Programming Assignment 1

Write a program (you can do that in a group of 3 to 5 people, but everybody has to submit individually!) that implements a (batch) linear regression using the gradient descent method in *Python 3* (On the server, Python 3.6.8 is installed.). Use the following gradient calculation:

$$gradient = \sum_{i=1}^{N} \vec{x_i}(y_i - f(\vec{x_i}))$$

$$\vec{w} \leftarrow \vec{w} + \eta \cdot gradient$$

where $\vec{x_i}$ is one data point (with N being the size of the data set), η the learning rate, y_i is the target output and $f(\vec{x_i})$ is the linear function defined as $f(\vec{x}) = \vec{w}^T\vec{x}$ or equivalently $f(\vec{x}) = \sum_i w_i \cdot x_i$. Whereas $\vec{w}$ and $\vec{x}$ include the bias/intercept, i.e. w_0 (x_0 is always 1). All weights should be initialized as 0.
Given are the two random example data sets (uploaded in Moodle) named *random1* and *random2* as csv files. Use those data sets as examples of what the server will give your program as input. Your task is to correctly implement the gradient descent method and return for each iteration the weights and sum of squared errors until a given threshold of **change** in the error is reached[1]. The output of your algorithm should be printed onto the console/terminal and should look like this.

```
iteration_number,weight0,weight1,weight2,...,weightN,sum_of_squared_errors
```

Please do **NOT** print any extra information onto the output. The solution (rounded to 4 decimals) for the data sets are given with a learning rate of 0.0001 and a threshold of 0.0001. With that, you can check the correctness of your solution. Please be reminded, that small rounding errors are normal and will be treated as correct. If the program fails or the data format is incorrect you will get zero points.
Your program **must be** named `student.py` and **must** accept the following parameters:

1. **threshold** - The threshold, that the change in error has to fall below, before the algorithm terminates.
2. **data** - The location of the data file (e.g. /media/data/yacht.csv).
3. **eta** - The learning rate of the gradient descent approach.

The server will start your program in the following way:

```
python3 student.py --data random.csv --eta 0.0001 --threshold 0.0001
```

The final program code must be uploaded to Moodle (in the respective VPL assignment) until Wednesday, the 25th of November 2020, 1am. The code will be automatically checked against randomly created data sets. You will get at most two points, if the output of your regressor is correct.

2 points

[1] Meaning $e_t - e_{t+1} <$ threshold