Lab Course Machine Learning
Exercise Sheet 1

Prof. Dr. Dr. Lars Schmidt-Thieme, Mohsan Jameel
Information Systems and Machine Learning Lab
University of Hildesheim

October 21st, 2017
Submission on October 27th, 2017 at 11:55pm, (on moodle, course code 3112)

Exercise 1: Python Tutorial (10 Points)

Install Anaconda (4.2.0 (Python v3.5, v2.7); `https://www.continuum.io/downloads`.

Part A: (5 Points): IPython In this task you are required to use IPython (a web version provided by Jupyter nootbook). You have to write a word count program. Your program should read a text document (download from `https://raw.githubusercontent.com/python/cpython/master/README`. You can take help of modules covered in the lecture slides. You should save your IPython session and it should include Headings and comments at some important steps to explain the working of code.

Part A: (5 Points): numpy Using numpy you are required to use numpy for operation on matrices. Create a matrix A of dimensions $n \times m$, where $n = 100$ and $m = 20$. Initialize Matrix A. Create a vector **v** of dimension $m \times 1$. Initialize the matrix with a random values and vector with normal distribution using $\mu = 2$ and $\sigma = 0.01$. Perform following operation on them

1. Iterative multiply (element-wise) each row of matrix A with vector **v** and sum the result of each iteration in another vector **c**
2. Find mean and standard deviation of the new vector **c**.
3. Plot histogram of vector **c** using 5 bins

Exercise 2: Linear Regression through exact form (10 Points)

In this exercise you will implement linear regression that was introduced in the introduction lecture of Machine learning `https://www.ismll.uni-hildesheim.de/lehre/ml-15w/script/ml-01-0-overview.pdf`.The learning algorithm is given on the slide 20 and prediction algorithm is given on the slide 14.

1. Generate a simple data i.e. a matrix A with dimensions 100×2. Initialize it with normal distribution $\mu = 2$ and $\sigma = 0.01$
2. Implement LEARN-SIMPLE-LINREG algorithm and train it using matrix A to learn values of β_0 and β_1
3. Implement PREDICT-SIMPLE-LINREG and calculate the points for each training example in matrix A.
4. Plot the training points from matrix A and predicted values in the form of line graph.
5. In the end use numpy.linalg lstsq to replace step 2 for learning values of β_0 and β_1.