Practical task 11.

Implementation of the simplest solution for the Kaggle competition

In this practical task you will implement the simplest solution for the Kaggle competition based on the tf-idf representations and random forests.

1. Read the training data. In this task we will implement only the simplest solution and will use only the feedback texts and scores. Each line in the training set file can be converted to the Python dictionary via *eval(line)*. After this step you should obtain a list of texts and numpy.array of scores. NB: some of the feedback texts can be incorrectly encoded: just omit them at this step.

2. Obtain the tf-idf representations via TfidfVectorizer from the sklearn library. Set the parameter *stop_words* to "english". You have to find the optimal value for the parameter *min_df* by yourself. NB: small values of *min_df* result in a large number of features and considerable overfitting risk. Large values of *min_df* prevent overfitting but result in information losses.

3. Randomly split the data into training (50000 items) and validation sets. Train the random forest regression (RandomForestRegressor from the sklearn library) on the training set. Find the optimal parameters of the random forest by yourself. Evaluate the MSE error of the obtained regressor, make sure that it is significantly smaller than the MSE error of the trivial baseline (to predict the mean score for all items from the validation set).

4. Check the interpretability of the obtained regressor. Obtain the feature importances via the field *feature_importances_* of RandomForestRegressor. Print the words corresponding to the 10 most important features (Hint: use *get_feature_names()* method of TfidfVectorizer). Are they reasonable?