РЕФЕРАТ

Магистерская диссертация содержит 00 страниц, 00 рисунков, 00 таб- лиц. Список использованных источников содержит 00 позиций.

UPLIFT, МАШИННОЕ ОБУЧЕНИЕ, ИНКРЕМЕНТАЛЬНЫЙ ОТКЛИК, ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ, РАНЖИРОВАНИЕ КЛИЕНТОВ, ГРАДИЕНТНЫЙ БУСТИНГ, СЛУЧАЙНЫЙ ЛЕС.

Выпускная квалификационная работа посвящена исследованию возможных подходов к решению задачи прогноза инкрементального отклика клиента при получении СМС, при планировании рекламной кампании, с помощью UpLift моделирования.

С ростом глобализации и цифровизации появилась возможность работать с потребительскими данными, активно взаимодействовать с потребителями путем разных акций, особых предложений.

Но стоит взять во внимание, что каждая коммуникация стоит денег. Если клиентская база составляет 1 тыс. клиентов, то при стоимости одного СМС в 1 рубль, коммуникация будет не такой дорогой. Но если увеличить масштаб базы до миллиона или нескольких миллионов, то слепая рассылка всем клиентам подряд станет очень дорогой. Даже если у компании большой оборот выручки, каждая такая коммуникация будет ощутимо сказываться на общем бюджете.

Поэтому коммуникацию можно использовать гораздо более оптимальным способом. Например, совершать коммуникацию с потенциально ушедшим пользователем.

Однако с ростом клиентской базы даже выборочная коммуникация с потенциально потерянными клиентами будет затратной и следующей задачей является прогнозирование, повлияет ли коммуникация на пользователя.

В основной части приводится описание объектов исследования, теоре- тически обосновываются подходы к оценке работоспособности модели ре- комендаций в условиях отсутствия разметки, предлагаются новые функци- оналы качества для оценки и сравнения моделей рекомендаций, описыва- ются модели рекомендаций. В разделе 1.2 приводится описание подхода к оценке работоспособности системы рекомендаций на основе оценок пользо- вателей, формулируются гипотезы о качестве работы системы для различ- ных групп запросов. В разделе 1.3 предлагаются новые методы оценки ка- чества рекомендаций в условиях отсутствия разметки данных для задачи

ранжирования, приводится характеристика предлагаемых функционалов качества. В разделе 1.4 описываются архитектуры моделей рекомендаций, используемых для решения задачи ранжирования. В разделе 1.5 приво- дится описание процесса обработки текстовых данных для их дальнейше- го использования в моделях рекомендаций. ... В разделе 1.7 описываются параметры задачи. В разделе 2.1 изложен алгоритм решения сформули- рованной задачи. В разделе 2.2 приводятся использованные в численном эксперименте параметры задачи. В разделе 2.3 описывается практическая составляющая работы, представлены результаты численного эксперимен- та.

Результатами выпускной квалификационной работы являются предло- женный метод оценки работоспособности системы рекомендаций по оцен- кам пользователей, предложенные функционалы качества для оценки ре- комендаций в условиях отсутствия разметки данных для задачи ранжи- рования, спроектированные модели рекомендаций документов по запросу, программная реализация алгоритмов рекомендаций, а также проведенный численный эксперимент.

В заключении подводятся итоги данного исследования и формулиру- ются выводы по проделанной работе.

Оглавление

[**ВВЕДЕНИЕ** 7](#_Toc136184118)

[**ОСНОВНАЯ ЧАСТЬ** 9](#_Toc136184119)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 10](#_Toc136184120)

[1.1 Описание объектов исследования 10](#_Toc136184121)

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ](#_bookmark0) 5

[ОСНОВНАЯ ЧАСТЬ](#_bookmark1) 8

1. [ТЕОРЕТИЧЕСКАЯ ЧАСТЬ](#_bookmark2) 9
   1. [Описание объектов исследования](#_bookmark3) . . . . . . . . . . . . . . . . 9

[1.1.1 Система рекомендаций](#_bookmark4) . . . . . . . . . . . . . . . . . . 9

* + 1. [Запрос](#_bookmark7) 10
    2. [Рекомендация](#_bookmark9) 11
  1. [Оценка работоспособности системы](#_bookmark11) 12
  2. [Функционалы качества рекомендаций](#_bookmark14) 14
  3. [Рекомендательные модели](#_bookmark17) 17
     1. [Лингвистические алгоритмы](#_bookmark18) 18
     2. [Классификаторы соответствия документа запросу](#_bookmark23) 20
     3. [Рекомендации по векторной близости документа за-](#_bookmark25) [просу](#_bookmark25) 23
  4. [Предобработка текстовых данных](#_bookmark27) 24
  5. [Постановка задачи оценки вероятности обслуживания пре-](#_bookmark29) [следуемого игрока](#_bookmark29) 26
  6. [Описание параметров задачи](#_bookmark36) 28

1. [ПРАКТИЧЕСКАЯ ЧАСТЬ](#_bookmark37) 30
   1. [Алгоритм решения задачи](#_bookmark38) 30
   2. [Параметры задачи](#_bookmark40) 33
   3. [Результаты численного эксперимента](#_bookmark41) 35

[ЗАКЛЮЧЕНИЕ](#_bookmark44) 38

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#_bookmark44) 39

[ПРИЛОЖЕНИЯ](#_bookmark55) 42

# **ВВЕДЕНИЕ**

В данной выпускной квалификационной работе рассматривается проблема ранжирования клиентов для осуществления коммуникации самым убеждаемым клиентам, которые без той самой коммуникации не совершат целевое действие.

В данной работе решается проблема прогноза инкрементального отклика клиента при планировании коммуникаций с помощью UpLift моделирования методами машинного обучения, где на основании полученного значения будет происходить ранжирование клиентов от самых убеждаемых к самым неприкасаемым, для повышения эффективности коммуникации при сохранении объемов затрат на ее проведение.

Результаты данной работы будут использованы в отделе управления взаимоотношений с клиентами в ретейл компании косметики и парфюмерии.

Появление данной задачи обусловлено желанием проводить нативную коммуникацию только с теми людьми, которым это нужно, чтобы не тратить денежный ресурс в пустую на тех, кому коммуникация не нужна или даже вызовет негативные эмоции и заставит уйти к конкуренту.

Объектом исследования являются клиенты ретейл сети косметики и парфюмерии, которых мы хотим ранжировать для выделения наиболее убеждаемых

Предметом исследования выступает сравнение различных алгоритмов ранжирования методами машинного обучения на двух различных источниках данных.

Цель данной работы - разработка алгоритма UpLift моделирования методами машинного обучения для планирования проведения рекламной

кампании.

Основными задачами выпускной квалификационной работы являются:

1. Поиск и обработка информации по объектам исследования;
2. Исследование общих подходов при построении модели UpLift;
3. Построение и обучение моделей UpLift на собственных данных ретейл компании косметики и парфюмерии;
4. Оценка качества построенных моделей с помощью предложенных функционалов качества;
5. Анализ полученных результатов.

По итогам выполнения данной выпускной квалификационной работы поставленные задачи были успешно решены. Результат подтверждает релевантность существующих методов UpLift моделирования с помощью машинного обучения для повышения эффективности рекламной кампании.

Данная работа развивает описанные в [[1]](#_bookmark45) идеи прогнозированию эффекта от коммуникации для каждого клиента при планировании рекламной кампании. С помощью показателей качества обучения из [2] и [3] удалось определить наилучший алгоритм для Uplift моделирования из описанных в [1], [4].

# **ОСНОВНАЯ ЧАСТЬ**

## 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### 1.1 Описание объектов исследования

#### 1.1.1 Исходные данные

Как и оговаривалось ранее, объектом исследования являются клиенты розничной сети косметики и парфюмерии, по которым имеются исторические данные покупок, заработка и списания бонусов в программе лояльности и многое другое. За источник данных были взяты результаты массовой рассылки СМС в ноябре на 473 861 человек. По истечении недели после рассылки появляется возможность определить целевую переменную (target): 0 – нет покупки в течении недели, 1 – есть покупка в течении недели. И так как нам известно заранее, кому была отослана СМС, а кому нет, очень просто определяется параметр коммуникации (treat): 0 – человек не получал СМС, 1 – человек получил СМС. Для клиентов из эксперимента были рассчитаны покупательские показатели за 4 месяца до момента рассылки, которые будут использованы как обучающие признаки.

Опишем набор данных детальнее. Он состоит из:

* Общая информации о клиентах и целевые переменные для обучения (рис 1.1):

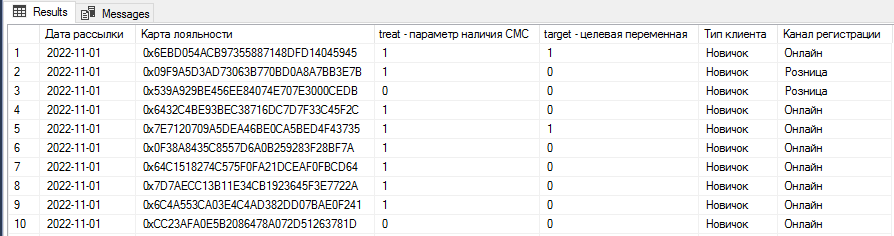


Рис 1.1 – Срез общих анкетные данные клиентов

* История покупок клиентов до коммуникаций (рис 1.2):

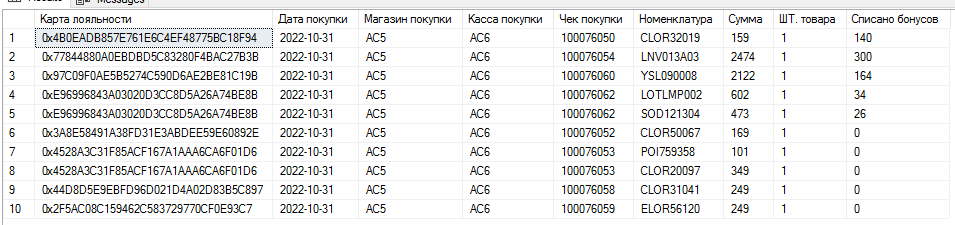


Рис 1.2 – Срез детализации покупок клиентов

#### 1.1.2 Анализ и агрегирование данных

Так как данные для UpLift моделирования составляют находятся в базе SQL Server компании, то было решено и взаимодействовать с ними через реляционный язык запросов T-SQL. Для этого был использован менеджер запросов SQL Management Studio (рис 1.3).

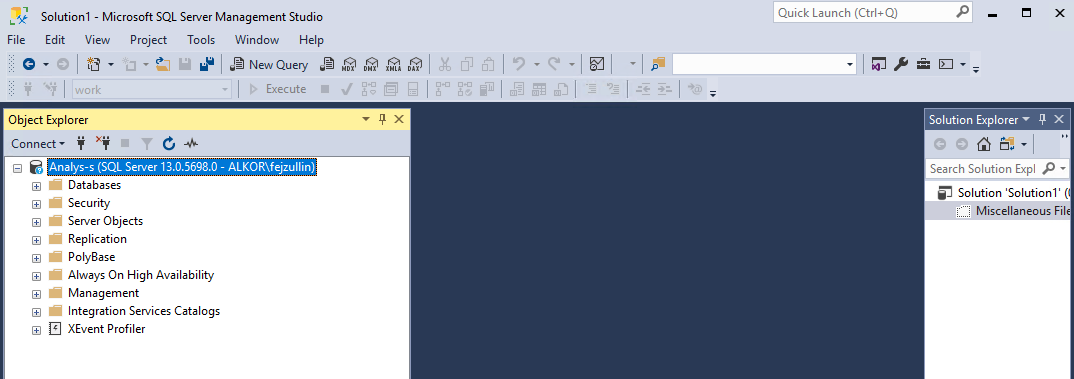


Рис 1.3 – Окно среды выполнения SQL запросов

Для моделирования основных обучающих признаков был использован принцип RFM – сегментации [5]. То есть, по покупкам клиентов были определены следующие параметры:

* Частота покупок – количество покупок за расчетный период.
* Период с момента последней покупки.
* Сумма товарооборота с клиента за расчетный период - в нашем случае возьмем средний чек, так как это стратифицировать клиентов явным образом.

Также была собрана статистика по среднему времени между покупками, минимальном и максимальному интервалу между покупками, а также по трате и заработку бонусов программы лояльности, средняя скидка за счет бонусов, количество покупок и суммы с тратой всех бонусов, количество покупок и суммы с тратой заработанных бонусов, количество покупок и суммы с тратой начисленных в периоды акций бонусов. Вдобавок к этому были учтены и анкетные данные.

Таким образом было получено пространство из 32-ух обучающих признаков (рис 1.4):

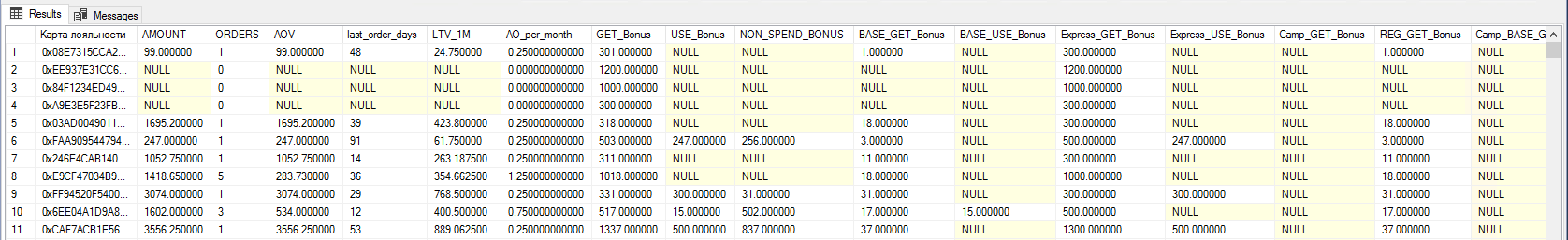


Рис 1.4 – Срез агрегированных показателей клиентов

* 1. Оценка работоспособности системы

Для оценки качества работы системы рекомендаций можно использо- вать поведение пользователей, а именно их оценки, описанные в 1.1.3.

Реализации обеих разновидностей оценок можно считать исходом опы- та по схеме Бернулли [[8].](#_bookmark52) При этом успехом в случае прямой оценки счита- ется положительная оценка пользователя, а в случае косвенной - открытие документа из выдачи по запросу (табл. 1.1). Таким образом, можно ви- деть, что при прямой оценке происходит оценивание качества на уровне документов-рекомендаций, а при косвенной - на уровне запроса.

Таблица 1.1: Исход в опыте по оцениванию рекомендаций

|  |  |  |
| --- | --- | --- |
|  | Реализация прямой оценки | Реализация косвенной оцен- ки |
| Успех | Документ оценен положи- тельно | По запросу открыт документ из выдачи |
| Неудача | Документ оценен отрица- тельно | По запросу нет открытий до- кументов |

Оба подхода к оценке успеха рекомендации можно использовать для

проверки гипотезы, что для различных групп запросов алгоритм рекомен- даций работает одинаково. Группировку запросов можно проводить по спо- собу задания запроса (копирование текста или описание своими словами), по сфере запроса, а также комбинируя группировки.

Отличительной чертой копируемых текстов является их длина, по- этому была выбрана граница длины запроса, после преодоления которой запрос считается скопированным текстом.

Таким образом, запросы группируются по длине:

1. Короткие (не более 70 слов);
2. Длинные (более 70 слов).

А также по сфере запроса:

1. Экономические;
2. В сфере общей юрисдикции.

Для каждого типа оценки (прямой и косвенной) была сформулирова- на гипотеза о равенстве долей [[8]](#_bookmark52) рекомендаций с успешным исходом для различных групп запросов (по длине запроса и сфере описанной в запросе ситуации).

В качестве нулевой гипотезы используется утверждение: вероятности успеха для обеих рассматриваемых групп запросов равны *p* [(1.1).](#_bookmark12) Альтер- нативная гипотеза является отрицанием нулевой [(1.2).](#_bookmark13)

*H*0 : *p*1 = *p*2 = *p* (1.1)

*HA* : *p*1 ̸= *p*2 (1.2)

где *pi* - вероятность успеха для *i* группы (*i* = 1, 2).

Таким образом, сформулирована гипотеза [(1.1)-(1.2):](#_bookmark13) для различных групп запросов алгоритм рекомендаций работает одинаково.

* 1. Функционалы качества рекомендаций

Задача формирования рекомендаций – задача ранжирования. Для оцен- ки качества ранжирования необходима специфическая разметка данных [[1]](#_bookmark45), где для каждой рекомендации в ранжированном списке ставится в со- ответствие некое значение, показывающее релевантность рекомендации за- просу. На основе подобной разметки вычисляются такие показатели каче- ства, как ...

В данной же работе рассматривается задача ранжирования в поста- новке, когда отсутствует такая разметка данных, поэтому стандартные спо- собы оценки качества предлагаемых алгоритмов в таких условиях непри- менимы.

В данной работе предлагается набор альтернативных функционалов качества ранжирования, не требующих разметку [[1]](#_bookmark45). Новые функционалы качества используют информацию о прямых оценках пользователей доку- ментам, описанных в 1.1.3. Далее под разметкой будет пониматься набор данных запрос-документ, где каждый документ имеет оценку от пользо- вателей. При этом важно, чтобы в таком наборе данных присутствовали только те запросы, для которых существует хотя бы одна релевантная и одна нерелевантная рекомендации.

На основе оценок пользователей предлагается оценивать показатели:

1. Правильности сортировки;
2. Ширины «окна» между релевантными и нет рекомендациями;
3. Степени поднятия релевантных документов.

Центральная идея новых функционалов - алгоритм рекомендаций с более высоким качеством ранжирования должен поднимать выше докумен- ты, которые пользователи считают релевантными, и опускать ниже нере-

левантные.

Показатель правильности сортировки представляет собой оцененную долю запросов, для которых алгоритм ставит все релевантные документы выше, а все нерелевантные - ниже. Данный показатель описывается фор- мулой:

*correctness* = *qgood*

|*Q*|

(1.3)

где *qgood* - количество запросов, все релевантные документы которых выше нерелевантных, *Q* - множество запросов в разметке.

Под шириной окна между релевантными и нет рекомендациями по- нимается разница между максимальным номером позиции релевантного документа и минимальным номером позиции нерелевантного. Такое окно вычисляется для каждого запроса из разметки. Далее проводится стати- стический анализ полученных значений, в том числе оценивается некий статистический показатель. В данной работе предлагается оценивать ме- дианный показатель:

*window* = *median* max

{

*good*∈*recomq*

*pos*(*good*) min

∈*recomq*

— *bad*

*pos*(*bad*), *q* ∈ *Q*} (1.4)

где *q* - запрос, *recomq* - множество рекомендаций по запросу *q*, *good* - реле- вантная рекомендация, *bad* - нерелевантная рекомендация, *pos*(·) - номер позиции рекомендации.

Степень поднятия релевантных документов можно считать дополни- тельной вспомогательной характеристикой алгоритма рекомендаций. Дан- ный показатель - доля релевантных документов, которые подняты выше относительно своих позиций в рекомендациях старого алгоритма:

*upli f t* = |*doc* : *posnew*(*doc*) < *posold* (*doc*), *doc* ∈ *gooddoc*|

|*gooddoc*|

(1.5)

где *gooddoc* - множество релевантных рекомендаций, *doc* - рекомендация,

*posnew*(·) - номер позиции в новых рекомендациях, *posold* (·) - номер позиции

в старых рекомендациях.

В отличие от описанных выше функционалов, данный показатель в общем случае не показывает, насколько ранжирование нового алгоритма качественнее, чем ранжирование старого, однако он может быть полезен для понимания, работает ли новый алгоритм на том же множестве доку- ментов, что и старый.

Таким образом, предлагаемые функционалы качества имеют характе- ристику, представленную в таблице 1.2.

Таблица 1.2: Характеристика функционалов качества рекомендаций

|  |  |
| --- | --- |
| Правильность сортировки | |
| Определение | Доля запросов, для которых все релевантные рекомен- дации находятся выше всех нерелевантных |
| Смысл | Показывает степень правильности сопоставления алго- ритмом смысла запроса и смысла рекомендуемых доку- ментов |
| Недостатки | Отсутствует градация степени правильности сортиров- ки, то есть показатель не учитывает случаи, когда прак- тически все релевантные рекомендации оказались выше, но имеет место незначительное количество нерелевант- ных рекомендаций, оказавшихся выше |
| Ширина окна | |
| Определение | Медиана разницы наибольшей позиции релевантного до- кумента и наименьшей позиции нерелевантного |

Продолжение таблицы 1.2

|  |  |
| --- | --- |
| Смысл | Показывает, насколько далеко позиционно находятся группа релевантных и группа нерелевантных докумен- тов |
| Недостатки | Показатель не учитывает случаи, когда практически все релевантные рекомендации оказались выше, но имеет ме- сто незначительное количество нерелевантных рекомен- даций, оказавшихся выше |
| Степень поднятия | |
| Определение | Доля релевантных документов, которые были подняты выше относительно позиционирования базовым алгорит- мом |
| Смысл | Показывает, какое количество релевантных документов пользователи увидят раньше, чем при базовом ранжиро- вании |
| Недостатки | Не учитывает, что новые алгоритмы могут не поднимать релевантные документы по причине нахождения более релевантных |

* 1. Рекомендательные модели

В качестве потенциальных рекомендательных алгоритмов были рас- смотрены как предобученные [[4,](#_bookmark48) [7]](#_bookmark51) модели, так и спроектированные линг- вистические и семантические модели.

Лингвистические модели более просты в реализации, однако не ис- пользуют информацию о семантике текста, а опираются только на различ- ные частоты встречаемости слов.

Семантические же модели более сложные в проектировании, так как требуют обучения, однако они учатся понимать смысл текстов. Также мож- но заранее создавать базы данных для семантических векторов докумен- тов, что значительно сокращает затраты по времени на обработку пар запрос-документ, так как требует лишь разовой обработки запроса.

Рассматриваемые семантические модели делятся на следующие груп-

пы:

1. Классификаторы соответствия документа запросу
2. Рекомендации по векторной близости документа запросу
   * 1. Лингвистические алгоритмы

Основная идея лингвистических алгоритмов - идея построения реко-

мендаций на основе ранжирования документов по различным показателям пересечения слов запроса и документа.

Были рассмотрены следующие лингвистические алгоритмы рекомен- даций:

1. Рекомендации по пересечению слов запроса и документа
2. Рекомендации по сумме IDF [[2]](#_bookmark46) слов запроса
3. Рекомендации по сумме TF-IDF [[2]](#_bookmark46) слов запроса
4. Рекомендации по сумме BM25 [[3]](#_bookmark47) слов запроса

Все эти алгоритмы имеют одну и ту же схему построения рекоменда- ций, которую можно описать рисунком [1.5.](#_bookmark19)

Пусть *Qword* - множество слова запроса; *stop* - множество стоп-слов; *Dword* - множество слов документа; *N* - количество документов; *d fword* - количество документов, содержащих *word*; *k*, *b* - выбираемые параметры; *avgdl* - среднее количество слов в документе.

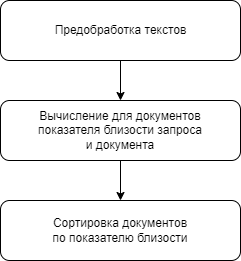


Рис. 1.5 – Лингвистический алгоритм рекомендаций

Также вводится понятие индикаторной функции, которая показывает наличие или отсутствие слова во множестве слов:

*I* (*w*, *d*) = 1, *w* ∈ *d*

∈

0, *w* ∈/ *d*

а также индикаторной функции, сигнализирующей о совпадении слов:

*I*(*w*, *word*) =

1, *w* = *word*

0, *w* ̸= *word*



В [[2]](#_bookmark46) описаны показатели TF-IDF, которые ориентируются на показа- тели частотности слов в запросе и документах:

*TF*(*word*, *doc*) = ∑*d*∈*doc I*(*d*, *word*)

|*doc*|

*IDF*1(*word*) = log *N*

*d fword*

В [[3]](#_bookmark47) предлагается альтернативное представление IDF:

*IDF*2(*word*) = log *N* − *d fword* + 0.5

*d fword* + 0.5

(1.6)

(1.7)

Показатель близости при построении рекомендаций по пересечению слов выглядит следующим образом:

*similarity*∩

= ∑*word*∈*Qword*\*stop I*∈(*word*, *Dword* )

|*Qword* \ *stop*|

(1.8)

При сортировке по сумме IDF показателем близости является:

*similarityIDF* = ∑

*word*∈(*Dword*∩*Qword*)\*stop*

*IDF*1(*word*) (1.9)

Показателем близости при сортировке по сумме TF-IDF является:

*similarityTF*−*IDF* = ∑

*word*∈(*Dword*∩*Qword*)\*stop*

*TF*(*word*, *Dword* )*IDF*1(*word*) (1.10)

Показатель близости при построении рекомендаций по сумме BM25 слов:

*similarityBM*25 = ∑

*TF*(*word*, *Dword* )(*k* + 1)

·

*avgdl*

*word*∈(*Dword*

∩*Qword*

)\*stop TF*(*word*, *Dword* ) + *k*(1 − *b* + |*Dword* |)

· *IDF*2(*word*) (1.11)

Таким образом, схема [1.5](#_bookmark19) и показатели близости [(1.8)-(1.11)](#_bookmark22) представ- ляют собой соответствующие лингвистические алгоритмы.

Главным недостатком лингвистических алгоритмов является наличие вычислительных и временных затрат для каждой пары запрос-документ, которые могут быть достаточно большими, так как невозможно заранее создать базу данных лингвистических признаков документов, ведь они ста- новятся известными непосредственно в связке с текстом запроса.

* + 1. Классификаторы соответствия документа запросу Основная идея данного подхода - построить классификатор для пар

векторных представлений текстов запроса и документа. В задаче присут- ствует два класса пар запрос-документы: документ подходит запросу и документ не подходит запросу. В качестве классификаторов рассмотрены:

1. kNN-классификатор семантических векторов [[5,](#_bookmark49) [7]](#_bookmark51)
2. LightGBM-классификатор показателей векторной близости [[6](#_bookmark50), [7]](#_bookmark51)
3. BERT-классификатор [[4](#_bookmark48), [7]](#_bookmark51)

В основе первых двух классификаторов лежит идея использования результатов работы предобученной модели с архитектурой BERT [[4].](#_bookmark48) По- лучаемые векторные представления используются в качестве входных дан- ных моделей машинного обучения kNN и LightGBM. Таким образом, общая схема работы моделей выглядит следующим образом:

1. Получение векторных представлений текстов запроса и документа с помощью предобученной модели;
2. Обработка полученных представлений, формирование признаков классификатора;
3. Классификация пары запрос-документ по степени соответствия до- кумента запросу.

Для kNN-классификатора обработка векторных представлений сво- дится к их конкатенации. При такой обработке представлений выбор моде- ли обусловлен тем, что он направлен на поиск ближайших соседей в мет- рическом пространстве, а семантические векторы образуют метрическое пространство.

Для LightGBM-классификатора обработка семантических векторов пред- ставляет собой формирование набора значений показателей близости век- торов текстов запроса и документа. Соответствующие показатели близости приведены в таблице 1.3. Так как показатели близости часто могут иметь различные области значений, то в качестве модели используется градиент- ный бустинг над деревьями решений.

BERT-классификатор представляет собой предобученный BERT с до-

обучаемой на задачу классификации головой. Отличие данной модели от двух предыдущих в том, что она не является дополнительной надстройкой над результатом работы предобученной модели, а ее признаками являются сами тексты запросов и документов, обработанные особенным образом.

Таким образом, модели классификаторы имеют разную архитектуру и разные наборы признаков, представленные в таблице 1.3.

Таблица 1.3: Классификаторы соответствия документа запросу

|  |  |
| --- | --- |
| Архитектура модели | Входные признаки модели |
| Предобученный BERT  + kNN | Сконкатенированные векторные представления текстов запроса и документа, полученные пре- добученной моделью |
| Предобученный BERT  + LightGBM класси- фикатор | Показатели близости векторного представления документа и запроса: несходство Брея — Керти- са; расстояние Канберра; расстояние Чебышева; Манхэттенское расстояние; коэффициент корре- ляции; косинусовое расстояние; евклидово рас- стояние; расстояние Минковского |
| Предобученный BERT  + голова на классифи- кацию | Текст запроса и текст документа со специальны- ми токенами |

Итоговая сортировка документов осуществляется по показателю веро- ятности принадлежности документа классу подходящих запросу. Соответ- ственно, в итоговом ранжированном списке документы отсортированы по убыванию данного показателя. Показатель вероятности является выходом

модели и представляет собой значение сигмоидальной функции [(1.12).](#_bookmark24)

*σ* (*x*) = 1

1 + exp−*x*

(1.12)

* + 1. Рекомендации по векторной близости документа запросу Идея данного подхода - использование дообученной модели для по-

лучения векторных представлений текста запроса и текста документа и дальнейшее ранжирование документов по показателю близости между по- лученными векторами.

Для дообучения берется предобученная модель с архитектурой BERT (Bidirectional Encoder Representation Transformers) [[4]](#_bookmark48) - модель представле- ния языка, которая предназначена для предварительного обучения глубо- ких двунаправленных представлений на простых немаркированных текстах путем совмещения левого и правого контекстов во всех слоях. Это позволя- ет настраивать предварительно обученную модель BERT с помощью лишь одного дополнительного выходного слоя и получать наиболее актуальные результаты для широкого спектра задач.

Проще говоря, дообучение осуществляется путем решения с помощью модели двух задач классификации [[4]:](#_bookmark48)

1. NSP (Next Sentence Prediction) - классификация пары предложений на предмет их последовательного расположения в тексте;
2. MLM (Masked-Language Modeling) - предсказание скрытого токена по контексту.

При ообучении модель пытается как можно лучше научиться решить данные задачи, что приводит к получению достаточно качественных се- мантических векторов для подающихся на вход текстов.

Так как данная модель учится «понимать» смысл специфических юри- дических текстов, то в качестве показателя семантической близости меж-

ду запросом и документом можно использовать косинусову меру близости между получаемыми векторами запроса и документа:

*similaritycos* = 1 − cos(*query*, *doc*) (1.13) где *query* - вектор текста запроса, *doc* - вектор текста документа.

Ранжированные списки документов по запросу получаются с помощью [(1.13)](#_bookmark26) путем сортировки показателя по убыванию его значения.

* 1. Предобработка текстовых данных

Для лингвистических моделей предобработка текста сводится к:

1. Удалению знаков препинания;
2. Разделению текста на слова и их приведение к некой начальной форме;
3. Формированию словаря при исключении из него стоп-слов (наиболее частотных слов, вспомогательных слов, союзов и предлогов).

Для kNN и LightGBM классификаторов предварительная обработка не требуется, так как все необходмые для обучения и работы моделей при- знаки выделяются с помощью сторонней предобученной модели.

Входными данными для BERT-классификатора является последова- тельность токенов, которые сначала преобразуются в векторы, а затем об- рабатываются в нейронной сети. Но перед тем, как обработка может на- чаться, необходимо обработать входные данные и украсить их некоторыми дополнительными метаданными:

1. Встраивание токенов: токен [CLS] добавляется к токенам входного слова в начале первого предложения, а токен [SEP] вставляется в конце каждого предложения;
2. Встраивание сегмента: к каждому токену добавляется маркер, обо-

значающий предложение A или предложение B. Это позволяет кодировщи- ку различать предложения;

1. Позиционные вложения. Позиционные вложения добавляются к каж- дому токену, чтобы указать его положение в предложении.

Токенизация выполняется с помощью соответствующего предобучен- ного токенизатора. В качестве предложения А предлагается использовать текст запроса, а в качестве предложения В - текст документа.

По итогу такой обработки получаются следующие векторы, которые в дальнейшем подаются на вход модели:

1. Вектор индексов токенов вида «[CLS] текст запроса [SEP] текст до- кумента [SEP]», который при необходимости в конце дополняется специ- альными токенами [PAD], обозначающими дозаполнение вектора токена до максимальной длины;
2. Вектор внимания, в котором на позициях токенов [PAD] находятся 0, а на позициях остальных токенов - 1;
3. Вектор типа токенов, в котором на позициях токенов предложения А находятся 1, а на позициях предложения В - 0.

При дообучении языковой модели к описанным выше векторам добав- ляются еще два: копия вектора индексов токенов и вектор метки характера последовательности предложений, а также оригинальный вектор индексов токенов преобразовывается под задачу MLM путем замены случайных то- кенов специальным токеном [MASK]. Однако в качестве исходных текстов для обработки необходимо использовать корпус текстов документов, для которого в качестве предложений А и В берутся реальные предложения из текстов.

Для получения разметки пар предложений на предмет их последова-

тельности используется алгоритм, представленный на рис. [1.6.](#_bookmark28)

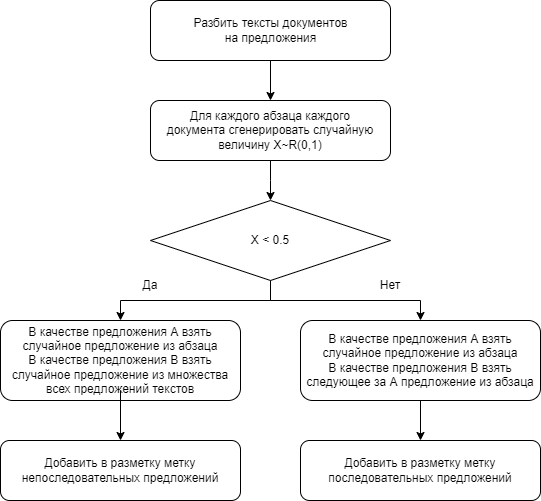


Рис. 1.6 – Алгоритм генерации NSP-разметки

* 1. Постановка задачи оценки вероятности обслуживания преследуемого игрока

Будем рассматривать случайную величину *ξ* , представляющую собой индикаторную функцию события *A* ={В результате игровой ситуации про- изошло обслуживание преследуемого объекта преследователем}:

*ξ* = 1, событие A произошло

0, событие A не произошло

(1.14)

и случайную величину терминального промаха *η* = *h*(Θ⃗), где ⃗Θ - вектор случайных координат преследователя с распределением (??), а *h*(Θ⃗) - некая

функция, значение которой вычисляется в процессе моделирования процес- са игры.

Для рассматриваемых случайных величин можно определить условное математическое ожидание I и II типов [[9].](#_bookmark53)

Условное математическое ожидание I типа представляет собой неко- торую функцию, принимающую детерминированные значения:

*M*[*ξ* |*η* = *d*] = *R*(*d*) (1.15)

В [(1.15](#_bookmark30)) описано математическое ожидание случайной величины *ξ* при условии, что случайная величина *η*, измеренная в опыте, приняла значение *d*.

[[9]:](#_bookmark53)

Условное математическое ожидание II типа - случайная величина *V* :

*V* = *M*[*ξ* |*η*] = *R*(*η*) (1.16)

Для [(1.16)](#_bookmark31) справедлива формула полного математического ожидания

*M*[*ξ* ] = *M*[*M*[*ξ* |*η*]] (1.17)

В то же время, нетрудно показать, что:

*M*[*ξ* ] = *P*(*A*), (1.18)

так как *ξ* является индикаторной функцией для события *A*, следовательно, среднее количество произошедших событий *A* в серии опытов - вероятность события *A*.

Из [(1.17)](#_bookmark32) и [(1.18)](#_bookmark33) следует, что:

*P*(*A*) = *M*[*M*[*ξ* |*η*]], (1.19)

то есть безусловная вероятность того, что в результате игровой ситуации произошло обслуживание преследуемого объекта преследователем, равна

значению полного математического ожидания случайной величины *ξ* от- носительно случайной величины *η*.

Оценку безусловной вероятности обслуживания можно получить с по- мощью метода Монте-Карло [?]. Для этого необходимо многократно разыг- рать игровую ситуацию между преследуемым и преследователем и полу- чить реализации терминального промаха.

Пусть

*θ*⃗*k* - реализация вектора случайных координат преследовате-

ля ⃗Θ, полученная в результате компьютерного моделирования. Тогда *dk* =

*h*(*θ*⃗*k*) - реализация терминального промаха, полученная в результате моде-

лирования процесса игры при

*θ*⃗*k*. Будем считать, что функция условного

математического ожидания *R*(*d*) I типа задана. В этом случае, *Rk* = *R*(*dk*) - реализации условной вероятности обслуживания. Тогда, согласно [(1.17)](#_bookmark32) и [(1.19),](#_bookmark34) оценка безусловной вероятности обслуживания имеет вид:

*P*ˆ(*A*) = *M*ˆ [*R*(*d*)] = *R*1 + ... + *Rn*, (1.20)

*n*

где *n* - количество реализаций терминального промаха.

Таким образом, необходимо с помощью компьютерного моделирования оценить безусловную вероятность обслуживания преследуемого объекта в задаче [(1.20).](#_bookmark35)

* 1. Описание параметров задачи

Опираясь на все вышесказанное, исходными данными к задаче явля- ются параметры для задачи проверки гипотез [(1.1)-(1.2),](#_bookmark13) задачи оценки доверительных интервалов оценок на разметке [(2.1),](#_bookmark39) а также параметры рекомендательных моделей.

Параметрами для проверки гипотез и задачи оценки доверительных интервалов оценок на разметке выступают уровни доверия.

Для лингвистических моделей единственными параметрами являют-

ся коэффициенты *k*, *b*, используемые для вычисления показателя близости текстов запроса и документа [(1.11)](#_bookmark22) в алгоритме ранжирования по сумме BM25.

Параметрами для kNN выступают количество соседей k и степень функции расстояния Минковского p.

Параметрами для алгоритма LightGBM являются скорость обучения lr, количество листьев дерева leaves, максимальная глубина дерева depth, количество деревьев n в ансамбле, над которым проводится градиентный бустинг.

Параметрами для моделей BERT выступают количество эпох обучения epoch, размер батча batch, скорость обучения lr и коэффициент регуляри- зации L. Параметры количества эпох и размера батча выбираются исходя из объема обучающих данных и сложности модели. Скорость обучения и коэффициент регуляризации являются параметрами функции оптимиза- ции.

* 1. ПРАКТИЧЕСКАЯ ЧАСТЬ
     1. Алгоритм решения задачи

Для решения задачи сравнения эффективности алгоритмов рекомен- даций семантически схожих документов необходимо предварительно ре- шить несколько подзадач, а именно:

1. Проверить гипотезу [(1.1)-(1.2)](#_bookmark13) о равенстве долей успешных исхо- дов в различных группах запросов. По полученным результатам сделать выводы о работоспособности системы;
2. Сформировать набор данных для оценки качества ранжирования и оценка погрешностей оценок на полученной разметке;
3. Обработать текстовые данные;
4. Реализовать рекомендательные модели;
5. Получить результаты ранжирования размеченных данных реализо- ванными моделями;
6. По полученным результатам провести оценку качества рекоменда- ций с помощью (1[.3](#_bookmark15)[)-(1.5).](#_bookmark16)

При решении задачи используется язык программирования Python.

Для проверки сформулируем 8 гипотез о равенстве долей (по 4 ги- потезы для каждого вида оценки: прямой и косвенной). В качестве групп для каждой гипотезы рассматриваем следущие пары множеств запросов: короткие и длинные, экономические и в сфере общей юрисдикции, корот- кие и длинные в экономической сфере, короткие и длинные в сфере общей юрисдикции. Для проверки гипотез используем библиотеку SciPy.

Для дальнейшего применения фнукционалов качества [(1.3)-(1.5)](#_bookmark16) фор-

мируем разметку данных, в которой для всех запросов которой существу- ют как положительно, так и отрицательно оцененные документы. Так как пользователи достаточно редко ставят и положительные, и отрицательные оценки по одному запросу, расширим понятие нерелевантной рекоменда- ции. Таковой будет считаться документ с отрицательной оценкой или про- пущенный документ.

Пропущенный документ – документ на 1-4 позициях, который не был открыт пользователем при том, что на позиции ниже были открытия до- кументов. Ограничение по позициям обусловлено тем, что на экране поис- ковой выдачи помещаются 4 документа, то есть можно считать, что поль- зователь всегда видит первые 4 рекомендации.

Для полученной разметке с использованием библиотеки SciPy оцени- ваем доверительные интервалы [[9]](#_bookmark53) потенциальных оценок по формуле:

*δ* = *u*1

*α* r *p*(1 − *p*) 1 − *n* (2.1)

где *p* = 0.5 - доля респондентов с оценкой, *α* - уровень значимости, *u*1− *α*

2

— 2

*n*

*N*

- квантиль уровня 1 − *α* стандартного распределения, *n* - количество ре-

2

спондентов, *N* - объем генеральной совокупности.

Формирование словаря для лингвистических моделей, а также пред- варительный расчет показателей IDF (??), [(1.7)](#_bookmark20) осуществляем на корпусе текстов документов. Обработку текстов производим с помощью библиоте- ки pymorphy2, которая позволяется выделять начальные формы слов сред- ствами морфологического анализа слов. Множество стоп-слов определяем по графикам частотности - исключаем наиболее частотные слова.

Для моделей kNN и LightGBM предварительная обработка текстов не требуется.

При обработке текстов для BERT-классификатора предварительно вы-

бираем значение максимальной длины входных векторов и ограничиваем длину текста запроса и текста документа значением, равным половине от максимальной длины. Для каждой пары запрос-документ проводим токе- низацию каждого из текстов и отсекаем лишние токены, которые не поме- щаются в ограничения по длине векторов. Полученные пары текстов токе- низируем с помощью токенизатора соответствующей модели из библиотеки transformers.

Тексты для языковой модели обрабатываем по алгоритму рис. [1.6.](#_bookmark28) При разделении текстов на предложения учитываем специфические особенно- сти юридических текстов, а именно, что точка может использоваться не только для обозначения конца предложения, но и в качестве сокращения, например, в ФИО. По этой причине используем более сложный алгоритм разбиения на предложения (приложение ...

Проводим токенизацию с помощью библиотеки для каждой получен- ной пары предложений и далее маскируем 15% токенов специальным то- кеном [MASK].

Для лингвистических моделей реализуем функции вычисления пока- зателей лингвистической близости [(1.8)-(1.11)](#_bookmark22) запроса и документа. Реали- зация соответствующих моделей рекомендаций далее заключается лишь в сортировке документов по вычисляемым показателям близости.

Для семантических моделей в качестве предобученной языковой моде- ли возьмем LaBSE [[10]](#_bookmark54) с архитектурой BERT [[4]](#_bookmark48) из библиотеки transformers. Реализации алгоритмов kNN и LightGBM берем из библиотеки sklearn.

При построении классификаторов в качестве разметки данных будем использовать прямые оценки пользователей. На этапе обучения оценку ка- чества соответствующих моделей будем проводить с помощью привычных функционалов качества для задачи классификации: показателей точности,

полноты и F1 для каждого класса, а также с помощью AUC под ROC- кривой. [] В случае kNN и LightGBM обучим только эти модели машинно- го обучения. В случае BERT-классификатора дообучим голову на задачу классификации.

Для получения рекомендаций по векторной близости документа за- просу дообучим на корпусе текстов документов предобученную языковую модель, которая будет учиться прогнозировать пропущенное слово по кон- тексту, а также понимать, является ли следующее предложение продол- жением предыдущего. Для этого используем обработанный ранее корпус текстов документов.

После проектирования моделей и их обучения получим результаты ранжирования на сформированной разметке данных. Для лингвистических моделей ранжирование проводим по соответствующим показателям близо- сти текстов [(1.8)-(1.11).](#_bookmark22) Для моделей классификации в качестве показателя близости используем значение сигмоидальной функции [(1.12)](#_bookmark24) класса реле- вантного документа, которая является выходом модели. Ранжирование при использовании языковой модели осуществляем по показателю косинусо- вой близости [(1.13)](#_bookmark26) между получаемыми от языковой модели векторными представлениями тестов запроса и документа. При построении рекоменда- ций дополняем основную разметку неразмеченными документами, которые будут использоваться только для разбора примеров рекомендаций и рас- ширенного описания характеристик алгоритмов.

Для полученных рекомендаций используем функционалы качества [(1.3)-](#_bookmark15) [(1.5)](#_bookmark16) для их оценки и сравнения между собой.

* + 1. Параметры задачи

Гипотезы [(1.1)-(1.2)](#_bookmark13) проверяются на уровне доверия 0.99. Оценка до- верительных интервалов оценок на разметке [(2.1)](#_bookmark39) проводится на уровне

доверия 0.95.

Для вычисления показателя близости текстов запроса и документа [(1.11)](#_bookmark22) в алгоритме ранжирования по сумме BM25 используются параметры *k* = 0.3, *b* = 0.

Параметры моделей рекомендаций представлены в соответствующих таблицах.

Таблица 2.1: Параметры kNN-классификатора

|  |  |  |
| --- | --- | --- |
| Параметр | k | p |
| Значение | 5 | 2 |

Таблица 2.2: Параметры LightGBM-классификатора

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | lr | leaves | depth | n |
| Значение | 0.1 | 31 | -1 | 100 |

Таблица 2.3: Параметры BERT-классификатора

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | epoch | batch | lr | L |
| Значение | 4 | 32 | 0.00002 | 0 |

Таблица 2.4: Параметры языковой модели

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | epoch | batch | lr | L |
| Значение | 1 | 1 | 0.00002 | 0 |

В качестве функции оптимизации для BERT-моделей используется AdamW, при этом скорость обучения линейно уменьшается после каждого шага обучения.

* + 1. Результаты численного эксперимента

Результаты проверки гипотез [(1.1)-(1.2)](#_bookmark13) приведены в приложении 1. Можно видеть, что для всех групп запросов гипотеза была отвергнута. При этом для ряда случаев статистика критерия значительно больше, чем квантиль соответствующего уровня.

Таким образом, по совокупности проверенных гипотез, алгоритм ра- ботает не одинаково для различных групп запросов, то есть настоящий алгоритм может быть улучшен и рассматриваемая в данной работе задача имеет смысл.

Результаты оценки доверительных интервалов на сформированной раз- метке представлены в приложении 2. Для всех групп запросов и реле- вантных документов погрешность оценок на сформированной разметке не должна превышать 5 процентных пункта, то есть оценки будут достаточно точными.

На этапе обучения оценка качества классификаторов производилась с помощью привычных функционалов качества для задачи классификации. В приложениях 4-9 представлены результаты классифкации на отложенной выборке для соответствующих моделей. Первый класс - класс документов, которые не подходят запросу. Второй класс - класс подходящих запросу документов.

Достаточно похожими по качеству классификации являются модели kNN (приложения 4-5) и LightGBM (приложения 6-7), хотя вторая модель по показателям и дает незначительный прирост качества.

Наиболее точная классификация получается с помощью классифика- тора BERT (приложения 8-9), однако при сравнении точности на обучаю- щей и валидационной выборке можно увидеть, что эта модель склонна к переобучению.

Таким образом, обученные классификаторы показывают достаточно неплохое качество на задаче классификации.

На рисунках [2.1,](#_bookmark42) [2.2](#_bookmark43) представлена оценка качества ранжирования до- кументов различными алгоритмами. Оценки были получены для различ- ных групп запросов с помощью функционалов [(1.3)-(1.5).](#_bookmark16)

Рис. 2.1 – Оценка качества лингвистических алгоритмов

По полученным результатам можно сделать следующие выводы по за- даче:

1. Лингвистические алгоритмы поднимаю преимущественно те доку- менты, которые были найдены текущим алгоритмом;
2. Семантические алгоритмы поднимают преимущественно незамечен- ные текущим алгоритмом документы, которые не были оценены пользова- телями ввиду своей недоступности;
3. Наилучшие результаты показывают BERT и LightGBM классифи- каторы;
4. Все новые алгоритмы по показателю правильности сортировки луч-



Рис. 2.2 – Оценка качества семантических алгоритмов

ше, чем текущий алгоритм;

1. Предложенные функционалы качества позволяют сравнивать меж- ду собой ранжирование алгоритмов, однако не является исчерпывающей характеристикой алгоритмов.

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе предлагается исследо- вание подходов к решению задачи оценки безусловной вероятности обслу- живания в терминах описанного игрового процесса.

Были выбраны и математически описаны модели движения участни- ков игры: преследуемый объект совершает движение в соответствии с урав- нениями движения в области с действующими силами тяжести и сопротив- ления воздуха, преследователь движется в соответствии с идеальным ме- тодом наведения на фактическое место встречи с маневрирующей целью. В основу моделирования случайного местоположения на плоскости лег- ли идеи метода Неймана, а также был разработан собственный алгоритм распределения случайных координат для сети дорог. Было произведено компьютерное моделирование игровой ситуации, и на основе результатов компьютерного моделирования с применением метода Монте-Карло были получены оценки безусловной вероятности обслуживания для данной по- становки задачи.

Численные результаты эксперимента показали, что оценка вероятно- сти, полученная с помощью метода Монте-Карло, сходится при стремле- нии количества реализаций к бесконечности. Также была подтверждена зависимость оценки вероятности обслуживания от параметров движения преследуемого.

Разработанный алгоритм имеет линейную зависимость времени рабо- ты от количества требуемых реализаций, что является одним из наиболее эффективных показателей оптимальности программы.

В работе приведены обзоры на различные способы решения проблемы для более сложных постановок задачи, и полученные результаты в пер-

спективе могут быть обобщены на более общие случаи.

ЛИТЕРАТУРА

1. Tie-Yan Liu Learning to Rank for Information Retrieval. — Foundations and Trends in Information Retrieval: Vol. 3: No 3, с. 225—331. — 2009. — 103 с.
2. Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. — Information Processing Management. 24(5): 513—523 с. — 1988.
3. Nick Craswell, Hugo Zaragoza, Stephen Robertson. In Proceedings of the Fourteenth Text REtrieval Conference. — Microsoft Cambridge at TREC- 14: Enterprise Track. (TREC 2005). — Gaithersburg, USA — 2005.
4. Polosukhin, Illia; Kaiser, Lukasz; Gomez, Aidan N.; Jones, Llion; Uszkoreit, Jakob; Parmar, Niki; Shazeer, Noam; Vaswani, Ashish. Attention Is All You Need. — 2017. — 15 c.
5. Daniel T. Larose. Discovering Knowledge in Data: An Introduction to Data Mining. — 2004. — 90-106 c.
6. Sagi, Omer; Rokach, Lior. Approximating XGBoost with an interpretable decision tree. — Information Sciences. 572 — 2021. — 522-542 c.
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. — 2019. — 16 c.
8. Ивченко Г. И., Медведев Ю. И. Введение в математическую статистику.

— М.: Издательство ЛКИ — 2010. — 600 c.

1. А. И. Кибзун, Е. Р. Горяинова, А. В. Наумов, А. Н. Сиротин ТЕОРИЯ ВЕРОЯТНОСТЕЙ И МАТЕМАТИЧЕСКАЯ СТАТИСТИКА. БАЗО-

ВЫЙ КУРС С ПРИМЕРАМИ И ЗАДАЧАМИ – М.: ФИЗМАТЛИТ, 2002. — 224 с.

1. Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, Wei Wang Language-agnostic BERT Sentence Embedding. — Google AI Mountain View — 2020. — 14 c.

ПРИЛОЖЕНИЯ

Приложение 1.

Оценка работоспособности системы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группировка | Статистика критерия | p- value | Гипотеза | Вердикт |
| Прямая оценка | | | | |
| По сфере запроса | 7.25 | 0 | Отвергнута | Для запросов экономи- ческого характера поло- жительная оценка более вероятна |
| По длине запроса | 23.41 | 0 | Отвергнута | Для длинных запросов положительная оценка значительно более веро- ятна |
| По длине запроса  в эконо- мической  сфере | 10.09 | 0 | Отвергнута | Для длинных запросов положительная оценка более вероятна |
| По длине запроса  в сфе-  ре общей юрисдик- ции | 20.33 | 0 | Отвергнута | Для длинных запросов положительная оценка значительно более веро- ятна |
| Косвенная оценка | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| По сфере запроса | 95.39 | 0 | Отвергнута | Для запросов эконо- мического характера открытие рекоменда- ции существенно более вероятно |
| По длине запроса | 40.19 | 0 | Отвергнута | Для длинных запросов открытие рекомендации существенно более веро- ятно |
| По длине запроса  в эконо- мической  сфере | 8.82 | 0 | Отвергнута | Для длинных запросов открытие рекомендации более вероятно |
| По длине запроса  в сфе-  ре общей юрисдик- ции | 44.53 | 0 | Отвергнута | Для длинных запросов открытие рекомендации существенно более веро- ятно |

Приложение 2.

Оценка погрешности при оценивании доли запросов с правильной сортировкой

|  |  |  |
| --- | --- | --- |
| Запросы | Объем раз- метки | Погрешность (п.п.) |
| Все | 744 | +-2.93 |
| В сфере экономической дея- тельности | 447 | +-3.59 |
| В сфере общей юрисдикции | 297 | +-4.41 |
| Длинные запросы | 398 | +-3.35 |
| Короткие запросы | 346 | +-3.86 |

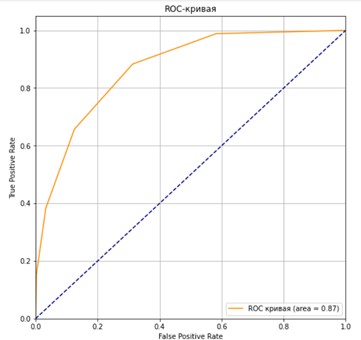
Приложение 3.

Оценка погрешности при оценивании доли поднятых релевантных документов

|  |  |  |
| --- | --- | --- |
| Документы | Объем раз- метки | Погрешность (п.п.) |
| Релевантные документы | 2726 | +-1.56 |

Приложение 4.

ROC-кривая качества классификации kNN-классификатора



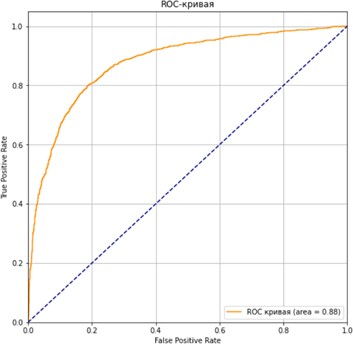
Приложение 5.

Показатели качества классификации kNN-классификатора

|  |  |  |  |
| --- | --- | --- | --- |
|  | Точность | Полнота | F1 |
| Первый класс | 0.81 | 0.87 | 0.84 |
| Второй класс | 0.76 | 0.66 | 0.7 |
| Среднее по классам | 0.78 | 0.77 | 0.77 |
| Всзвешенное среднее по классам | 0.79 | 0.79 | 0.79 |

Приложение 6.

ROC-кривая качества классификации LightGBM-классификатора



Приложение 7.

Показатели качества классификации LightGBM-классификатора

|  |  |  |  |
| --- | --- | --- | --- |
|  | Точность | Полнота | F1 |
| Первый класс | 0.79 | 0.93 | 0.85 |
| Второй класс | 0.83 | 0.58 | 0.68 |
| Среднее по классам | 0.81 | 0.75 | 0.77 |
| Всзвешенное среднее по классам | 0.8 | 0.8 | 0.79 |

Приложение 8.

Значение функции потерь при обучении BERT-классификатора



Приложение 9.

Показатели точности классификации BERT-классификатора

|  |  |
| --- | --- |
| Выборка | Точность классификации |
| Валидационная | 0.95 |
| Отложенная | 0.88 |