# Разметка наборов данных

### Описание

В данной работе рассматривается оригинальный необработанный набор Large Movie Review Dataset, а именно самая первая его версия, составленная 13 лет назад в 2011 году.

Основные причины выбора данного набора:

- оригинальный датасет
- необходимость скрейпинга и парсинга данных
- содержит 100 тысяч отзывов, 50 тысяч из которых вообще не имеют оценок
- практика работы с большими данными в label studio

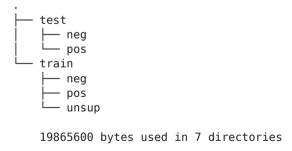
Архивный файл находится на странице, автором которой является Эндрю Маас, профессор Стенфордского университета.

В архивном файле, кроме сотни тысячи файлов с отзывами, присутствуют служебные файлы, представляющие научный интерес:

- README содержит описание от автора
- \*.vocab словарь, содержащий 89 526 английских слов
- \*.feat файлы в формате LIBSVM, содержащие матрицу векторов для каждого маркированного слова
- url.txt файлы со ссылками на оригинальный отзыв

Один файл это отзыв без маркировки (без оценки). Мне нужно обработать все сто тысяч, извлечь из каждого отзыв, присвоить ему нужный лейбл и объединить их, в зависимости от контекста, с нужным датасетом, в формате пригодным для импорта в jupyter-ноутбук.

Наборы данных имеют вложенную иерархию папок со следующей структурой:



В папке **test** находятся тестовые данные, которые разбиты еще на две подпапки **neg** и **pos**: негативные и позитивные отзывы.

В папке **train**, набор таких же папок, как и в test: **neg** и **pos**, там же расположена еще одна, представляющая для меня интерес, папка **unsup** с неразмеченными отзывами.

Всю работу от получения данных их обработки, разметки до описания выводов по машинному обучению, можно разбить на 15 этапов (задач)

# Задачи

- 1. установка и подключение необходимых библиотек
- 2. скрейпинг архива с оригинального сайта
- 3. распаковка архива
- 4. парсинг 100 тысяч маленьких файлов: чистка от мусора и лишних символов
- 5. rule-based labeling создание 5 файлов: test neg.txt, test pos.txt, train neg.txt, train pos.txt, unsup.txt
- 6. загрузка файлов в jupyter-блокнот с авторазметкой столбца label
- 7. перемешивание данных
- 8. обучение модели на тренировочном датасете
- 9. проверка на тестовом датасете
- 10. расчет эффективность модели
- 11. предсказание большого (неразмеченного) датасета
- 12. сохранение большого датасета в csv файл
- 13. загрузка большого датасета в label-studio
- 14. ручная проверка выборочных отзывов из большого датасета
- 15. выводы

## 1. установка и подключение необходимых библиотек

Раскомментируйте строку ниже в случае необходимости

```
In [1]: # pip install scikit-learn pandas numpy matplotlib requests tqdm
In [2]: from sklearn.feature extraction.text import TfidfVectorizer
        from sklearn.linear model import LogisticRegression
        from sklearn.model selection import train test split
        from sklearn.metrics import f1 score
        from sklearn.utils import shuffle
        from tqdm import tqdm
        import pandas as pd
        import numpy as np
        import requests
        import tarfile
        import glob
        import os
        import re
        import warnings
        warnings.filterwarnings('ignore')
```

### 2. скрейпинг архива

```
In [3]: url = 'https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz'
    response = requests.get(url, stream=True)
    total_size = int(response.headers.get('content-length', 0))
    block_size = 1024

with open('archive.tar.gz', 'wb') as file:
    with tqdm(total=total_size, unit='B', unit_scale=True, ncols=100) as pbar:
```

```
for data in response.iter_content(block_size):
    file.write(data)
    pbar.update(len(data))

100%| 84.1M/84.1M [00:31<00:00, 2.65MB/s]</pre>
```

## 3. распаковка архива

```
In [4]: with tarfile.open("archive.tar.gz", 'r:gz', errorlevel=0) as tar:
    members = tar.getmembers()
    for member in tqdm(members, desc='Extracting', unit='files', ncols=100):
        tar.extract(member)
Extracting: 100% | 100019/100019 [01:13<00:00, 1369.78files/s]
```

## 4. фунция парсинга файлов базы

## 5. rule-based labeling

```
In [6]: # 06ραδοτκα φάйπου ε παπκε aclImdb/train/pos process_text_files('aclImdb/train/pos/*.txt', 'train_pos.txt')

0%| | 10/12500 [00:00<02:05, 99.37file/s]100%| | 12500/12500 [02:19<00:00, 89.81file/s]

In [7]: # 06ραδοτκα φάйπου ε παπκε aclImdb/train/neg process_text_files('aclImdb/train/neg/*.txt', 'train_neg.txt')

100%| | 12500/12500 [02:13<00:00, 93.50file/s]

In [8]: # 06ραδοτκα φάйπου ε παπκε aclImdb/test/pos process_text_files('aclImdb/test/pos/*.txt', 'test_pos.txt')

100%| | 12500/12500 [02:12<00:00, 94.16file/s]

In [9]: # 06ραδοτκα φάйπου ε παπκε aclImdb/test/neg process_text_files('aclImdb/train/neg/*.txt', 'test_neg.txt')

100%| | 12500/12500 [00:43<00:00, 284.54file/s]
```

```
In [10]: # Обработка файлов в папке aclImdb/train/unsup
         process text files('aclImdb/train/unsup/*.txt', 'unsup.txt')
                       50000/50000 [17:43<00:00, 47.01file/s]
         6. загрузка файлов в jupyter-блокнот с авторазметкой столбца label
In [11]: test neg = pd.read csv('test neg.txt', engine='python',
                                   on bad lines='warn', header=None, names=['text', 'label'], sep =';', encoding='utf-8')
         test neg["label"] = 0
         test neg
Out[11]:
                                                       text label
              0 Story of a man who has unnatural feelings for ...
                                                               0
              1 Airport '77 starts as a brand new luxury 747 p...
                                                                0
              2 This film lacked something I couldn't put my f...
                                                               0
              3 Sorry everyone,,, I know this is supposed to b...
              4 When I was little my parents took me along to ...
                                                               0
```

12495 Towards the end of the movie, I felt it was to... 0
12496 This is the kind of movie that my enemies cont... 0
12497 I saw 'Descent' last night at the Stockholm Fi... 0
12498 Some films that you pick up for a pound turn o... 0

This is one of the dumbest films, I've ever se...

0

12500 rows × 2 columns

12499

```
Out[12]:
                                                                 text label
                 0
                      I went and saw this movie last night after bei...
                                                                           1
                       Actor turned director Bill Paxton follows up h...
                                                                           1
                 2 As a recreational golfer with some knowledge o...
                                                                           1
                        I saw this film in a sneak preview, and it is ...
                                                                           1
                 4
                       Bill Paxton has taken the true story of the 19...
                                                                           1
                ...
           12495
                       I was extraordinarily impressed by this film. ...
                                                                           1
           12496
                      Although I'm not a golf fan, I attended a snea...
                                                                           1
           12497
                      From the start of The Edge Of Love, the view...
                                                                           1
           12498
                                                                           1
                      This movie, with all its complexity and subtle...
           12499
                      I've seen this story before but my kids haven'...
                                                                           1
          12500 rows × 2 columns
```

```
text label
                                                               0
     0 Story of a man who has unnatural feelings for ...
     1 Airport '77 starts as a brand new luxury 747 p...
                                                               0
         This film lacked something I couldn't put my f...
                                                               0
                                                               0
          Sorry everyone,,, I know this is supposed to b...
     4 When I was little my parents took me along to ...
                                                               0
    • • • •
12495
           Towards the end of the movie, I felt it was to...
                                                               0
12496 This is the kind of movie that my enemies cont...
12497
          I saw 'Descent' last night at the Stockholm Fi...
                                                               0
12498
         Some films that you pick up for a pound turn o...
                                                               0
12499
          This is one of the dumbest films, I've ever se...
                                                               0
```

#### 12500 rows × 2 columns

Out[13]:

Out[14]:		text	label		
	Bromwell High is a cartoon comedy. It ran at t.  Homelessness (or Houselessness as George Carli				
	1	Homelessness (or Houselessness as George Carli	1		
	2	Brilliant over-acting by Lesley Ann Warren. Be	1		
	3	This is easily the most underrated film inn th	1		
	4	This is not the typical Mel Brooks film. It wa	1		
	12495	Seeing as the vote average was pretty low, and	1		
	12496	The plot had some wretched, unbelievable twist	1		
	12497	I am amazed at how this movie(and most others $\dots$	1		
	12498	A Christmas Together actually came before my t	1		
	12499	Working-class romantic drama from director Mar	1		
	12500 rd	ows × 2 columns			

Out[15]:		text	label		
	0	I admit, the great majority of films released $\dots$	NaN		
	<b>1</b> Take a low budget, inexperienced actors doubli				
	<b>2</b> Everybody has seen 'Back To The Future,' right				
	<b>3</b> Doris Day was an icon of beauty in singing and		NaN		
	<b>4</b> After a series of silly, fun-loving movies, 19				
•••					
	49995 Delightfully awful! Made by David Giancola, a .				
	49996	Watching Time Chasers, it obvious that it was $\dots$	NaN		
	49997	At the beginning we can see members of Troma $t$	NaN		
	49998	The movie was incredible, ever since I saw it $\dots$	NaN		
	49999	TCM came through by acquiring this wonderful, $\dots$	NaN		

#### 50000 rows × 2 columns

# 7. перемешивание данных

```
In [16]: test = pd.concat([test_neg, test_pos], ignore_index=True)
    test = shuffle(test)
    test
```

Out[16]:		text	label
	21349	Lifeforce starts in outer space where the HMS $\dots$	1
	0		
	18517	All the characters in this cartoon were hilari	1
	1940	Although it got some favorable press after pla	0
	2974	The British claymation series putting witty $\dots$	0
	10330	Suppose you've been on a deserted island the l $$	0
	9918	This is a badly made, poor remake of Bimalda's	0
	23950	I usually steer clear of Film Festivals and do	1
	10027	What a load of rubbish I can't even begin to	0
	12055	Very strange but occasionally elegant exploita	0

#### 25000 rows × 2 columns

```
In [17]: train = pd.concat([train_neg, train_pos], ignore_index=True)
    train = shuffle(train)
    train
```

Out[17]:		text	label		
	4762	This movie was absolutely ghastly! I cannot fa	0		
	22652	The Movie Freddy's dead the final nightmare is	1		
18761 14205	This is one powerful film. The first time I sa	1			
	14205	Bill Crain's rarer than rare 'slasher' movie c	1		
	<b>16841</b> Kubrick again puts on display his stunning abi				
	<b></b>				
	2850 Oliver Hardy awakens with a hangover and soon		0		
	12838	Bruce Almighty looks and sounds incredibly s	1		
	12378	Why is it that any film about Cleopatra, the l $$	0		
	7541	This is not really a zombie film, if we're def	0		
	19188	The only show I have watched since 90210! Why $\dots$	1		

25000 rows × 2 columns

# 8. обучение модели на тренировочном датасете

```
In [18]: def train_model(label):
    vectorizer = TfidfVectorizer()
    X = vectorizer.fit_transform(train['text'])
    y = label['label']
```

```
model.fit(X, y)
               return model, vectorizer
In [19]: model, vectorizer = train model(train)
          9. проверка на тестовом датасете
In [20]: x test = vectorizer.transform(test['text'])
          y test = model.predict(x test)
In [21]: test['predicted'] = y test
          test.head()
Out[21]:
                                                          text label predicted
          21349 Lifeforce starts in outer space where the HMS ...
                                                                               0
           2697
                   I loved the first two movies, but this movie w...
                                                                               0
          18517
                     All the characters in this cartoon were hilari...
                                                                   1
                                                                               1
                                                                               0
           1940
                  Although it got some favorable press after pla...
                                                                               0
           2974
                     The British claymation series putting witty ...
                                                                   0
          10. эффективность модели
In [22]: test['loss'] = test['label'] ^ test['predicted']
          test
                                                           text label predicted loss
Out[22]:
          21349
                   Lifeforce starts in outer space where the HMS ...
                                                                     1
                                                                                 0
                                                                                      1
           2697
                    I loved the first two movies, but this movie w...
          18517
                      All the characters in this cartoon were hilari...
                                                                     1
                                                                                1
                                                                                       0
                                                                     0
           1940
                    Although it got some favorable press after pla...
                                                                                0
                                                                     0
                                                                                0
           2974
                      The British claymation series putting witty ...
                                                                                      0
          10330
                  Suppose you've been on a deserted island the I...
                                                                     0
                                                                                0
                                                                                      0
                                                                     0
           9918 This is a badly made, poor remake of Bimalda's...
          23950
                                                                     1
                                                                                1
                      I usually steer clear of Film Festivals and do...
          10027
                    What a load of rubbish.. I can't even begin to...
                                                                     0
                                                                     0
                                                                                0
                                                                                      0
          12055
                   Very strange but occasionally elegant exploita...
         25000 rows × 4 columns
```

model = LogisticRegression()

In [23]: guess, loss = test['loss'].value counts()

```
In [24]: print(f"Предсказано: {guess / (loss+guess) * 100} %")
Предсказано: 90.572 %

In [25]: f1 = f1_score(test["label"], y_test)

In [26]: print(f"Метрика эффективности F1: {f1}")
Метрика эффективности F1: 0.9033976802327964
```

## 11. предсказание большого (неразмеченного) датасета

	text	iabei	predicted
0	I admit, the great majority of films released	NaN	1
1	Take a low budget, inexperienced actors doubli	NaN	0
2	Everybody has seen 'Back To The Future,' right	NaN	0
3	Doris Day was an icon of beauty in singing and	NaN	1
4	After a series of silly, fun-loving movies, 19	NaN	1
49995	Delightfully awful! Made by David Giancola, a	NaN	1
49996	Watching Time Chasers, it obvious that it was $\dots$	NaN	0
49997	At the beginning we can see members of Troma $t$	NaN	0
49998	The movie was incredible, ever since I saw it $\dots$	NaN	1
49999	TCM came through by acquiring this wonderful,	NaN	1

 $50000 \text{ rows} \times 3 \text{ columns}$ 

## 12. сохранение большого датасета в табулированный сѕу файл

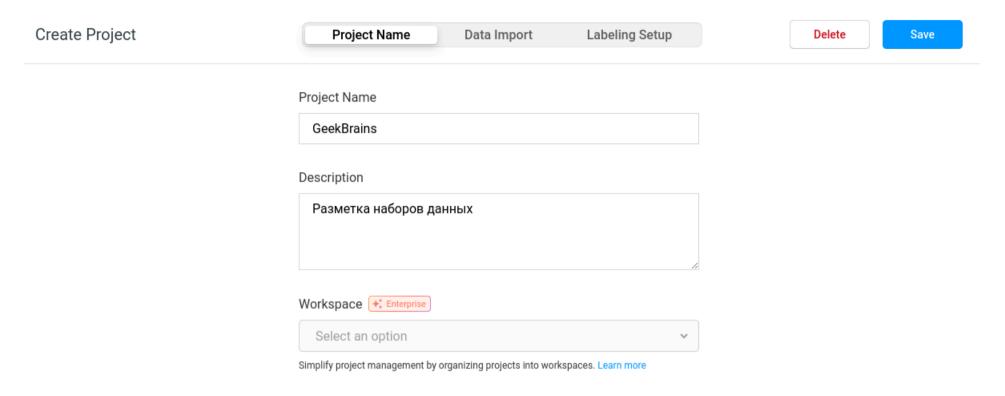
```
In [29]: unsup.to csv('unsup.tsv', index=False, encoding='utf-8', sep='\t')
```

## 13. загрузка большого датасета в label-studio

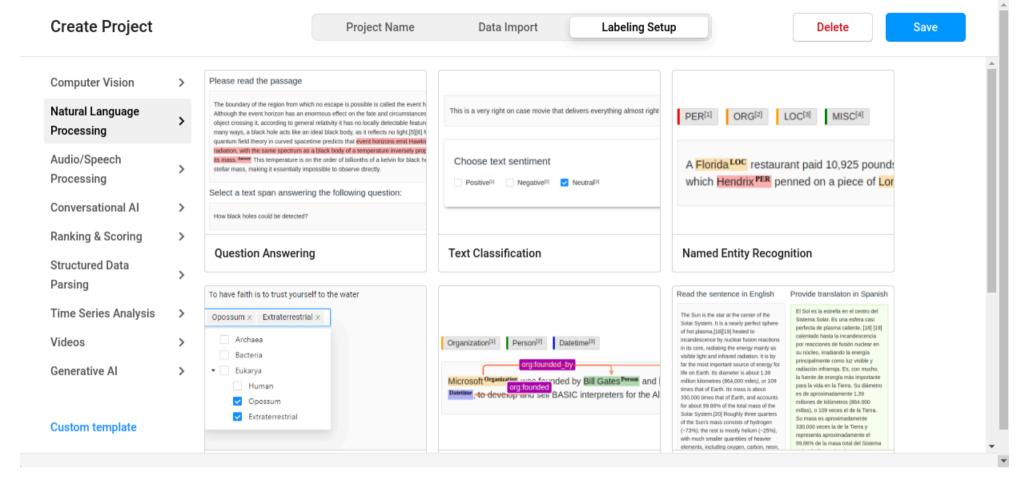
Загрузка огромного CSV файла 68 Мегабайт, с 50 тысячами строк, оказалось не простым делом для label studio (LS).

Постоянно вываливались ошибки, о большом количестве SQL данных. Т.к. под капотом LS переводит полученную информацию в json-формат и умеет работать с SQL, то в огромном массиве английских слов встречаются их комбинации со спецсимволами, которые LS воспринимает как служебные инструкции для баз данных.

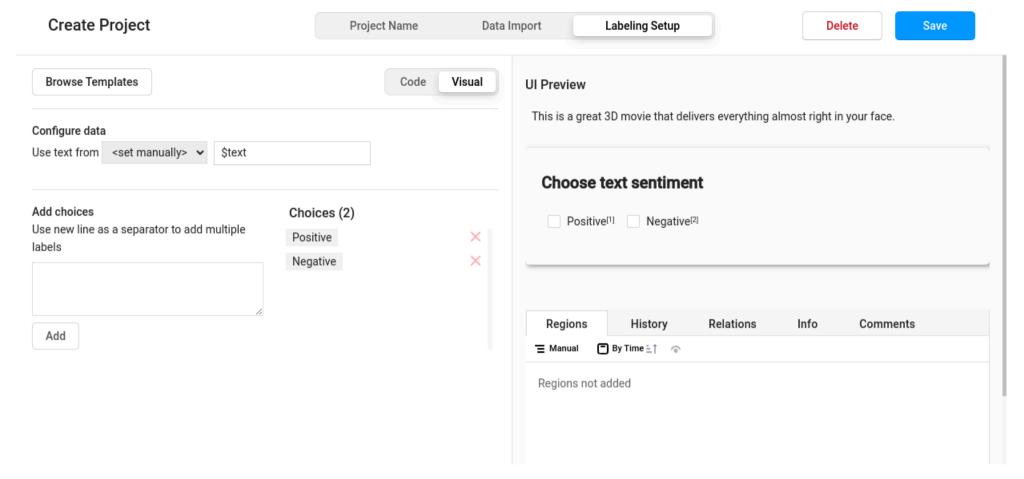
Удалось победить данную проблему с помощью табулированного формата CSV или как его обозначают TSV. Ошибки все равно появлялись, но данные полностью загрузились и с ними можно работать.



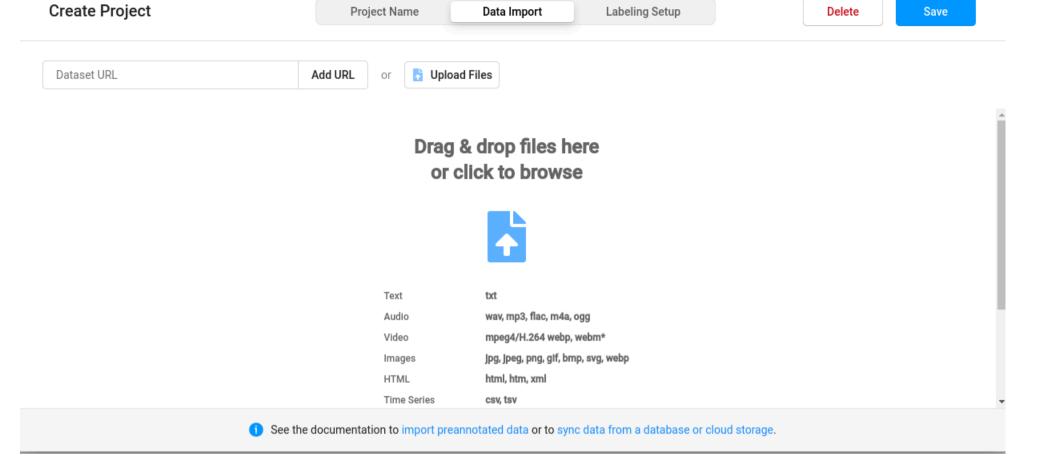
После создания проекта LS, необходимо выбрать тип данных для маркировки:



Далее, задаем два лейбла для маркировки: Positive, Negative

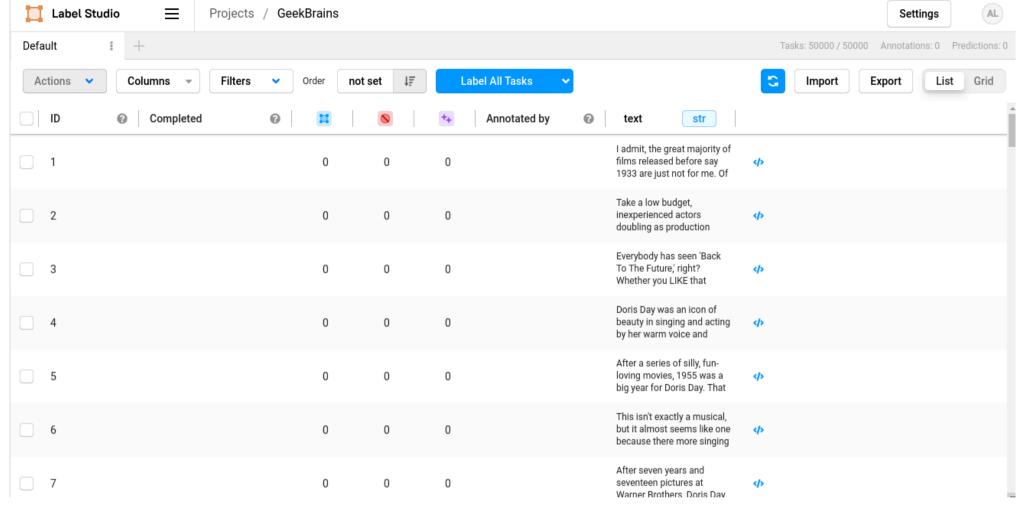


Выбираем TSV файл и импортируем его в проект LS:

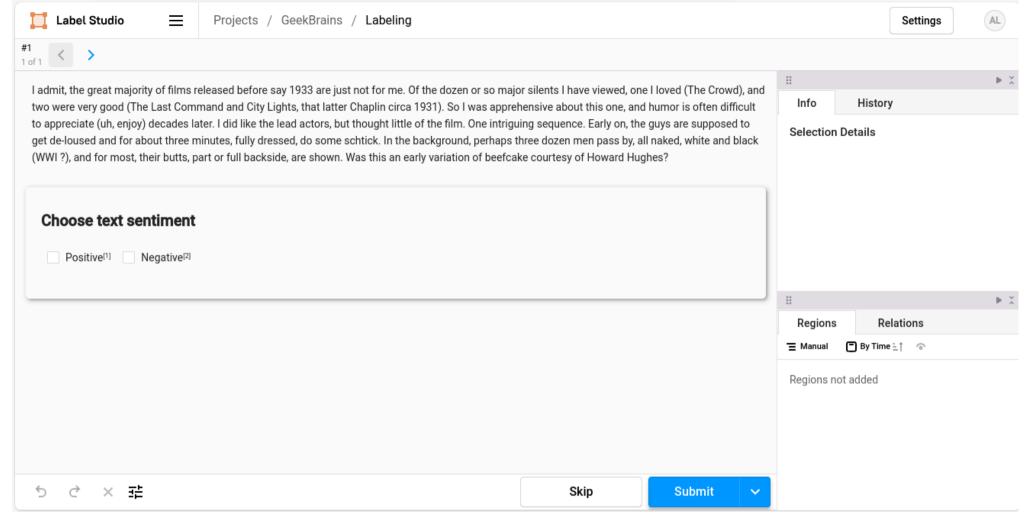


# 14. ручная проверка выборочных отзывов

Загруженные данные в LS представлены в табличном виде



После нажатия на кнопку **Label All Tasks** запустится процесс ручной разметки с помощью "карточек" с отзывами и возможность присвоить нужный лейбл:

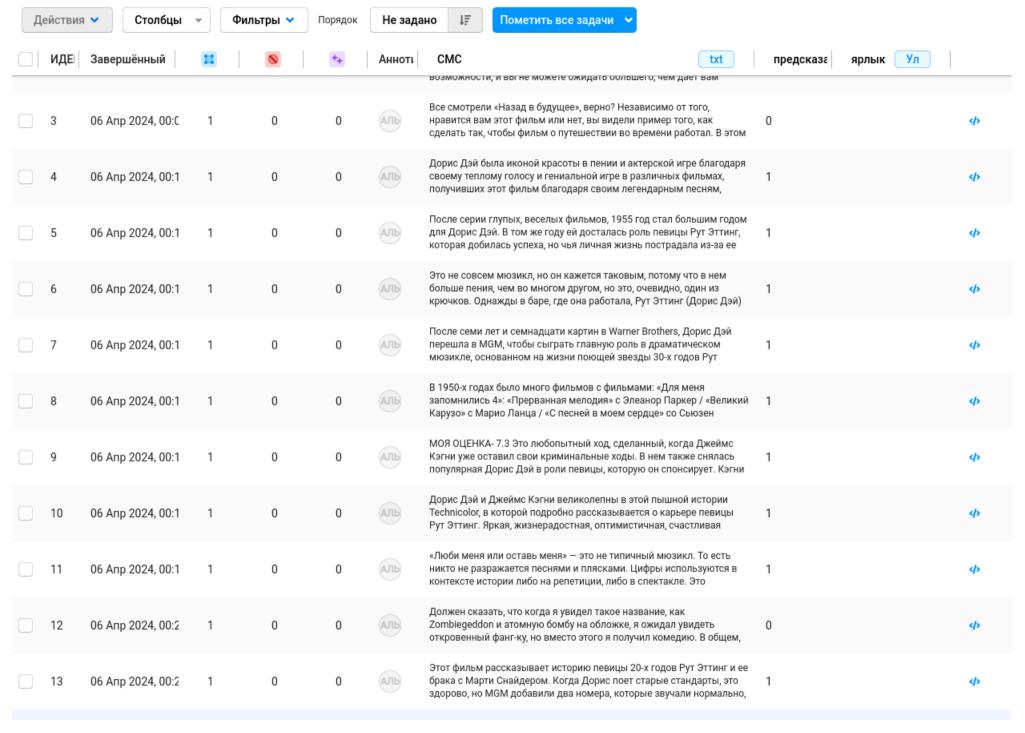


Благодаря развитию нейронных сетей, машинный перевод с иностранных языков позволяет получить приемлемый русский текст за доли секунд. Встроенная функция перевода в интернет браузере:

Это не совсем мюзикл, но он кажется таковым, потому что в нем больше пения, чем во многом другом, но это, очевидно, один из крючков. Однажды в баре, где она работала, Рут Эттинг (Дорис Дэй) познакомилась с продюсером Мартином Снайдером (номинант на премию «Оскар» Джеймс Кэгни), который мог пообещать новую и более высокооплачиваемую карьеру на сцене. Она не сразу приступает к делу, так как она включена в течение нескольких секунд, но вскоре у нее появляется шанс на более длительные шоу. Она наслаждается собой, а критики и продюсеры забрасывают ее предложениями. Позже, после свадьбы, Рут и Мартин становятся еще больше, и она становится одной из самых успешных женщин на сцене, а ее карьера вскоре приходит в Голливуд. Только она обнаруживает, что над фильмом работает старый друг (почти любовь), Джонни Олдермен (Кэмерон Митчелл), что для нее и брака/отношений с Мартином все идет немного не так. Также в главных ролях Роберт Кит в роли Бернарда В. Лумиса, Том Талли в роли Фробишера и Гарри Беллавер в роли Джорджи. Он получил «Оскар» за лучший сценарий к фильму, а также был номинирован на «Лучшую песню» за «I'll Never Stop Loving You», «Лучшую музыку» для Перси Фейта и Джорджа Э. Столла, «Лучший звук» и «Лучший сценарий». Дорис Дэй заняла 84-е место в списке 100 величайших кинозвезд, а Джеймс Кэгни занял 8-е место в списке 100 лет, 100 звезд. Очень хорошо!

Choose text sentiment		
Positive <sup>[1]</sup> Negative <sup>[2]</sup>		

После окончания ручной выборочной разметки получаем таблицу, в которой виден наш лейб и лейбл расставленный с помощью машинного обучения.



### 15. выводы

Прежде всего, хочу отметить монотонную сложность выставления лейблов вручную.

Каждый отзыв нужно прочитать, осмыслить и вынести свой вердикт. Может уйти несколько минут на обработку одного отзыва.

В результате, некоторые мои лейблы, отличались от лейблов выставленных машиной.

После изучения таких разногласий, перечитывал еще раз конкретный отзыв и к своему ужасу осознавал, что машина права! Т.е. человек хуже справляется с задачами разметки эмоциональной окраски, в данном случае, не говоря уже о скорости.

По грубым расчетам, для маркировки 50 тысяч записей (по 5 минут на каждую) может понадобиться 4167 человеко-часов работы или 173 суток. Если учесть восьми часовой рабочий день, два выходных дня в недели и отпуск раз в году, то специалист за год сможет осилить 2080 отзывов. Т.е. на конкретный большой датасет из нашего примера у человека уйдет два года. Машина это сделала за несколько секунд.

В результате те, кто сделал ставку на машинное обучение и нейросети 15 лет назад, стали очень богатыми людьми. Например, как автор данного набора Эндрю Маас, попавший в журнал Форбс.