ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Московский институт электроники и математики
им. А.Н. Тихонова**

Студенты учебной группы МСКМ-181:
Горчавкина Анастасия Александровна
Пехтерев Денис Олегович
Чертенков Владислав Игоревич

Техническая документация

Анализ отзывов о лекарственных препаратах в социальных медиа

Научный руководитель к.ф-м.н., доцент
Артамонов С.Ю.

Москва 2019

1 Техническое задание

Проект относится к проблеме анализа информации, оставленной пользователями в Интернете о лекарственных препаратах и медицинской сфере в целом.

Цель проекта состоит в применении современных средств прикладной математики для анализа данных.

Основные задачи проекта:

- Определение тональности отзывов методами машинного обучения
- Многоклассовая классификация отзывов
- Подбор гиперпараметров для эффективной реализации методов
- Визуализация данных на платформе Tableau Public

Для выполнения задач использованы следующие методы:

- Для определения тональности:
  - Метод логистической регрессии
- Для многоклассовой классификации:
  - Random Forest
  - Метод опорных векторов (SVM)
  - Градиентный бустинг над решающими деревьями

Получены следующие результаты:

- Визуализация данных на платформе Tableau Public (Горчавкина А.А.)
- Разработана модель машинного обучения по определению тональности отзывов (Пехтерев Д.О.)
- Реализована модель многоклассовой классификации, использующая три метода:
  - Random Forest
  - Метод опорных векторов (SVM)
  - Градиентный бустинг над решающими деревьями (Чертенков В.И.)

2 Общее описание системы

2.1 Многоклассовая классификация:

В Python-библиотеке sklearn.svm реализован метод опорных векторов. Функция SVC позволяет создавать и задавать параметры классификатора. Она также поддерживает метод многоклассовой классификации и подходы «Один против всех» и «Каждый против каждого». В Python-библиотеке sklearn.ensemble реализован метод классификации случайный лес. Функция RandomForestClassifier позволяет создавать и задавать параметры классификатора. Она также поддерживает метод многоклассовой классификации. Для классификатора можно настроить количество признаков k, но по умолчанию используется $k = sqrt(n)$ признаков. Также у функции есть параметр n_jobs, позволяющий запускать сразу несколько процессов тренировки либо построения предсказаний модели параллельно. В Python-библиотеке sklearn.linear_model реализован метод опорных векторов.

2.2 Определение тональности:

Функция LogisticRegression позволяет создавать и задавать параметры классификатора, решающего задачу бинарной классификации методом регуляризованной логистической регрессии.

LogisticRegression(C=c), изменяя в данной функции параметр c и наблюдая за изменением метрик качества модели, производился подбор параметра регуляризации.

3 Техническое описание

3.1 Существующие компании

На рынке услуг для анализа лекарственных препаратов уже сейчас существует ряд компаний, предлагающих своим клиентам получить обратную связь от потребителей о качестве их товаров и услуг, на основе имеющихся открытых источников в Интернете.

3.2 Многоклассовая классификация

Задача, в которой классов, на которые нужно разделить объекты, больше, чем 2. В отличии от бинарной классификации, где возможные значения классов +/- 1. Существует ряд методов, которые сразу решают задачу многоклассовой классификации, но в моем случае я свел задачу многоклассовой классификации к задаче бинарной классификации.

Были реализованы следующие методы:

1. СВМ – линейный классификатор. Просто реализуется, хорошо работает с большим объемом данных, любыми типами признаков (вещественные, категориальные и разреженные). 2. Случайный лес – строит композицию независимых друг от друга классификаторов (решающих деревьев). А после, путем голосования выбирается тот класс, к которому его отнесло большинство классификаторов 3. Градиентный бустинг – также строит композицию классификаторов, но использует взвешенное голосование, где каждый классификатор имеет собственный вес при голосовании. А также, в отличии от СЛ, где классификаторы строятся независимо друг от друга, в бустинге базовые алгоритмы строятся друг за другом, исправляя ошибки предыдущих алгоритмов.

Набор данных Представляет собой текстовые отзывы потребителей сервиса Amazon, извлеченные пользователем Julian McAuley, разделенных на 24 категории. Я проводил классификацию по 3 категориям из 24.

- Медицина и здоровье
- Красота и уход
- Товары для детей

Эти группы выбраны не случайно. Они относятся к близким предметным областям и используют схожу лексику, в отличии от более контрастных групп (например, если бы на ряду с медициной рассматривались электроника и кино).

Процесс обучения Из каждой из 3-х групп было взято одинаковое количество отзывов. Далее, данные разделы на 2 группы в соотношении 70/30. На 70% данных мы обучаем наши модели (т.н. обучающая выборка), а на оставшихся 30% мы оцениваем их качество (т.н. тестовая выборка).

3.3 Метрики качества

Существует несколько видов метрик качества, которые позволяют оценивать качество алгоритмов с разных сторон. Мы оценивали качество классификации тремя видами:

- ДОЛЯ ПРАВИЛЬНЫХ ОТВЕТОВ (Accuracy). Показывает число верных ответов к общему числу отзывов в выборке. Это самая простая и самая понятная метрика, но не всегда отражает реальную эффективность алгоритма.
- СРЕДНЕЕ ГАРМОНИЧЕСКОЕ (F2-мера). Объединяет показатели точности работы алгоритма и его полноту. (то есть, на сколько хорошо алгоритм определяет ложные срабатываний и ложные пропуски)
- ПЛОЩАДЬ ПОД ROC-КРИВОЙ (AUC-ROC). Является распространенной метрикой в машинном обучении. Этот показатель означает вероятности того, что случайно взятый объект класса 1 получит оценку принадлежности к классу 1 выше, чем случайно взятый объект класса 0.

3.4 Результаты многоклассовой классификации

Для всех трех методов были подобраны параметры, при которых каждый метод работает лучше всего. На рис.1 представлены показатели качества этих трех методов, а также параметры при которых достигаются эти показатели.

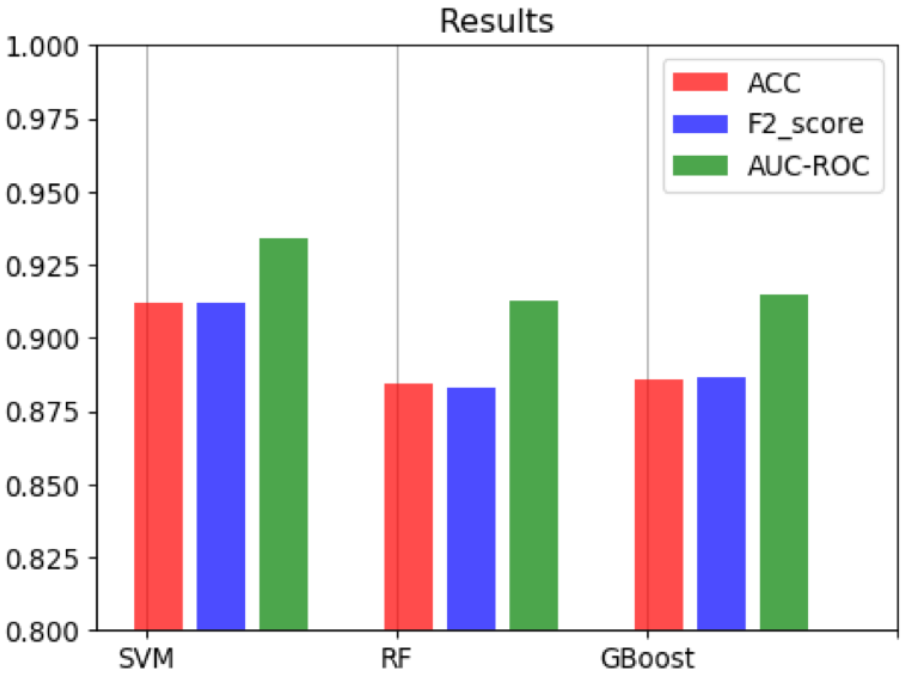


Рис. 1: Results.

Количество отзывов по классам тональности

3.5 Обработка данных для задачи анализа тональности

Первым шагом в данном упражнении, переведём оценки из формата 1-5 в бинарный (-1/1), для этого воспользуемся следующей логикой:

- при overall $\in$ (1,2) присвоим значение -1, что будет означать, что отзывы данной категории имеют отрицательную тональность;
- при overall $\in$ (3) не будем присваивать никакого значения, так как мы будем рассматривать задачу бинарной классификации;
- при overall $\in$ (4,5) присвоим значение 1, так как оставляя отзывы с такими оценками, мы можем быть уверены, что пользователь был удовлетворён своей покупкой, а значит отзыв имеет положительную тональность;

Имеем 5 наборов данных, состоящих из отзывов пользователей и их оценок, где в каждом наборе данных содержится приблизительно миллион записей. Так как в задачах машинного обучения важно иметь

сбалансированные выборки, а в разных категориях товаров разное количество положительных и отрицательных отзывов, то была написана функция, которая считает в каждом наборе данных количество: отдельно положительных и отдельно отрицательных отзывов, берёт наименьшее из них и обрезает класс, где отзывов больше, тем самым получается равное количество положительных и отрицательных отзывов в выборке.

Далее удаляем из данных знаки препинания и другие лишние символы (отмечу, что они могут быть исследованы при более глубоком анализе) и приводим все слова к нижнему регистру.

3.6 Анализ тональности

3.6.1 Векторизация отзывов

Первым этапом при анализе тех или иных текстов методами машинного обучения, текстовые данные требуется перевести в числовой формат. Метод, суть которого состоит в оценке важности слова как в контексте рассматриваемого экземпляра документа, так и относительно других документов, находящихся в коллекции. Зачастую метод TF-IDF также используется для оценки схожести текстов;

Алгоритм помогает избавиться от предлогов, союзов и других частей речи, которые часто встречаются в любом тексте. Например, в русском языке: "это" ,"и" , "а" , "или" , а в английском языке: "is" , "a" , "the" и др.

Частота слов (TF) вычисляется как отношение количества вхождений конкретного слова в документе к общему количеству слов в этом документе и записывается в следующем виде:

$$tf(t,d) = \frac{n_t}{\sum_k (n_k)}, \tag{1}$$

где n_t - количество вхождений слова t в документ n, $\sum_k (n_k)$ - общее число слов в документе n.

Обратная частота документа (IDF) - отношение общего количества документов в коллекции к числу документов, где присутствует данное слово и записывается в виде:

$$idf(t) = \ln \frac{1+n}{1+df(t)} + 1, \tag{2}$$

где n отвечает за общее число документов, входящих в набор данных, тогда как $df(t)$ - количество документов, которые содержат слово t.

В конечном итоге, метод принимает следующий вид:

$$tf - idf(t,d) = tf(t,d) \times idf(t). \tag{3}$$

Также в большинстве случаев, после того как выполнено преобразование TF-IDF принято полученные вектора нормализовать, используя Евклидову норму, имеющую вид:

$$v_{norm} = \frac{v}{||v||_2} = \frac{v}{\sqrt{{v_1}^2 + {v_2}^2 + \ldots + {v_n}^2}}. \tag{4}$$

Теперь рассмотрим пример применения данного метода на следующих предложениях:

- "Тебя знаю точно" ;
- "Знаю ли тебя" ;
- "Знаю точно себя" .

Словарь уникальных слов будет состоять из 5 слов и иметь следующий вид: 'знаю', 'ли', 'себя', 'тебя', 'точно'.

Рассмотрим первое предложение: "Тебя знаю точно" .

Начнём со слова 'тебя': $n = 3; df(t)_{term1} = 2;$

$$idf(t)_{term1} = \ln \frac{1+n}{1+df(t)} + 1 = \ln(4/3) + 1 = 1.2877. \tag{5}$$

После нормализации Евклидовой нормой, величина будет иметь вид:

$$\frac{(\ln(4/3)+1)}{((\ln(1)+1)^2 + (\ln(4/3)+1)^2 + (\ln(4/3)+1)^2)^{(1/2)}} = 0.6198. \tag{6}$$

Итоговое значение слова 'тебя' будет равно значению слова 'точно', так как встречается два раза в наборе данных и является одним из трёх слов в каждом предложении.

По аналогии, подставляя корректные значения в формулы выше, получаем следующие значения, которые удобно записываются в виде матрицы. При расположении слов в таком порядке: 'знаю', 'ли', 'себя', 'тебя', 'точно', для трёх предложений выше получается следующая матрица:

0.48133417 0 0 0.61980538 0.61980538
0.42544054 0.72033345 0 0.54783215 0
0.42544054 0 0.72033345 0 0.54783215

3.6.2 Логистическая регрессия

Для решения данной задачи бинарной классификации будет использоваться логистическая регрессия, которая является одним из методов построения линейного классификатора и даёт оценивать апостериорные вероятности принадлежности объектов классам.

Логистическая регрессия строит регрессионную модель, которая может предсказывать вероятности того, что запись относится к классу 1. Как линейная регрессия использует линейную функцию, так логистическая регрессия основана на функции вида сигмоида, которая записывается следующей формулой:

$$\sigma(z) = \frac{1}{1 + e^{(-x)}}. \tag{7}$$

И имеет вид, изображённый на рисунке 2:

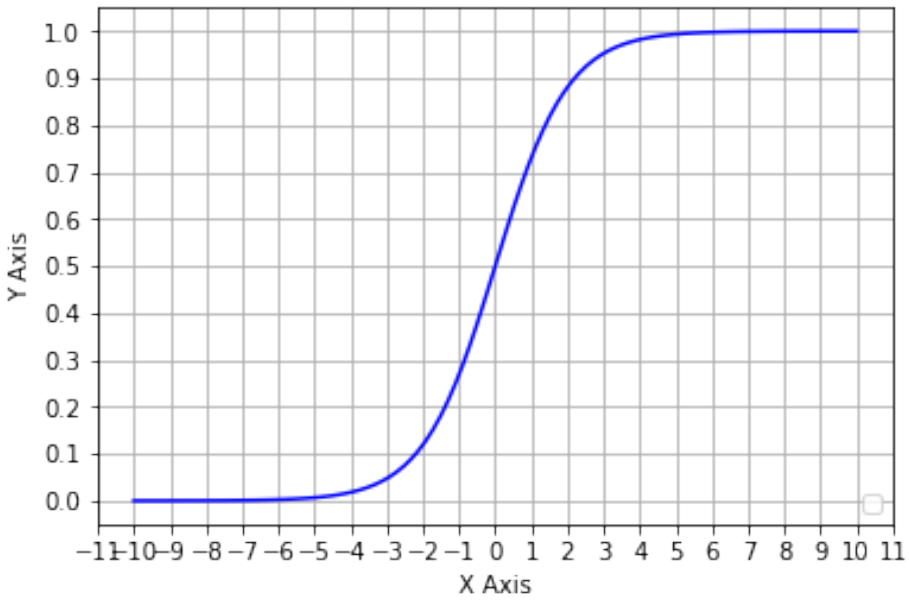


Рис. 2: Логистическая функция (сигмоида).

Моделью классификации логистическая регрессия становится только после введения порогового значения. На самом деле, подбор пра-

вильного порогового значения зависит от значений метрик: точность и полнота.

В зависимости от количества категорий, которые необходимо классифицировать, логистическая регрессия может иметь вид:

- Биноминальная: целевая переменная принимает только два значения, например: "да-нет" , "выиграет-проиграет" ;
- Мультиномиальная: целевая переменная принимает больше двух значений, например: категории товаров - "категория А" ,"категория Б" ,"категория В" ;
- Порядковая: целевая переменная принимает значения, принадлежащие порядковой шкале, например: "неудовлетворительно" , "удовлетворительно" ,"хорошо" ,"отлично" .

Мною была рассмотрена именно биноминальная регуляризованная логистическая регрессия.

3.6.3 Оценка точности классификации

Исследователями выделяется большое количество оценки точности классификации.

Данную формулу также принято записывать в следующем виде:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{8}$$

где:

- TP (True Positive) - количество истинно-положительных результатов, когда заданные объекты класса 1 равны результату алгоритма 1;
- TN (True Negative) - количество истинно-отрицательных результатов, когда заданные объекты класса 0 равны результату алгоритма 0;
- FP (False Positive) - количество ложно-положительных результатов, заданные объекты класса 0 равны результату алгоритма 1;
- FN (False Negative) - количество ложно-отрицательных результатов, заданные объекты класса 1 равны результату алгоритма 0.

3.6.4 Результаты

Результаты работы классификатора:

Название категории	AccuracyRate, %
"Рецензии на книги"	89.96%
"Здоровье и личная гигиена"	73.35%
"Кино и Телевидение"	88.61%
"Электроника"	87.54%

3.7 Визуализация

3.7.1 Платформа Tableau Public

При решении задачи анализа большого количества данных целесообразно применять удобные средства визуализации. В рамках нашего проекта было решено использовать платформу Tableau Public (public.tableau.com) – бесплатный редактор инфографики, позволяющий представить большинство типов данных в доступной форме. Это современное удобное средство построения разного рода зависимостей с возможностью последующей публикации на веб-ресурсах. Этот сервис позволяет работать с разными форматами данных - Microsoft Exel, Text, JSON, pdf и др. Полученные в результате рисунки автоматички сохраняютя в личном кабинете пользователя, что обеспечивает доступ к ним из любых устройств. Общий объем файлов пользователя ограничивается 10 Гб. Свое изображение можно дополнять текстом, а также прикреплять к нему гиперссылки и другие рисунки (см. рис. 3).

Соотношение занимаемой площади элементов и их порядок задается пользователем. Результат работы можно публиковать на веб-страницах. При этом пользователь может интерактивно взаимодействовать с полотном: выделять зависимости, приближать изображение, изменять интервалы фильтрации.

Приобтаемые навыки работы с платформой при построении графиков ROC-кривых алгоритмов классификации, подбора параметра C и различных вспомогательных изображений будут применяться в визуализации результатов анализа нашего конечного продукта.

3.7.2 Облака слов

Если алгоритм машинного обучения обрабатывает текстовые данные, полезно иметь представление о них. В частности, для алгоритма ло-

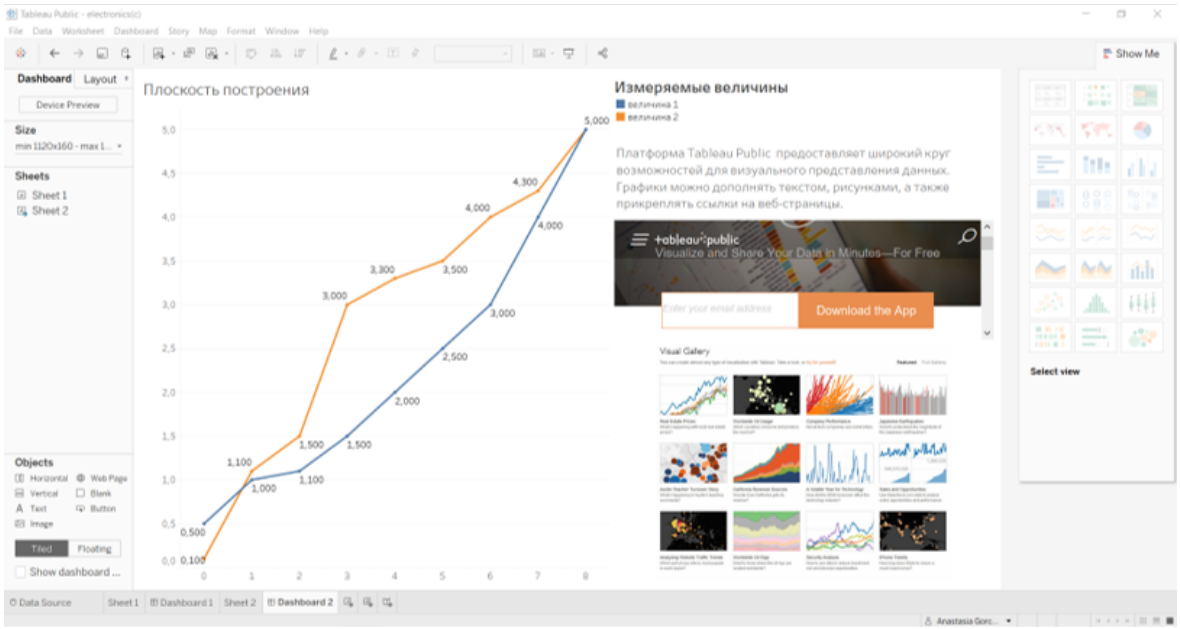


Рис. 3: Типичный вид интерфейса плтформы Tableau Public

гистической регрессии мы строим облака слов [1] – изображение некоторого множества слов в соответствии с их критерием значимости. Критерий значимости определяется алгоритмом машинного обучения и представляет собой дествительное положительное или отрицательное число.

Построение облака слов в Tableau Public выполнено на примере классификации методом логистической регрессии. Для каждой категории данных получено изображение, где значимость слова определяется его близостью к центру, размером и интенсивностью оттенка. На рис. приведдены несколько из них для сравнения. В первой паре совпадений крайне мало, хотя обе категории тесно связаны с общим понятием «досуг». Во второй паре совпадений намного больше, хотя категории и кажутся и более отдаленными. Из этого можно сделать вывод, что нельзя выделить какое-то единственное множество слов разумного объема, используемое при описании чего-либо.

3.7.3 Визуализация процесса обучения

На основе данных, полученных из метода логистическоой регрессии были построены кривые, визуализирующие процесс обучения. На рис. 4.3.3 видно, как зависит точность классификатора от параметра алгоритма C. Максимальная точность достигается в интервале от 1 до 10.

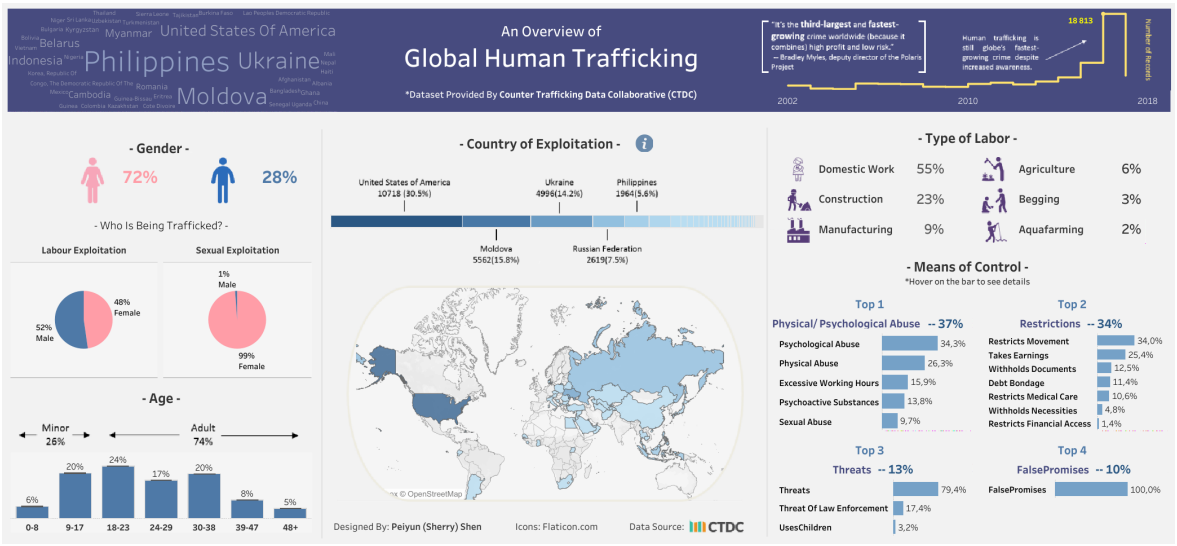


Рис. 4: Пример инфографики, выполненной в Tableau Public

Для визуального представления точности работы классификатора удобно использовать ROC-кривые. На рис.4.3.3 такая кривая для категории Health and Personal Care.

Конечно, приведенные зависимости можно построить и пользуясь другими ресурсами, но эта работа была проделана для определения особенностей редактора. На данный момент мы знаем какую структуру должны иметь данные, их объемные ограничения, а также опробовали способы построения наиболее популярных форм визуализации. Полученный опыт будет применен для оформления конечного результата нашего продукта.

4 Руководство пользователя

4.1 Задача определения тональности:

1. Подготовить заранее размеченные данные в формате “отзыв” – “оценка”:
   - Так как проводится бинарная классификация, поэтому оценка должна быть в формате 1 или 0.
   - Можно взять данные по ссылке: http://jmcauley.ucsd.edu/data/amazon/
2. Файл необходимо сохранить в формате json

Рис. 5: Визуализация облака слов для разных категорий

3. Открыть скрипт (скачав предварительно с github и изменить месторасположение файла

4. Запустить скрипт https://github.com/pekshin/kursach/blob/master/v_1_reviews_analysis/english_dataset/Logistic_modeling_reviews_Amazon_v3.1.ipynb

5. По итогу будут подготовлены следующие файлы в формате .csv:

   - Файл по подбору параметра “c”, поместив который в Tableau Public, вы увидите как происходил подбор параметра “c”
   - Файл с данными для построения “ROC” кривой, поместив которые в Tableau Public, вы увидите качество модели
   - Файл с весовыми коэффициентами слов, поместив которые в Tableau Public, вы увидите качество модели

6. На выходе также будут натренированные модели для каждой категории товаров, которые можно использовать, например, прогнозировать тональность отзывов товаров в категории в тех случаях, когда тональность была не проставлена

4.2 Задача многоклассовой классификации

Для того, чтобы воспользоваться продуктом, перейдите по ссылке и скачайте Python-код.

https://github.com/satankov/SML/blob/master/project_.ipynb

Далее, для работы с программой нужны данные. Скачать их можно по ссылке http://jmcauley.ucsd.edu/data/amazon/

В работе используются 3 датасета: Beauty, Baby, Health and Personal Care.

Вы можете скачать любой другой набор данных по другим категориям.

Количество датасетов также можно менять.

Чтобы ввести свои данные перейдите к следующей строке:

Введите необходимые данные

```
# название групп и путь к файлам

path = np.array(['baby','data/reviews_Baby_5.json',
                 'beauty','data/reviews_Beauty_5.json',
                'health','data/reviews_Health_and_Personal_Care_5.json']).reshape(-1,2)

# DATA FROM http://jmcauley.ucsd.edu/data/amazon/
# 5-star rattings groups: Beauty, Baby, Health and Personal Care
# I can't upload this files on GitHub, bcoz they're bigger than 25 MB


# кол-во записей каждой из групп товаров

N_1 = 10000     # кол-во записей 1-й группы
N_2 = 10000     # кол-во записей 2-й группы
N_3 = 10000     # кол-во записей 3-й группы


# параметры для моделей
C_ = 1          # параметр для SVM
N_ = 200        # параметр для RF
I_ = 250        # параметр для Gboost
```

Рис. 6: Необходимые данные

В переменную “path”, придерживаясь синтаксиса языка Python, введите через запятую название группы и путь к файлу.

В переменные “N_1”, “N_2”, “N_3” введите количество отзывов, которые нужно взять с каждой группы товаров (группы 1, 2, 3 соответственно).

В работе используются 3 метода для решения задачи классификации:

- Метод опорных векторов (SVM)
- Метод случайного леса (RF)

- Метод градиентного бустинга над решающими деревьями (Gboost)

Вы можете настроить дополнительно параметры для каждого из этих трех методов, где параметр "C" – для SVM, параметр "N" – для RF, параметр "I" – для Gboos

4.3 Визуализация в Tableau Public

4.3.1 Авторизация в системе

Для начала работы нужно перейти на сайт public.tableau.com и ввести логин и пароль (gorchavkina5@outlook.com и lunanavsegda1997). После авторизиции пользователю доступны для просмотра все выпоненные шаблоны визуализаций. На этой же сранице нужно загрузить приложение на рабочий компьютер. Это позволит применять в шаблонах новые данные.

4.3.2 Входные данные

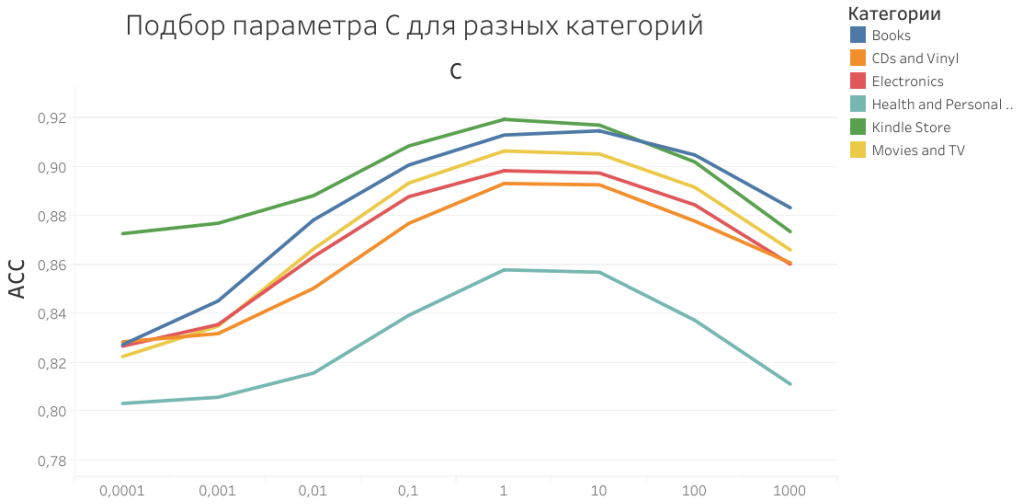
На данный момент существуют три типа готовых шаблонов: графики подбора параметра C и ROC-кривой алгоритма классификации, а также построение облака слов.

Для построения облака слов пользователь заменяет в одноименном шаблоне данные на свои (слова и их коэффициент значимости, подсчитанный в алгоритме логистической регрессии). По завершению работы нужно выпонить команду Save to Tableau Public или Save to Tableau Public as, что автоматически сохраняет график в личном кабинете с возможностью последующих изменений. Посроение кривой подбора параметра C выполняется аналогичным образом.

Может случиться, что входные данные для построенния ROC-кривой превышают максимально допустимуй объем (1000 строк). Поскольку кривая существует только в интевале (0, 1) по определению, рисунок не потеряет точность, если выполнить перезапись данных. Для этого нужно воспользоваться кодом на https://github.com/Gorchavkina/ROC/blob/master/Health_and_personal_care.ipynb.

4.3.3 Публикация результатов

Для публикации рисунков в социальных сетях или на веб-страницах, пользователь должен воспользоваться функцией Share в личном кабинете, расположенной в нижнем правом углу рамки полотна рисунка.


Подбор параметра C для разных категорий
C
ACC
0,92
0,90
0,88
0,86
0,84
0,82
0,80
0,78
0,0001
0,001
0,01
0,1
1
10
100
1000
Категории
Books
CDs and Vinyl
Electronics
Health and Personal ..
Kindle Store
Movies and TV


ROC-кривая для категории Health and Personal Care
fpr
tpr
1,0
0,8
0,6
0,4
0,2
0,0
0,00
0,01
0,01
0,02
0,02
0,03
0,03
0,04
0,04
0,05
0,06
0,06
0,07
0,08
0,08
0,09
0,10
0,11
0,11
0,12
0,13
0,14
0,16
0,17
0,18
0,19
0,20
0,21
0,22
0,24
0,25
0,27
0,28
0,30
0,33
0,35
0,38
0,41
0,44
0,47
0,52
0,58
0,65
0,80