# 1 Введение

- Получение фич
- Создание модели
- Предсказание

# 2 Получение фич

Считываем данные в pandas.DataFrame. Разобъём данные на train и test.

**lang**: На основе обучающей выборки посчитаем среднее количество ретвитов для каждого значения 'user.lang'. На основании этого получим значения для тестовой выборки. Если для какого-нибудь значения 'user.lang' в тестовой выборке небыло таковых в обучающей, то выберем в этом случае среднее средних значений по всем значениям 'user.lang'.

$$\begin{aligned} table &= train\_data[['user.lang','retweet\_count']].groupby(by='user.lang').mean() \\ table &= table.to\_dict()['retweet\_count'] \end{aligned}$$

**text**: Из текста возьмём количество вхождений '@', 'http', '#' и '?'.

**description**: Здесь берём то же, что из поля text.

**time zone**: Аналогично user.lang.

А так же возьмём в качестве фич все не текстовые поля.

# 3 Создание модели

В качестве моделей возьмём ensembly.RandomForestClassifier и ensembly.GradientBoostingClassifier. Было желание попробывать svm.SVC, но за 2 часа работы всё это так и не отработало :c

Для перебора параметров будем использовать model_selection.GridSearchCV (в старых версиях scikit-learn - grid_search.GridSearchCV) с 3-кратной кросс-валидацией.

# 4 Предсказание

GradientBoostingClassifier выдаёт примерно одинаковое качество на обучении и тестировании, поэтому для предсказания выберем его.