David Bolshov
March 18, 2023.

# Results Report
## Building a predictive NLP model

Task:

1. Train a model in Python to classify feedback and evaluate its quality and accuracy on a test sample.
2. Develop a web service based on Django framework for inputting movie reviews with automatic assignment of a rating (from 1 to 10) and comment status (positive or negative), and deploy the service in the public domain.

Tools used:

- Python, HTML, CSS, JavaScript.
- Python libraries including PyTorch, transformers, pandas, numpy, sklearn, scipy, catboost, Django.
- Pre-trained DistilBERT model for text classification (https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english).

Workflow:

A DistilBERT model pre-trained on the text classification task was chosen to solve the problem. DistilBERT was chosen because of its high performance and small size.

Different regression models trained on the logits of the DistilBERT pre-trained model were tested, such as linear regression, Lasso regression, Ridge regression, ElasticNet, LassoLars, SVR, LinearSVR, k-nearest neighbor method. The combination of ridge regression for rating prediction and ridge classifier for binary prediction of comment status showed the best result.

Next, the DistilBERT model was fine tuned on the IMDB Review Dataset (https://ai.stanford.edu/~amaas/data/sentiment/). Two approaches were tested - replacing the loss function with MSE so that the output logits were the final prediction (redefining into a regression model) and the classical 8-class prediction, each corresponding to a different rating (1, 2, 3, 4, 7, 8, 9, 10). The second approach showed significantly better results.

To solve the problem of incomplete classes (mean scores 5 and 6 were omitted from the data), a weighted sum of model scoring was attempted. In the course of which, after applying the softmax function to the model logits, the products of the scores (probability) of each class and its absolute value (1, 2, 10, etc.) were summed. This approach not only made it possible to predict the entire range of ratings, but also reduced the expected MSE of the model by 23% (from 3.1 to 2.4), and reduced the MAE by 4% (from 0.94 to 0.9), indicating that there are significantly fewer large errors of more than 1 rating unit in the predictions.

Several variations of the hyperparameters learning rate and weight decay were tested (testing was performed with no more than 15 epochs). Changing the latter had no significant effect on the learning process; the best value of the learning rate was chosen $10^{-5}$. Since pronounced overtraining of the model began after the 5th epoch, training at 5 epochs was used to train the final model and evaluate the quality on the deferred sample.

Next, experiments were conducted with ridge regression and classification over the logits of the fine tuned DistilBERT model with and without prior application of softmax, as the models performed best on the logits of the pre-trained neural network. However, this approach did not improve the expected quality.

The Django framework was used to develop the graphical web interface. The web service allows the user to enter a text of movie review and get a prediction of the rating (1 to 10) and comment status (positive or negative).

The service has been deployed on cloud hosting Pythonanywhere. The service is available at https://dlbolshov.pythonanywhere.com .

## Final results on a delayed sample for the pre-trained DistilBERT:

Movie rating prediction metrics:

MSE - 2.43

MAE - 0.9

Review status prediction accuracy:

accuracy - 0.94