Music Recommender System

data science minor final project
making recommenders uncool again

Group 16: E.Skriptsova
E.Voronova

план

1. Команда ~~и немного о том, как мы дошли до такой жизни~~
2. Идея
3. Данные
4. Задачи, которые мы решаем
5. Ресерч ~~или то, как страдали над этим другие~~
6. Как ~~страдали~~ это сделали и делаем мы
7. Что же уже получилось и получилось ли? ~~(да)~~
8. Интерфейс~~, пользователи и вот это вот всё~~
9. Что дальше?

#команда

команда

Катя

Главный по тарелочкам
Мэйн датер
Делает всё круто
За сотрудничеством
обращаться к менеджеру
(Лизе)

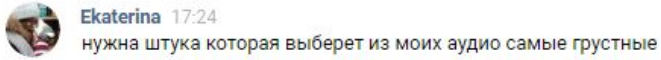
Лиза

Дезигнер от бога
Может быть, почистит данные
(если разберётся)

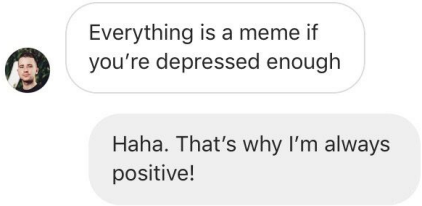
#идея

идея | что нас вдохновило?

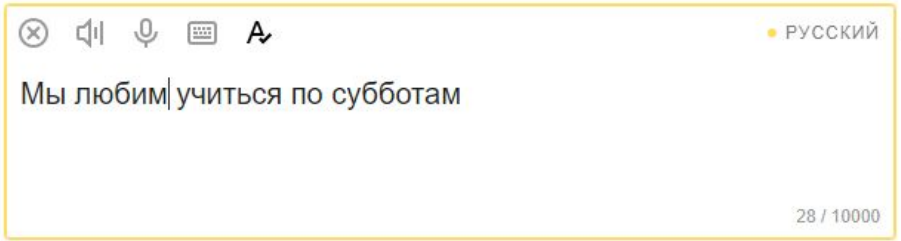
Как это всегда бывает, холодным зимним вечером...

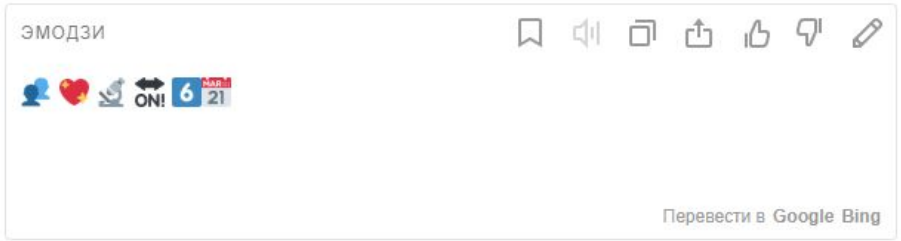


И мы подумали... А почему бы нам не сделать рекомендательную систему, которая будет предлагать пользователю музыку, исходя ещё и из его настроения?



А потом мы вспомнили про Яндекс и переводчик с emoji, и решили, что пользовательский ввод будет именно такой, ведь смайлик выбрать куда проще (но вообще мы еще думаем)



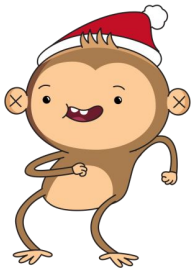


#данные

данные | музыка & тексты

Плейлисты:

- из одной известной нам **социальной сети**
- **950K** пользователей
- **90M** взаимодействий user-item
- 5GB, wow so big much data
- собирали не мы



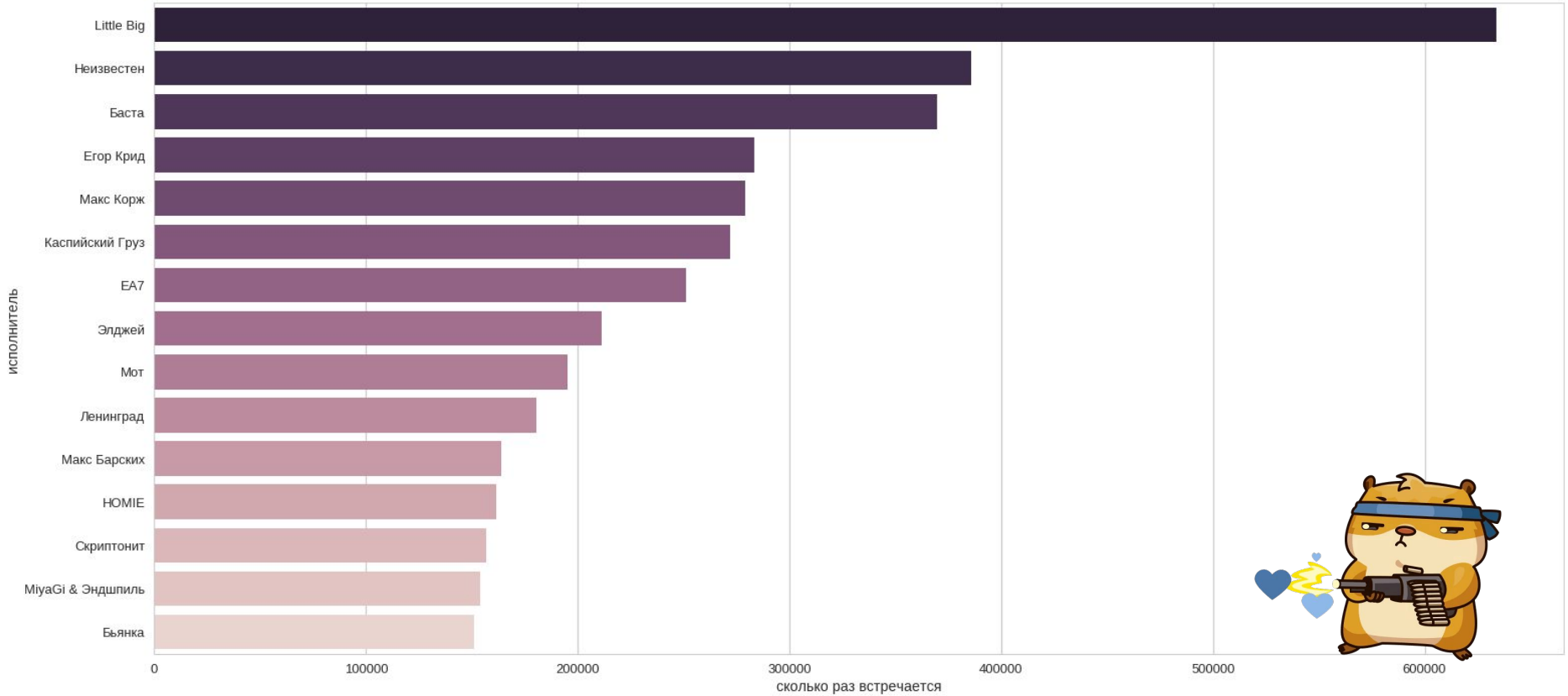
Тексты:

- собираем сами для каждой песни у которой есть текст и которая есть в базе genius
- **n** текстов песен, *n = много*
- сегодня картинок по текстам не будет :c

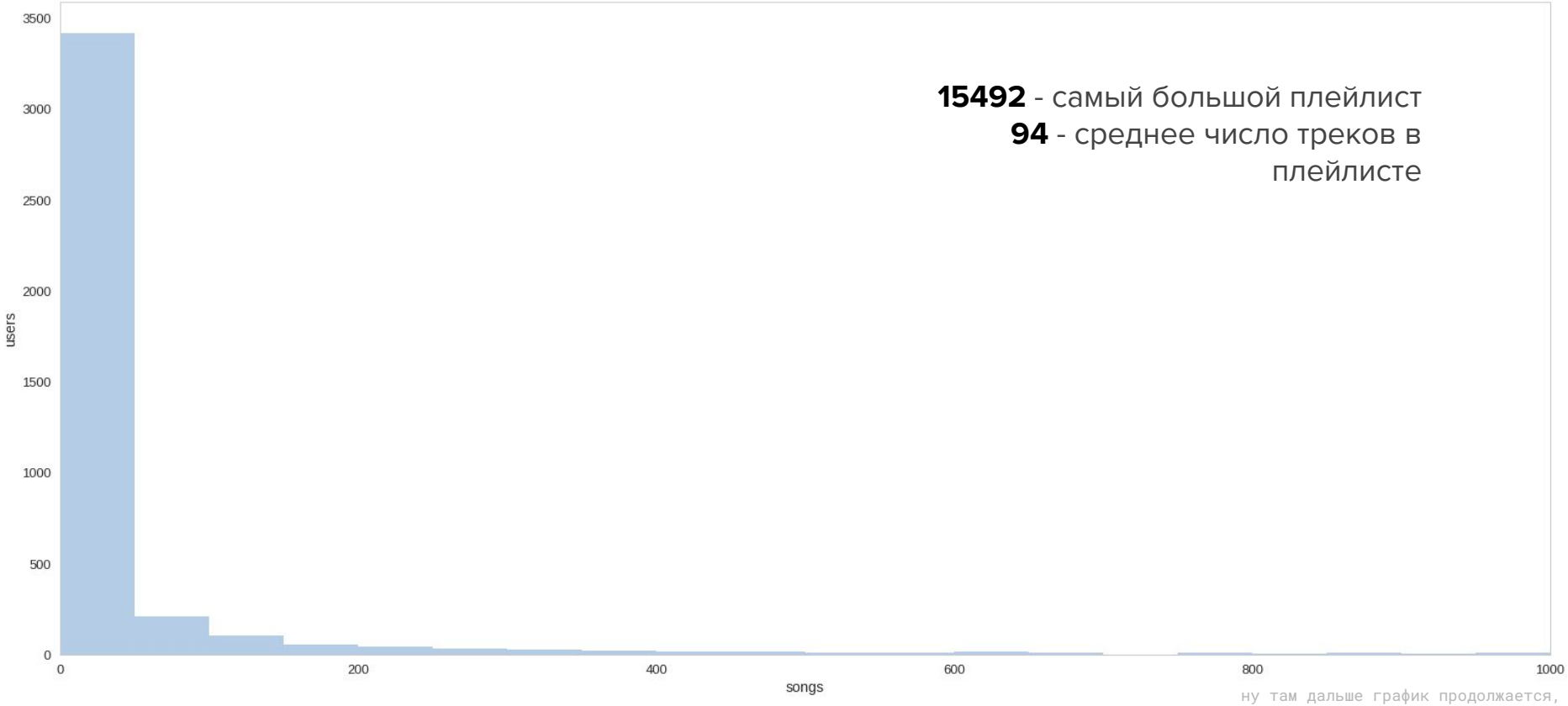
GENIUS

*K-тысяч
M-миллионов, да

данные | исполнители



данные | а что с пользователями?



15492 - самый большой плейлист
94 - среднее число треков в плейлисте

ну там дальше график продолжается,
а то иначе ничего не видно

данные | треки

нет, ну вы же не думали, что мы вам тут сразу всё покажем?

not today, not today

данные | beyonce béyonce beyoncé

Многие треки добавляются пользователями, поэтому и названия исполнителей бывают разными… и это мы ещё риа...rhi..riha…. не смотрели

-> в дальнейшем надо бы почистить данные

#задачи

цель| над чем и как страдаем

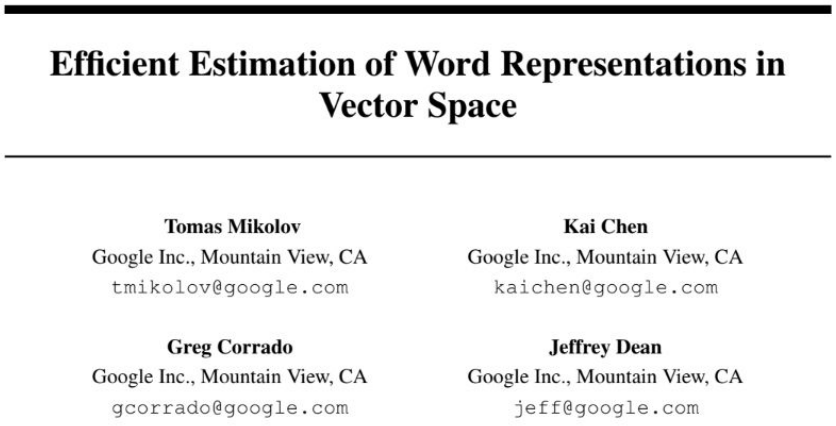
Наша основная и самая глобальная ***цель*** - создать рекомендательную систему, которая будет работать так, как мы изначально задумали :)

В итоге:

1. Собрать данные (плейлисты done/тексты almost)
2. А еще очень весело всё это обрабатывать (убрать треш, исправить опечатки)
3. Выделить настроения из текстов
4. Соединить вот это вот всё
5. Интерфейс

#ресерч

Ekaterina 3:34
you: good night sleep
me, an intellectual: time to read ml papers



Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

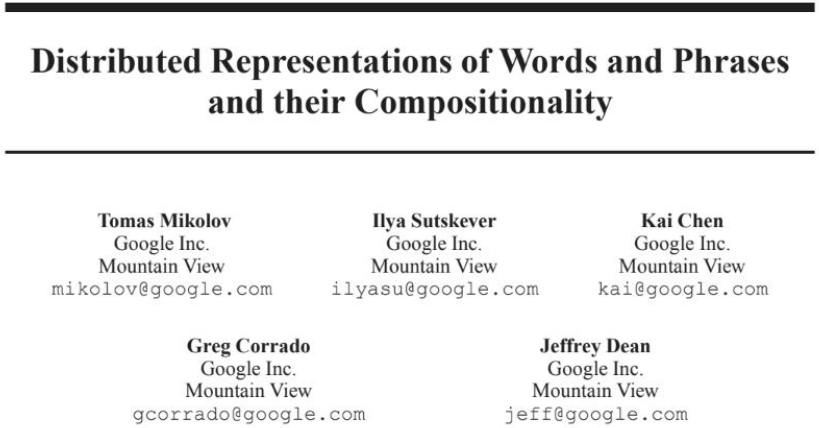
Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

word2vec — это набор алгоритмов для расчета векторных представлений слов

- Continuous Bag of Words (CBOW)
- Skip-gram

На вход подается корпус текста, а на выходе получается набор векторов слов.
Много хороших данных - прикольные результаты

Вывод: ничего не зная про сами слова можем много чего узнать



Distributed Representations of Words and Phrases and their Compositionality

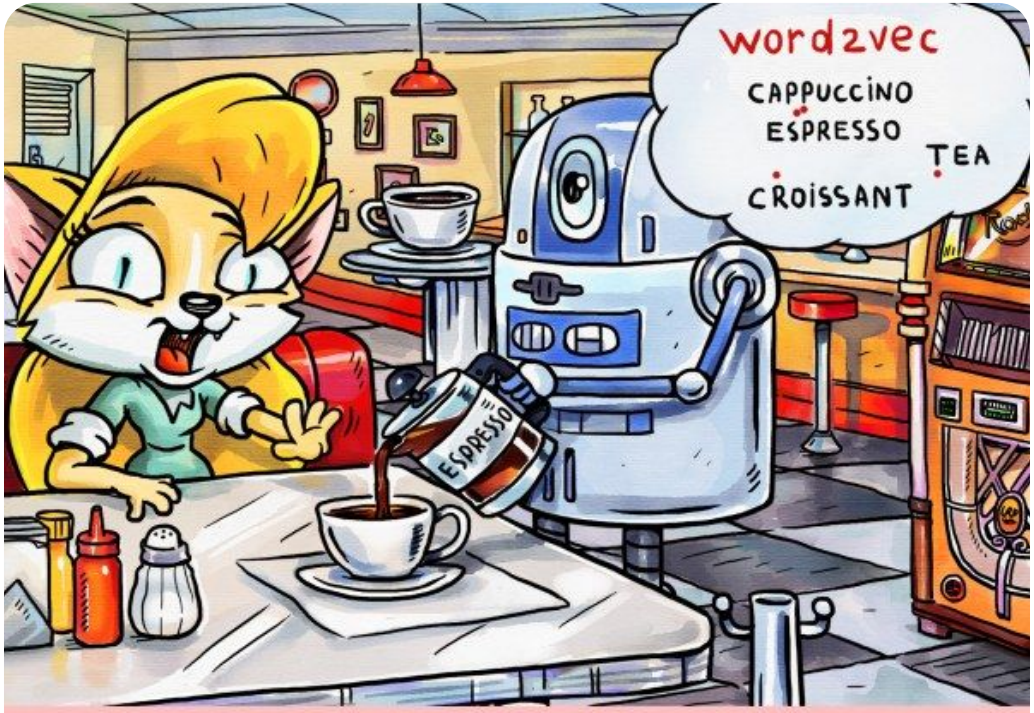
Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

Kai Chen
Google Inc.
Mountain View
kai@google.com

Greg Corrado
Google Inc.
Mountain View
gcorrado@google.com

Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

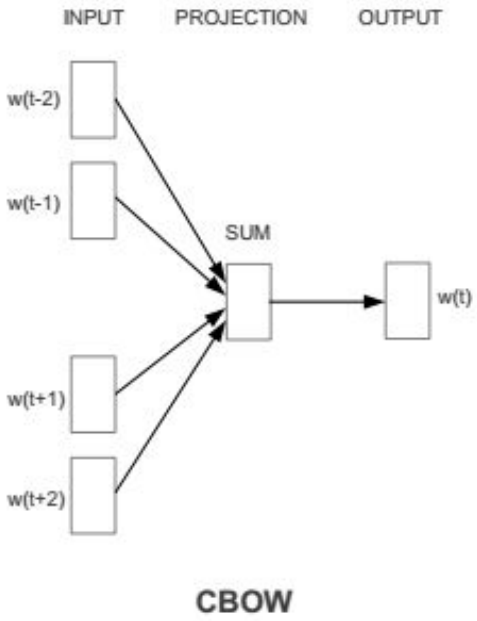


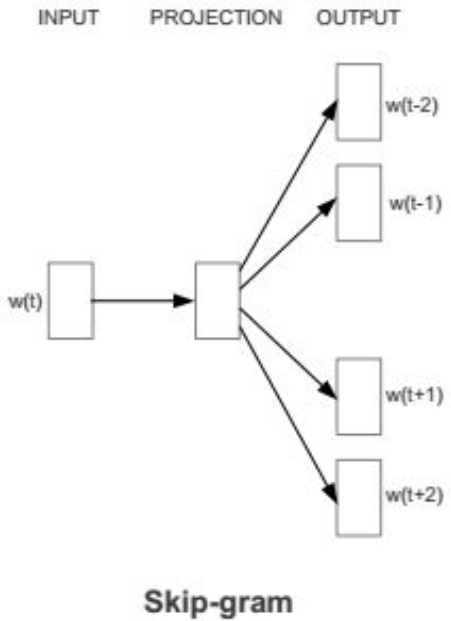
- Espresso? But I ordered a cappuccino!
- Don't worry, the cosine distance between them is so small that they are almost the same thing.

ресерч | word2vec

Принцип работы:

- CBOW: предсказывание слова при данном контексте
- Skip-gram: предсказывается контекст при данном слове





Continuous Bag of Words (CBOW) – обычная модель мешка слов с учётом четырёх ближайших соседей термина (два предыдущих и два последующих слова)

k-skip-n-gram — это последовательность длиной n, где элементы находятся на расстоянии не более, чем k друг от друга

ресерч | word2vec

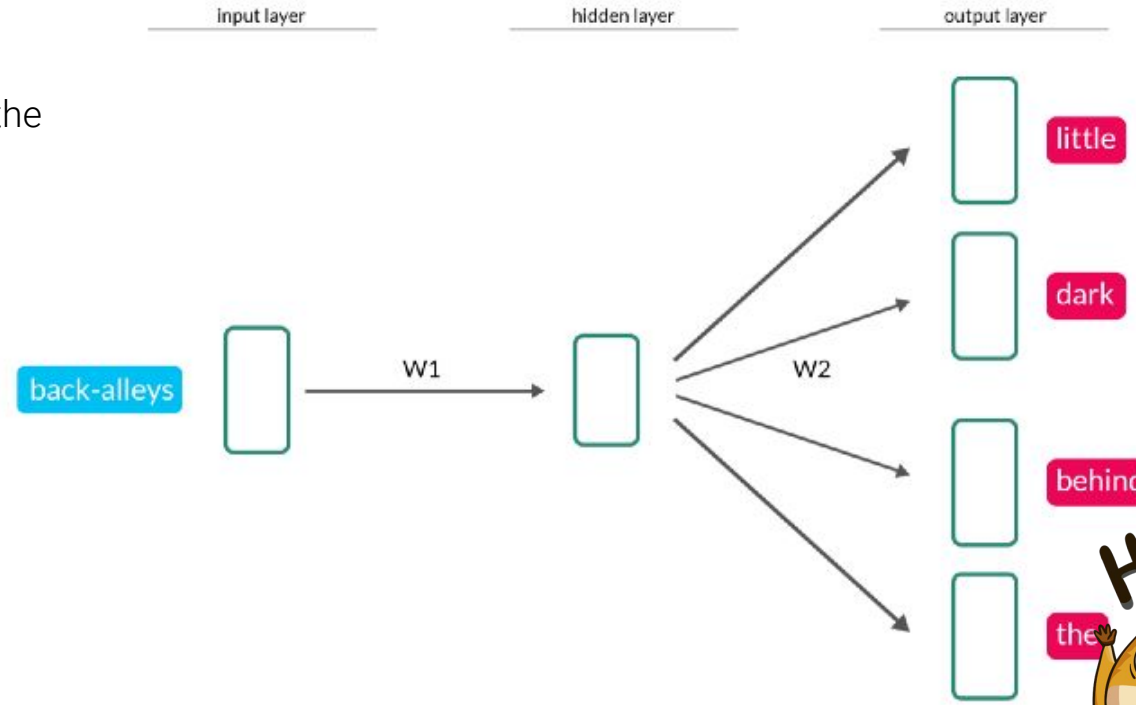
1. получаем на вход слово
2. пытаемся предсказать контекст

word2vec | skip-gram model

input: black-alleys

output: little, dark, behind, the

(то, что мы хотим предсказать)

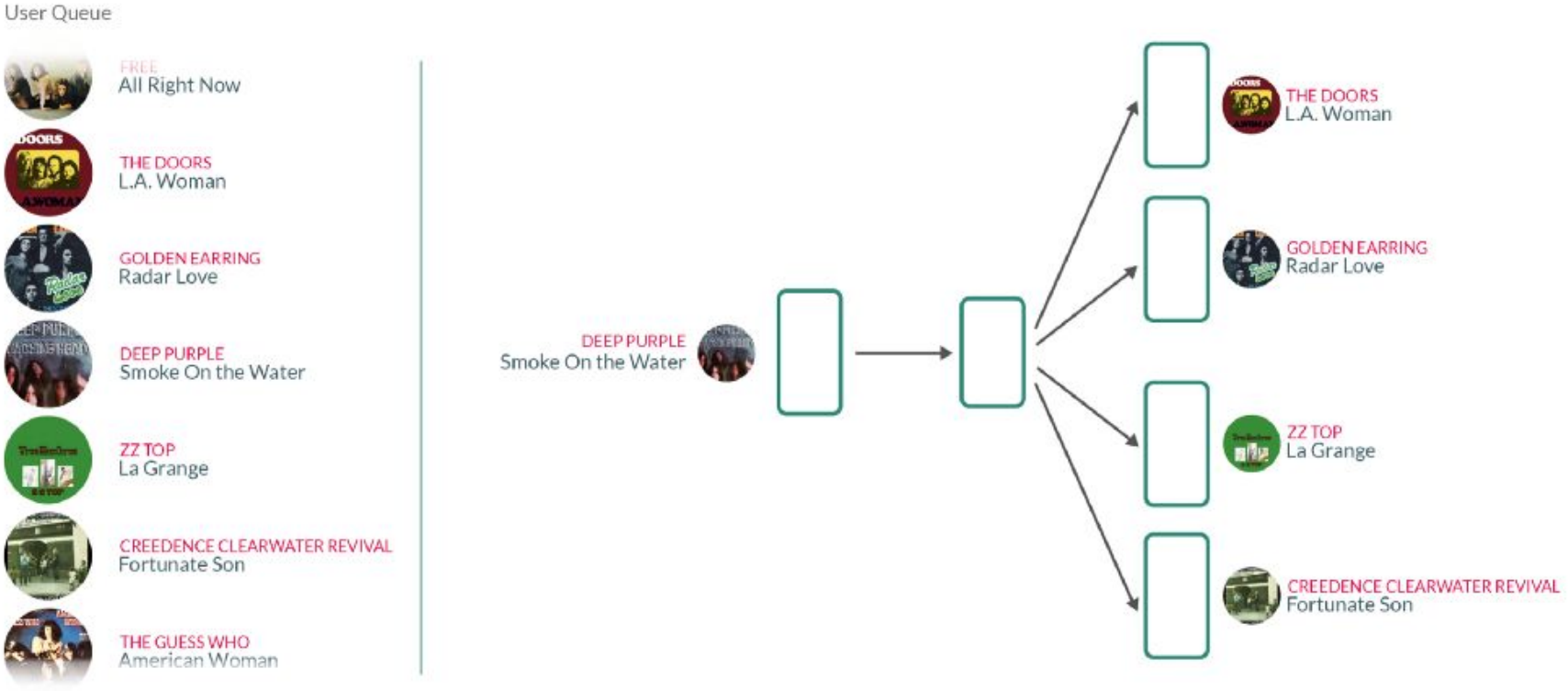


так, но ведь у нас тут не текст, а вроде как музыка..?

word2vec | Skip-gram model

Всё просто: по песне из плейлиста угадываем ее соседей в этом же плейлисте

В итоге можем рекомендовать новые песни или исполнителей



#поравпродакшн?

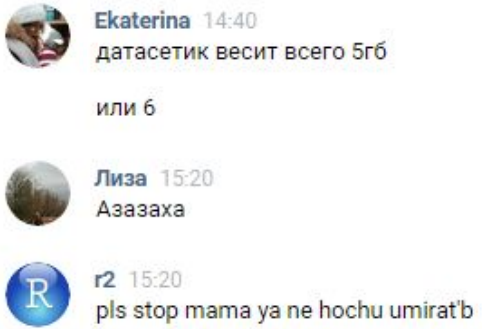
методы |  vs. 

Обработка данных: python

Графики: python + R ~~(на самом деле нет)~~

Вот эти вот ml штучки: python

Интерфейс: Shiny



model | word2vec

1. взяли вордтувек
2. обучили только на части данных (10M, ~42min)
   потом обучим целиком (90M, за это время катя успеет поспать)

```
> that was fun
```

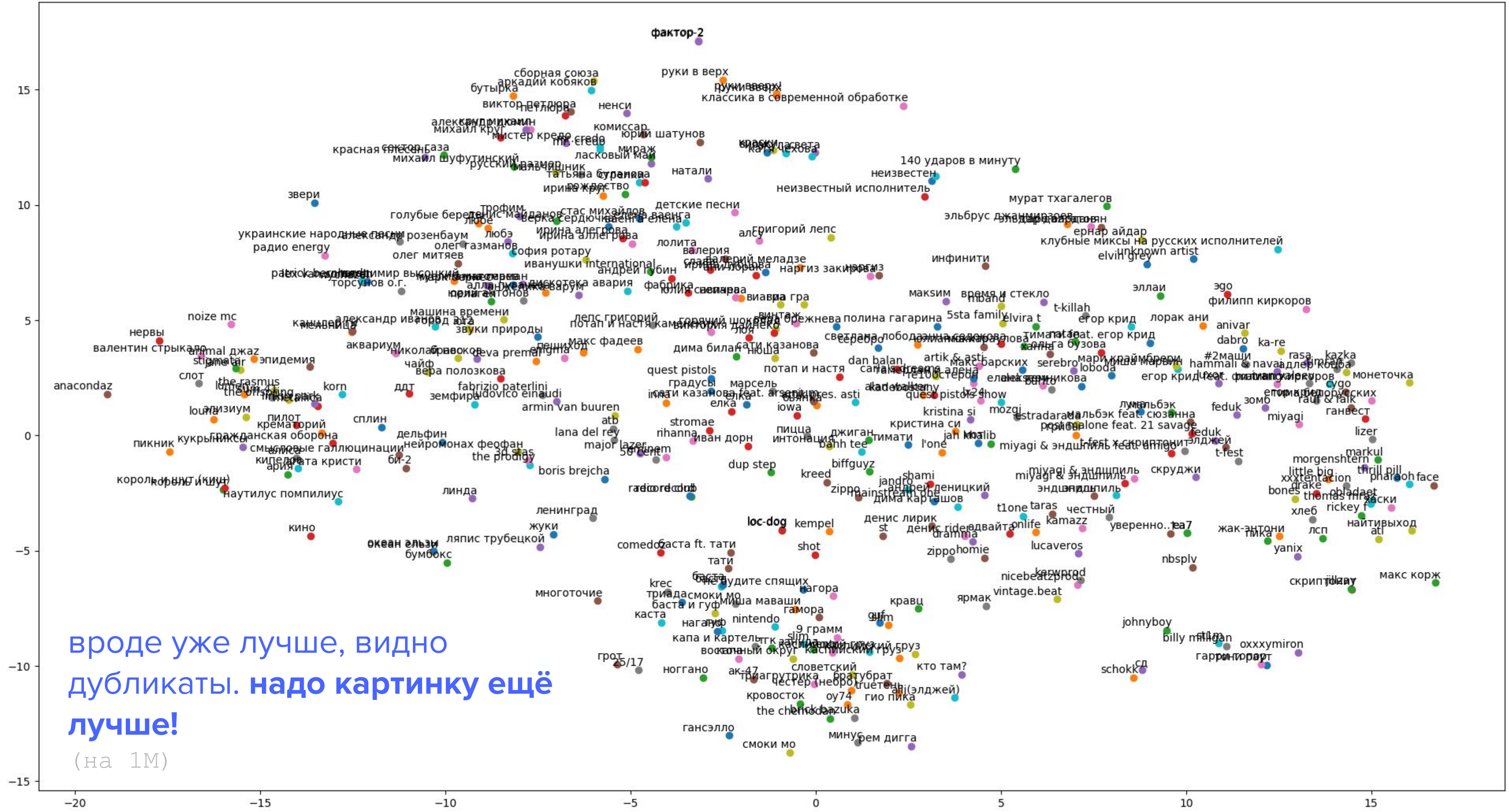
- пока что никак не учитываем настроение и вот это вот всё
- ещё подумаем как совместить

когда вспоминаешь как строилась модель

t-SNE | T-distributed Stochastic Neighbor Embedding

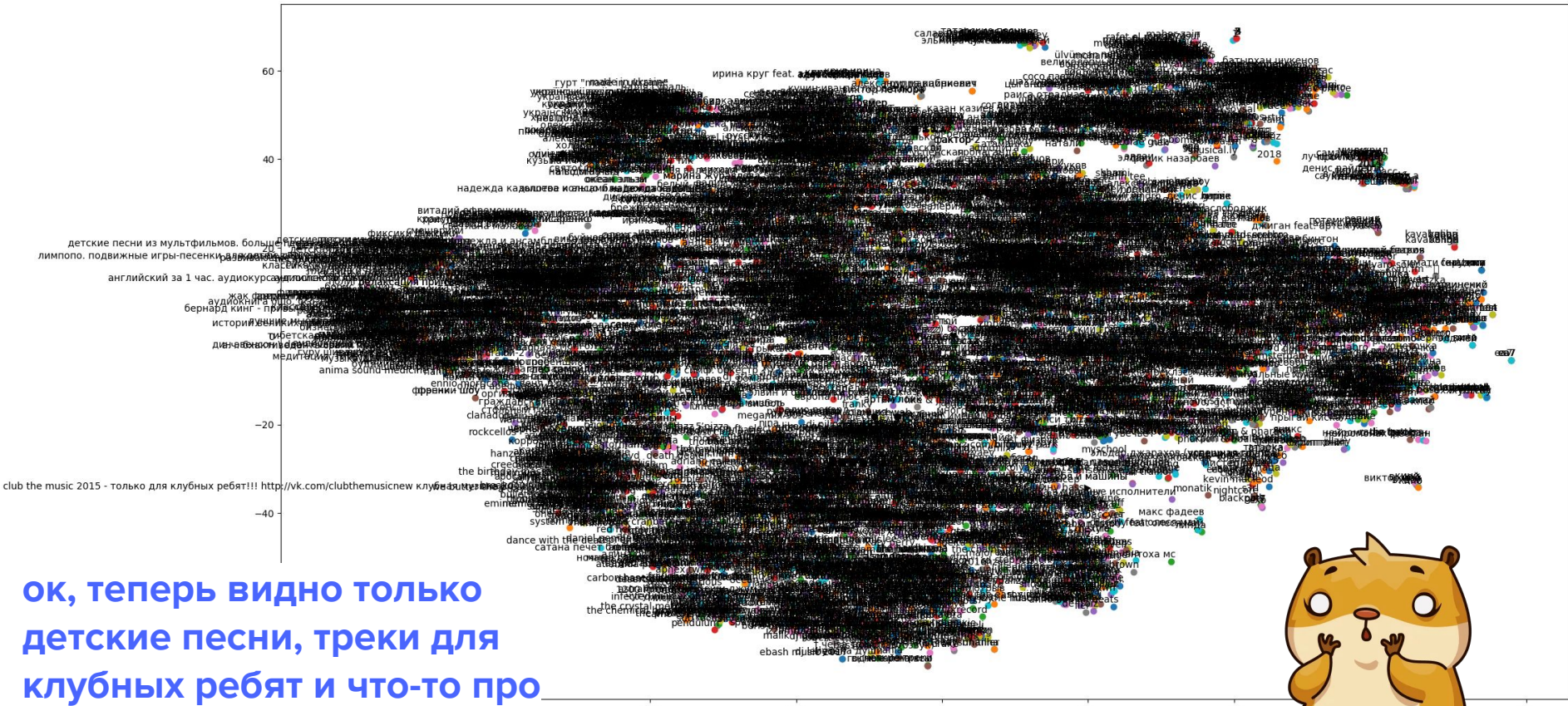
Получается многомерное пространство -> снижаем размерность пространства

Каждая точка - исполнитель.
Чем ближе расположены, тем более похожи
(на 100k)



вроде уже лучше, видно дубликаты. **надо картинку ещё лучше!**

(на 1M)



ок, теперь видно только детские песни, треки для клубных ребят и что-то про сатану??

(на 10М)

модель | хорошо работает!

получилось прикольно, но есть одно но:
в дальнейшем стоит разобраться с разными написаниями одних и тех же исполнителей

```
3 [a[0].strip() for a in model_w2v.wv.most_similar('beyonce', topn=10)]
```

```
['rihanna',
 'beyoncé',
 'christina aguilera',
 'justin timberlake',
 'the weeknd',
 'sam smith',
 'ciara',
 'miguel',
 'jeremih',
 'mary j. blige']
```

```
1 model_w2v.wv.most_similar('max richter', topn=10)
```

```
[('hans zimmer', 0.6437892913818359),
 ('tim hecker', 0.6351954936981201),
 ('tom waits', 0.6164076328277588),
 ('ludovico einaudi', 0.6063662171363831),
 ('the cinematic orchestra', 0.6058288812637329),
 ('james blake', 0.6048403382301331),
 ('iamx', 0.6029541492462158),
 ('bushwacker', 0.600149393081665),
 ('ulver', 0.5972044467926025),
 ('radiohead', 0.5968706607818604)]
```

модель обученная на 10М

модель | бывает странно \(o_o)/

в ходе препроцессинга встречались смешные композиции, и мы решили посмотреть что для этого рекомендуется.

`that was even funnier.`

```
user_music = ['эротический саксофон']
```

```
['саксофон',
 'звуки природы для детей - мой океан',
 'эротическая музыка',
 'ренди гейдж',
 'origen',
 'классика в современной обработке',
 'раймонд паулс',
 'саксофон',
 'егор денисевич',
 'медитации для женщин',
 '5. саксофон для влюбленных',
 'francis goya',
 'труба',
 'удивительный саксофон',
 'квітка цісик',
 'candee soulchillaz',
 'bellydance']
```

#интерфейс

примерно так выглядела Лиза после размышлений о дизайне в три часа ночи

интерфейс | чего как выглядеть будет

Пользовательский ввод:

- выбор одного из эмодзи
- ввод любимых исполнителей/мб треков
  (вып. список??)

Вывод:

- то, что моделька решит (topn рекомендаций)

Челлендж:

- подружить питон и ppp
- но мы уже подружили

что дальше? (; - ;)

1. Собрать тексты
2. Научиться детектить настроение
3. Дополнить модель
4. Как-то нормально оценить качество
5. Интерфейс
6. ???

That's all Folks!

questions?