# TECH-GB 3332 Fall 2020 Prof. Alex Tuzhilin

## Term Project: Deep Learning for Sentiment Analysis

*Due date:* December 16

In this term project, you will deploy Deep Learning models to build a classification model using RapidMiner to predict the sentiment of consumers towards US airlines based on their reviews expressed in the form of tweets. If you strongly prefer to use some other DL-based software/frameworks instead of RapidMiner, such as TensorFlow or PyTorch, let me know before starting the work. This is a group project, and you should work on it in the groups that you have formed already*.*

**Part 1**. Loading the Data (“Retrieve” operator)

The data is provided to you in two versions:

1. The original version of the tweets (and their sentiments) is located at <https://drive.google.com/file/d/1atyRH5Yz7TU-2ziyZknfd7ib2LLwYeuv/view?usp=sharing>
2. The preprocessed version of the tweets is located at <https://drive.google.com/file/d/1c96crlNZr7XiF3-9lmZ1nEJaY3MHTTz5/view?usp=sharing>, where text preprocessing and pre-training of the text embeddings of the tweets using autoencoders have already been done to make your life simpler. This preprocessed version contains the sentiments about the tweets in column 1 of the spreadsheet (either positive (1) or negative (0)) and the 8-dimenisonal pre-trained embeddings of the tweets (in columns 2 – 9 of the spreadsheet).

I recommend that you use the preprocessed version of the tweets since it will save you a lot of preprocessing work to build these embeddings that is non-trivial. However, if you like challenges, you can do preprocessing and building the embeddings using autoencoders yourself and, therefore, work directly with the “raw” tweets. As a “reward” for this extra work, *you will be awarded 10 extra points* (the max score of this project is 100) if you preprocess tweets yourself.

**Part 2**. Preprocessing Data (“SubProcess” operator)

Your task is to predict the score of the sentiment (positive or negative) between 0 and 1 based on the embeddings of the tweets specified in columns 2 – 9 of the pre-possessed spreadsheet (or the original tweets if you decided to work with the raw tweeting data). To evaluate the performance of your model, please split the dataset into the train set and the test set in the 0.8:0.2 ratio and use cross-validation to calculate the prediction performance.

(*Hint*: you need to set the “Set Role” and “Split Data” operators properly).

**Part 3**. Neural Networks (“Deep Learning” operator)

You can use any neural network model you like for this classification task. In particular, you may start with a simple single fully connected network as a “baseline” and then try to use more complex models, including CNN and RNN based models, to achieve better performance results than this simple baseline model. Your goal is to reach the mean absolute error of at least 0.48, which should not be too difficult. If you want to be more ambitious, you can try to reach the mean absolute error of 0.47 (medium difficulty), or even 0.46 (this is difficult). The higher accuracy you get, the more points you will be awarded.

In addition to the simple NN baseline mentioned above, you should also build another basic baseline, such as a logistic regression model (similar to the one we used in the RapidMiner Lab done in the class) and compare the performance results of your DL-based model with that baseline. The expectation is that the more sophisticated DL-model should outperform simple baselines.

(*Hint*: You need to set the number of neurons and the activation function for the output layer properly.)

**Part 4**. Evaluation (“Apply Model” and “Performance” operators)

After you build your neural network, apply the trained deep learning model to the test set and evaluate its performance using the accuracy measures.

**Deliverables:**

1. Design of your deep learning model in the form of screenshots or any other means clearly showing the architectural design of the network, including the inputs, the outputs and various modules of the model. Also, you should explain your network design decisions (in a separate document), i.e., why you decided to use specific types of networks, selection of hyperparameters and other types of considerations you used when working on the network design.
2. Performance results (mean absolute error) of your implemented model and their comparison with the simple baseline models described in Part 3. Also report the running times of your model (the more efficient it is, the better).

Your project will be judged based on the quality of the neural network design (and its justification) and on the performance accuracy of your results (the higher, the better).