Python:

Mark Pilgrim - Dive into python: `http://www.diveintopython3.net/`

Python 3.7 Documentation: `https://docs.python.org/3.7/`

!! Submit your solutions via the GATE-System !!

There is guide in Moodle on how to register and how to upload your solutions.

Task 1 **5 Marks**

In the first task, we use so called *lambda-functions.* You can get general information about it, if you follow this link: `https://www.w3schools.com/python/python_lambda.asp`

(Important note: Do NOT mix them up with the term "lambda architecture"!)

We use the map, filter and reduce paradigms. For them, you can find more information and tutorials here: `https://www.learnpython.org/en/Map%2C_Filter%2C_Reduce`. Solve the following tasks and write your solutions into a .txt file.

(a) Write a list comprehension, which takes a number n and returns a list with even numbers, using a lambda function.

(b) First write a function, which takes a length in inch and returns a length in cm. Given a list l with lengths in inches: $l = [4, 4.5, 5, 5.5; 6, 7]$. Write a list comprehension, which takes l and returns a list with all values converted to cm using *map*.

(c) Write a list comprehension, which filters the list l from (b) by returning only sizes between 4 and 6 inches. Use $filter$ for this!

(d) Write a list comprehension which reduces the list l by summing up all lengths.

Hint: For using the *reduce* function, you need to import it first by adding the line: *from functools import reduce*

Task 2 **5 Marks**

In this task, we use the *Pandas* library for big data analysis. For the documentation see here: `https://pandas.pydata.org/pandas-docs/stable/`. We use data records of a movie database, which is provided as a csv-file in Moodle, namely "*Moviedata.csv*". Do the following tasks and write the code, which solves the task, into a .txt-file.

(a) Read the .csv-file as a DataFrame for further processing using *pandas.read_csv()*. Afterwards inspect the read .csv-file using *.shape*, *.columns*, *.info* and *.describe()*. Lastly display the first five records of the data-set using *.head(5)* and the last five records using *.tail(5)*.

(b) Select the first five records from the data set, but those records shall only contain the following columns in your output: *movie title*, *duration* and *num voted users.* Write the code, which archives this output.

(c) Select the first five movies containing the genre "*Action*". Display only the columns "*movie*", "*title*" and "*genres*".

(d) Sort the action movies by their *imdb score* and display the names and scores of the top-10 scored movies.

(e) Group the movies by column *director* and display the top-10 directors with the highest mean *gross* of their movies.

(f) Delete all rows, which contain at least one missing value. Visualize parts of the data using *pandas:plotting:scatter_matrix* and *DataFrameGroupBy:hist.*

+++++