Lab Course Machine Learning
Exercise 1

Prof. Dr. Dr. Lars Schmidt-Thieme,
Mofassir ul Islam Arif
Information Systems and Machine Learning Lab
University of Hildesheim
Submission: 9.11.2018 LearnWeb 3112

November 2, 2018

1 Exercise Sheet 1

1.1 Pandas and Numpy (10 Points)

- **Word Count Program**: In this task your are required to use the provided text file and write a program that will count the number of occurances of unique words.

  The program should ignore words like {'the', 'a','an','be'}

  Finally you are required to generate the histogram of the top 10 most occuring words. Click here to download Text File

- **Matrix Multiplication**: 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

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

1.2 Linear Regression through exact form. (10 Points)

In this exercise you will implement linear regression that was introduced in the introduction Machine Learning Lecture.

- Generate 3 sets of simple data. i.e. a matrix A with dimensions 100×2. Initialize it with normal distribution $\mu = 2$ and $\sigma = [0.01, 0.1, 1]$

- Implement LEARN-SIMPLE-LINREG algorithm and train it using matrix A to learn values of β_0 and β_1
- Implement PREDICT-SIMPLE-LINREG and calculate the points for each training example in matrix A.
- Plot the training points from matrix A and predicted values in the form of line graph.
- Comment on the effect that σ has on the line that is predicted.
- Put β_0 to zero and rerun the program to generate the predicted line. Comment on the change you see for the varying values of σ
- Put β_1 to zero and rerun the program to generate the predicted line. Comment on the change you see for the varying values of σ
- In the end use numpy.linalg lstsq to replace step 2 for learning values of β_0 and β_1