PREDICT VIRALITY - EXERCISE

In this exercise you will attempt to predict the future of a video uploaded online. The question we are asking is: how popular a given video will be one week after its publication? Although we know that having a cat in a video increases the chances of your video becoming viral, for the purpose of this toy example we will use only time evolution of the number of views within the first 24 hours after publishing a video. The popularity of our video is measured as the number of views one week (168 hours) after publication. Attached document `data.csv` contains the evolution of views for almost 1,000 videos of a single Facebook page and it is stored in the following format:

```
id(1), v(1), v(2), ...  , v(168)

id(2), v(1), v(2), ...  , v(168)

...

id(916), v(1), v(2), ...  , v(168)
```

where $v(n)$ defines number of views recorded n hours after publishing a video which can be identified by its *id*.

Your task is to propose a method to predict the number of views one week after publication using this data. You should analyse the input data to discover underlying pattern and propose an algorithm that predicts the future popularity. You should also evaluate the method and plot the results of evaluation for $n = 1, 2, ..., 24$ hours. We will guide your through this process.

You can use whichever programming/statistical tool you like, but we encourage you to become familiar with Matlab or Octave, as this will be useful when you start working with us. Use github[1] *to version your code.*

(1) *Read in the* `data.csv` *file and analyse the basic statistics of the* $v(n)$ *or* $n = 24, 72, 168$.
(2) *Plot the distribution of the* $v(168)$. *How would you describe the distribution of the views?*
(3) *Plot the distribution of the log transformed* $v(168)$. *Does it ring a bell?*
(4) *To improve the generalization and performance of your prediction algorithm, you can remove the so-called outliers from the dataset. To that end, compute the mean value* μ *and standard deviation* σ *of the log transformed* $v(168)$. *Remove from the dataset data for videos where* $v(168)$ *does not fit within* 3σ *rule.*
(5) *Compute correlation coefficients between the log-transformed* $v(n)$ *for* $n = 1, 2, ..., 24$ *and* $v(168)$.
(6) *Randomly split the dataset into training and test sets (10% of the dataset should be used for testing, rest for training).*
(7) *Using training data, find linear regression model that minimizes Ordinary Least Squares (OLS) error function. It should take as the input* $v(n)$ *and output* $v(168)$.
(8) *Extend the above linear regression model with multiple inputs, that is it for a given time* n *the model should take an array of view counts preceding time* n: $\{v(i)\}_{i=1}^{n}$

[1] `https://github.com/`

(9) *To evaluate the proposed predictors, compute mean Relative Squared Error (mRSE), that is defined as:*

$$mRSE = \frac{1}{|T|} \sum_{id \in T} \left(\frac{\hat{v}_{id}(168)}{v_{id}(168)} - 1 \right)^2$$

where $\hat{v}_{id}(168)$ is the number of views of video id from a testing dataset T estimated by a predictor and $v_{id}(168)$ is a true recorded value.

(10) *Plot the mRSE values for $n \in (1, 24)$ computed on the test dataset.*

Voilà! You have just completed the first step towards predicting the future and you're on a good way to become the next Diviner Maciej[2]*. The final plot should look more or less like that (the values can be different):*

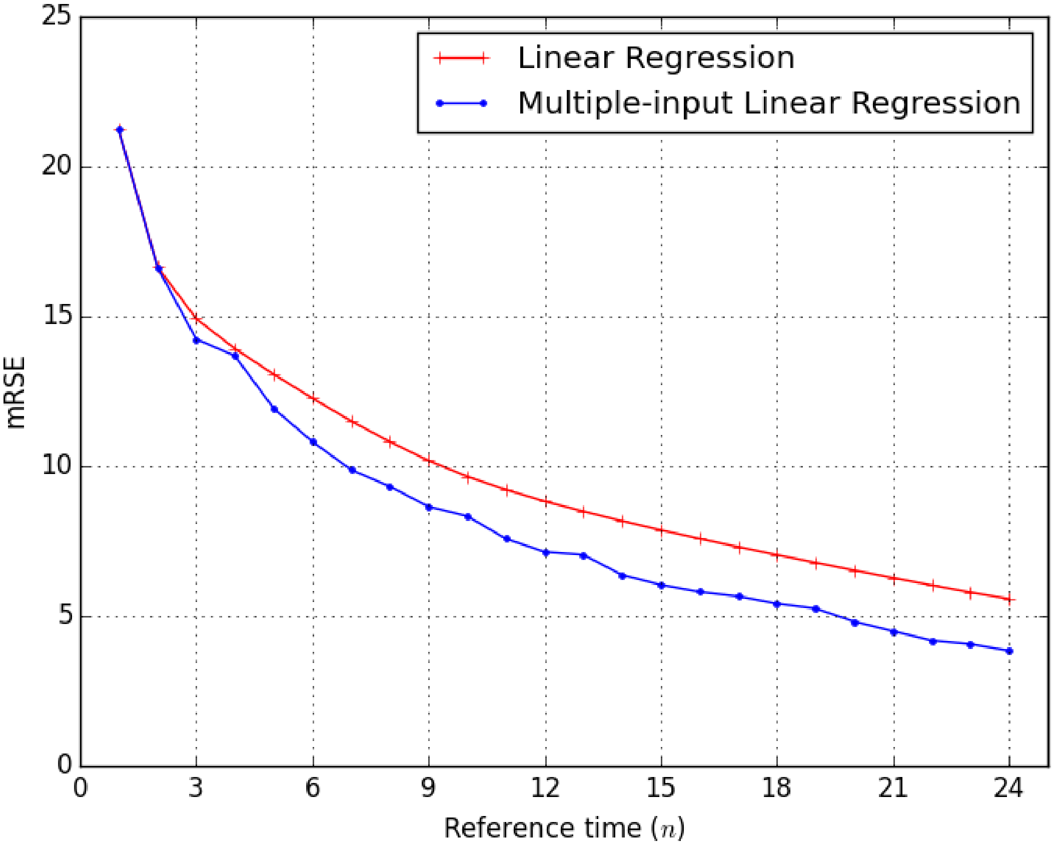

FIGURE 1. Performance of linear regression models for $n \in (1, 24)$ hours measured as mean Relative Squared Error (mRSE).

[2] `http://wrozbita-maciej.pl/`