

# 1 Описание

- Получение фич
- Создание модели
- Предсказание

## 2 Получение фич

Считываем данные в `pandas.DataFrame`. Разобьём 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 не было таковых в обучающей, то выберем в этом случае среднее средних значений по всем значениям `'user.lang'` в обучающей выборке.

```
table = train_data[['user.lang', 'retweet_count']].groupby(by='user.lang').mean()
table = table.to_dict()['retweet_count']
```

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

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

**time zone:** Аналогично `lang`.

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

## 3 Создание модели

В качестве моделей возьмём `ensemble.RandomForestClassifier` и `ensemble.GradientBoostingClassifier`. Было желание попробовать `svm.SVC`, но за 2 часа работы всё это так и не отработало :с

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.