МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

Институт дополнительного образования

Высшая инженерная школа

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

СОЗДАНИЕ ЧАТ-БОТА, ОПРЕДЕЛЯЮЩЕГО СРОК ХРАНЕНИЯ ДОКУМЕНТА

по программе профессиональной переподготовки:

«Анализ данных на языке Python»

Выполнил(а):

Власова Татьяна Григорьевна

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель:

Кандидат экономических наук,

доцент по научной специальности

«Математические и инструментальные

методы экономики»,

Заграновская Анна Васильевна

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2022

СОДЕРЖАНИЕ

[Введение 3](#_Toc107003275)

[1. Обзор предметной области 5](#_Toc107003276)

[2. Постановка цели и задач 9](#_Toc107003277)

[3. Моделирование 11](#_Toc107003278)

[4. Реализация 20](#_Toc107003279)

[5. Анализ результатов, тестирование 24](#_Toc107003280)

[ЗАКЛЮЧЕНИЕ 29](#_Toc107003281)

[Список использованных источников 31](#_Toc107003282)

# Введение

Выпускная квалификационная работа выполнена на 46 листах, содержит 1 таблицу, 15 рисунков, 11 использованных источников и 3 приложения.

Темой работы является создание чат-бота, определяющего сроки хранения документов.

Актуальность темы обусловлена производственной необходимостью знать и учитывать сроки хранения документов, которые возникают в процессе деятельности компании, в связи с тем, что несоблюдение правил хранения документов может привести к налоговым и уголовным последствиям.

Целью работы является разработка вспомогательного инструмента, подсказывающего рекомендуемый срок хранения документа, для сотрудников, не обладающими специальными знаниями и навыками в сфере документооборота.

Для достижения этой цели был выполнен ряд задач:

- собраны данные о фонде организации, на документах которой была обучена модель. Так же была построена тематическая модель для дальнейшей интеграции в чат-бот;

- проведен разведочный анализ собранных данных, для определения того, документы с какими сроками хранения образуются в ходе деятельности организации;

- выбрана лучшая модель, с точностью более 80% для решения задачи классификации документов по срокам хранения.

- создан чат-бот, в качестве инструмента, которым сможет воспользоваться любой сотрудник организации.

В структуре работы можно выделить несколько составных частей:

- в части «Обзор предметной области» обосновывается актуальность темы исследования, формулируется проблема и предлагаются варианты ее решения;

- в части «Постановка цели и задач» определяются цель и задачи работы, разрабатываются требования к создаваемому инструменту;

- в части «Моделирование» описывается сбор данных, строится тематическая модель для определения тематик документов, описываются результаты разведочного анализа и выбирается лучшая модель для решения задачи классификации документов по их срокам хранения;

- в части «Реализация» создается чат бот, в который встроена выбранная модель классификации;

- в части «Анализ результатов, тестирование» - оценивается выполнение поставленных цели и задач, на ряде примеров анализируется работа созданного чат-бота.

Предполагается дальнейшее практическое использование результатов работы, дополненных и расширенных, в том числе с помощью тематической модели, после согласования доступа к электронному архиву со стороны службы безопасности.

# Обзор предметной области

* 1. Обзор ситуации

В настоящий момент для успешного решения бизнес-задач необходима высокая скорость обмена информацией, а это означает, что документооборот, должен быть соответствующим образом структурирован и регламентирован. Несмотря на постепенный переход к электронному документообороту, на данный момент большая часть информации продолжает передаваться на бумажных носителях.

Организация хранения документов регламентируется Приказом Министерства культуры РФ от 31 марта 2015 г. № 526 "Об утверждении правил организации хранения, комплектования, учёта и использования документов Архивного фонда Российской Федерации и других архивных документов в органах государственной власти, органах местного самоуправления и организациях", а вот сроки хранения всевозможных документов на бумажных носителях регламентируются целым списком различных нормативных документов:

- «Перечень типовых управленческих архивных документов, образующихся в процессе деятельности государственных органов, органов местного самоуправления и организаций, с указанием сроков хранения» (утв. Приказом Росархива от 20.12.2019 № 236),

- ч.4 ст. 7 № 115-ФЗ «О противодействии легализации (отмыванию) доходов, полученных преступным путем, и финансированию терроризма» от 07.08.2001,

- ст.22 № 125-ФЗ «Об архивном деле в Российской Федерации» от 22.10.2014(ред. от 11.06.21)

- и т.д.

В зависимости от специфики деятельности организации сроки хранения документации могут так же регламентироваться:

- письмом Минздрава России от 7 декабря 2015 г № 13-2/1538 «О сроках хранения медицинской документации»,

- ч.15 ст.4 № 44-ФЗ «О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд» от 05.04.2013(ред. от 16.04.22)

- и т.д.

В организациях, в деятельности которых создаются документы Архивного фонда Российской Федерации, на основе типовых ведомственных и иных перечней разрабатывается номенклатура дел и согласуется с соответствующим государственным/муниципальным архивом. В номенклатуре указываются все документы, которые заведены или могут быть заведены в организации и сроки их хранения. Те же организации, которые не являются источниками комплектования Архивного фонда, номенклатуру дел могут составить самостоятельно, но чаще всего этого не делают, потому что процесс составления номенклатуры длительный и сложный. В результате, в ходе деятельности рядового сотрудника любой более-менее крупной компании, не имеющей номенклатуры дел, образуется огромное количество документов на бумажных носителях, о сроках хранения которых он не имеет представления.

* 1. Формулирование проблемы.

Для сотрудников организаций, не являющихся источниками комплектования государственных и муниципальных архивов и не имеющих актуальных и проработанных номенклатур, определение срока хранения документа является нетривиальной задачей. Сотрудник, не работающий на постоянной основе с архивным законодательством, не знает к какому нормативному акту обратиться, чтобы узнать срок хранения документа. Кроме того, даже если верный нормативный акт был определен, точно узнать по нему срок хранения документа не всегда бывает легко. Так, например, в «Перечне типовых управленческих архивных документов, образующихся в процессе деятельности государственных органов, органов местного самоуправления и организаций, с указанием сроков хранения» (далее – перечень) есть статья 11, в которой определяется срок хранения в 5 лет для договоров, соглашений, контрактов и документов к ним, не указанных в отдельных статьях перечня. Иными словами, для того, чтобы воспользоваться этой статьей, нужно знать все остальные статьи перечня, в которых идет речь о контрактах и договорах. Кроме того, существуют документы, не имеющие названий, по которым можно было бы осуществить алфавитный поиск. Например, реестр формы Ф-5, предназначенный для вручения отправления получателю от ФГУП ГЦСС, этот документ образуется в ходе деятельности любой организации, пользующейся услугами ФГУП ГЦСС, но в перечне такого документа нет. Тем не менее этот документ может являться бланком строгой отчетности и имеет срок хранения не менее 5 лет.

Неверное определение сроков хранения документов может привести к нежелательным последствиям как непосредственно для сотрудника, так и для организации в целом: отсутствие первичных учетных документов и документов кадрового учета часто влечет за собой налоговые последствия и административную ответственность, утрата документов, содержащих государственную тайну, согласно статье 284 УК РФ, наказывается ограничением свободы на срок до трех лет.

Таким образом, в рамках документооборота организации задача по определению сроков хранения документации должна быть разрешена в соответствии с законодательством.

* 1. Потенциальные пути решения проблемы

Существуют различные способы контролировать положенный срок хранения документов.

Самый регламентированный вариант – создание номенклатуры дел. Способы и особенности создания номенклатур подробно описаны в приказе Минкультуры России «Об утверждении правил организации хранения, комплектования, учета и использования документов Архивного фонда Российской Федерации и других архивных документов в органах государственной власти, органах местного самоуправления и организациях» № 526 от 31.03.2015, однако, они более применимы к государственным организациям, нежели к коммерческим. Кроме того, создание номенклатуры дел не только требует специалиста, обладающего соответствующими навыками и компетенциями, но и устоявшейся и строгой системы документооборота в компании, когда в конце одного делопроизводственного года можно точно сказать какие документы и в каком объеме будут созданы в следующем делопроизводственном году.

Второй вариант, которым иногда пользуются в небольших организациях – это полный запрет на уничтожение любой документации без согласования со специально созданной комиссией. Такой подход не оправдан для организаций с большим потоком документов и хотя бы несколькими филиалами, так как требует либо слишком частого созыва комиссии, либо очень больших площадей для хранения всех документов, ожидающих списания.

Третий вариант – обучение основам делопроизводства каждого сотрудника, так же применим только в маленьких компаниях с низкой ротацией кадров.

Чаще всего в больших компаниях используют четвертый вариант – создание специального архивного отдела, который отвечает за хранение и систематизацию документации, составляющей фонд организации, и уничтожает документы с истекшими сроками хранения. Это удобное и универсальное решение, но тем не менее площади таких архивов тоже не бесконечны, а передача каждого черновика или копии из удаленного филиала не оправдана. Соответственно, хорошим вспомогательным инструментом для архивного отдела, может являться справочник, подсказывающий пользователю срок хранения документа.

# Постановка цели и задач

* 1. Цель и задачи работы

Целью работы является разработка вспомогательного инструмента, подсказывающего рекомендуемый срок хранения документа, для сотрудников, не обладающими специальными знаниями и навыками в сфере документооборота.

Для достижения этой цели необходимо решить ряд задач:

- собрать данные о фонде организации;

- провести разведочный анализ собранных данных, чтобы понять, документы с какими сроками хранения образуются в ходе деятельности организации;

- выбрать лучшую модель для решения задачи классификации документов по срокам хранения. Точность выбранной модели должна быть не менее 80%;

- создать инструмент, которым сможет воспользоваться любой сотрудник организации.

* 1. Требования к разрабатываемому инструменту

Для создания инструмента справочника, нужно понимать, что в рамках бизнес-процесса в компании есть сотрудники, у которых возникают одни и те же сложности с определением срока хранения документа. За решением этого вопроса они обращаются в архивный отдел, к своему руководству или действуют наугад. Соответственно, существует высокий риск ошибок из-за человеческого фактора, большая нагрузка на архивный отдел и высокая цена за ошибку при неверном определении срока хранения.

Таким образом, к создаваемому в рамках работы инструменту – справочнику, определяющему сроки хранения, можно сформулировать ряд требований (см. таблицу 1).

Таблица 1

Требования к разрабатываемому инструменту

|  |  |
| --- | --- |
| **Параметр** | **Характеристика** |
| Тематика | Созданный справочник должен давать ответ на вопрос, сколько рекомендуется хранить тот или иной документ |
| Целевая аудитория | Сотрудники организации, не обладающими специальными знаниями в области архивного законодательства |
| Основные пользовательские функции | Решение задачи по определению сроков хранения, без необходимости просмотра пользователем всех возможных нормативных документов. |
| Дизайн | Удобный, интуитивно-понятный пользователю интерфейс справочника |

Как следует из таблицы 1, требуется простой, удобный инструмент, предсказывающий срок хранения документа, которым сможет воспользоваться любой сотрудник организации.

# Моделирование

* 1. Сбор данных

Самой первой задачей является сбор данных о фонде организации. Фонд – совокупность всех документов, образующихся в деятельности организации[[1]](#footnote-1). Документы в любой организации делятся по срокам хранения на: документы временного хранения, долговременного хранения и постоянного хранения. Соответственно, чтобы выполнить задачу по сбору данных о фонде организации, нужно рассмотреть виды документов, имеющиеся в организации и сроки их хранения. Однако, для дальнейшего построения модели, решающей задачу классификации, нужно сформулировать не просто список объектов (видов документов) и групп к которым они относятся (сроков хранения), но и набор признаков, от которых зависит отнесение вида документа к тому или иному сроку хранения.

Исходя из ч.4 ст. 7 № 115-ФЗ и примечаний к статьям Перечня, в качестве таких признаков обязательно должны выступать:

- место формирования документа. Если документ создан в ходе деятельности организации – то он регламентируется статьями перечня, если же документ запрошен у контрагента, то он чаще всего регламентируется ч.4 ст. 7 № 115-ФЗ и имеет срок хранения меньший, чем указан в соответствующей статье перечня.

- наличие не разрешенных споров/разногласий по вопросам, затронутым в документе. Если документ имеет отношение к вопросу, разбираемому в рамках спора, то документ должен быть сохранен до разрешения разногласий, независимо от того, какой срок хранения этого документа установлен законодательно.

- наличие в организации вредных\опасных условий труда или условий, приравненных к ним. Этот признак очень важен для кадровых документов, так как он меняет их срок хранения с 5 лет на 75.

- периодичность отчетности. Этот признак принципиален для любого рода обязательной отчетности. Статистические и бухгалтерские отчеты с годовой периодичностью хранятся постоянно, тогда как квартальная или помесячная отчетность имеет срок хранения до 5 лет включительно.

Кроме того, к признакам документов нужно отнести и тему или сферу деятельности, к которой относится документ. В любой организации, независимо от рода ее деятельности, всегда будут заводиться: организационно-распорядительные документы, кадровые документы и обязательная отчетность. Определить прочие темы документов можно:

- вручную, ознакомившись с бизнес-процессом организации, как было сделано при сборе набора данных, в дальнейшем использованном в данной работе.

- воспользовавшись тематическим моделированием. Для примера, на сайте Санкт-Петербургского государственного экономического университета, выложены в открытый доступ документы, связанные с деятельностью университета, в формате .pdf. Если преобразовать эти документы в формат .txt, то на их коллекции можно попробовать построить модель, определяющую, к каким темам относится каждый документ и из каких слов состоит каждая тема. В качестве такой модели можно использовать латентное размещение Дирихле, предложенное Д. Блеем. При использовании этой модели «количество тем фиксировано изначально и задаётся в явном виде»[[2]](#footnote-2). Соответственно, можно выделить следующие этапы для создания модели:

1. Создание корпуса

1.1. Загрузка документов с сайта

1.2. Преобразование их из формата .pdf в формат .txt

1.3. Создание датасета с текстами

2. Предобработка текста

2.1 Удаление стоп слов и пунктуации

2.2. Выделение устойчивых оборотов

2.3 Лемматизация или приведение слов к нормальной форме

3. Подбор оптимального количества тем

4. Создание модели

Для определения оптимального количества тем была использована метрика когерентности. «Тема называется когерентной, если её топ-слова встречаются вместе чаще, чем можно было бы ожидать от случайного распределения.»[[3]](#footnote-3)

В результате построения тематической модели было выявлено 8 тем, к которым относятся документы с сайта Санкт-Петербургского государственного экономического университета были распределены на 8 тем:

1. организационно-правовые документы,
2. документы о приеме,
3. документы по экзаменам,
4. документы по научной деятельности,
5. документы по учебным программам,
6. развитие университета,
7. деятельность филиалов,
8. о коррупции.

Код для определения тем документов, находящихся в открытом доступе, можно увидеть в приложении №1.

Использовать документы с сайта Санкт-Петербургского государственного экономического университета было удобно для построения тематической модели, но прогнозирование сроков хранения на них, к сожалению, бессмысленно, так как все документы на сайте являются документами постоянного хранения. Использование же тематической модели на документах лизинговой компании, которые имеют большее разнообразие сроков хранения, не было согласовано службой безопасности компании.

Так же к признакам документа можно отнести отдел, в ходе деятельности которого был создан документ и вид документа по признаку сферы использования, определенный спецификой деятельности организации.

Результативным признаком, который в дальнейшем, будет предсказывать построенная модель, должен являться срок хранения документа. В собираемых данных этот срок хранения определяется на основании нормативных документов.

В Перечне 657 статей, соответственно законодательно выделено 657 видов документов на настоящий момент. Однако, большая часть этих видов документов может изменять свой срок хранения с изменением одного из признаков: например, в статье 277 вид «первичные учетные документы и связанные с ними оправдательные документы» меняет свой срок хранения с 5 лет на «до принятия решения по делу», если существуют какие-либо споры или разногласия. А это значит, что такие виды документов, нужно будет вносить в набор данных с каждым новым составом признаков и соответственным им сроком хранения. В данном случае проще сформировать выборку: согласно А.Д. Наследову «ограниченную по численности группу объектов, специально отбираемую из генеральной совокупности для изучения ее свойств»[[4]](#footnote-4).

Таким образом, проанализировав документарный фонд лизинговой компании с использованием информации из книг учета поступлений документов в архив, можно выбрать самые часто встречающиеся виды документов и получить набор данных, состоящий из 185 объектов, каждый из которых обладает рядом номинальных признаков: место формирования, наличие спора, наличие вредности, принадлежность к отчетности с годовой периодичностью, сфера деятельности (тема), отдел.

* 1. Разведочный анализ

Имея набор данных, состоящий из объектов и их номинальных признаков, нельзя использовать такие меры центральной тенденции как среднее или медиана. Однако, можно обратиться к такому значению в множестве наблюдений, которое встречается наиболее часто – к моде (См. рисунок 3.1).[[5]](#footnote-5)

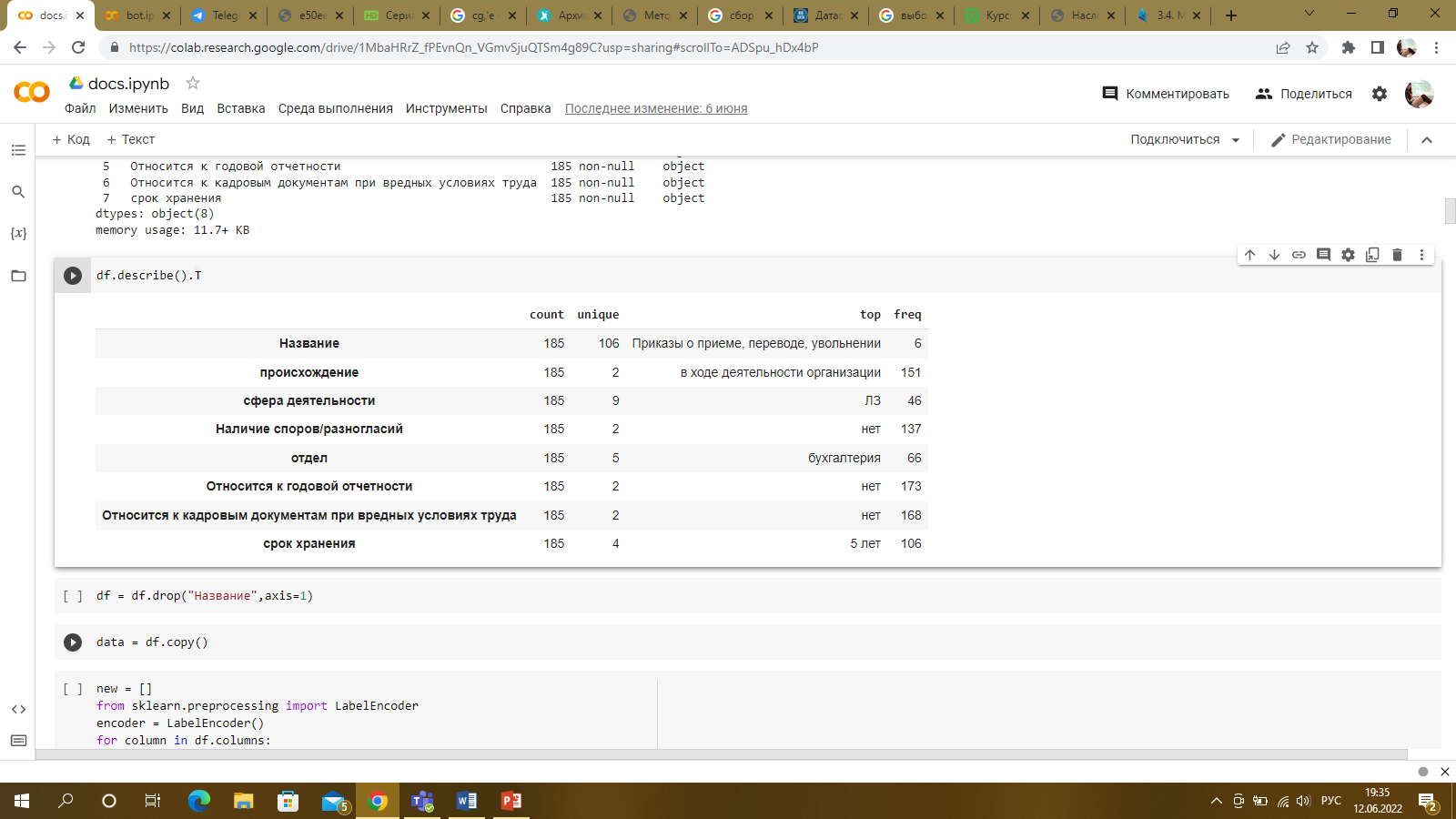


Рисунок 3.1 – Информация о количестве значений

Из рисунка видно, что документов со сроком хранения 5 лет больше, чем любых других. И соотношение количества документов для каждого из классов явно говорит о несбалансированности классов (См. рисунок 3.2).

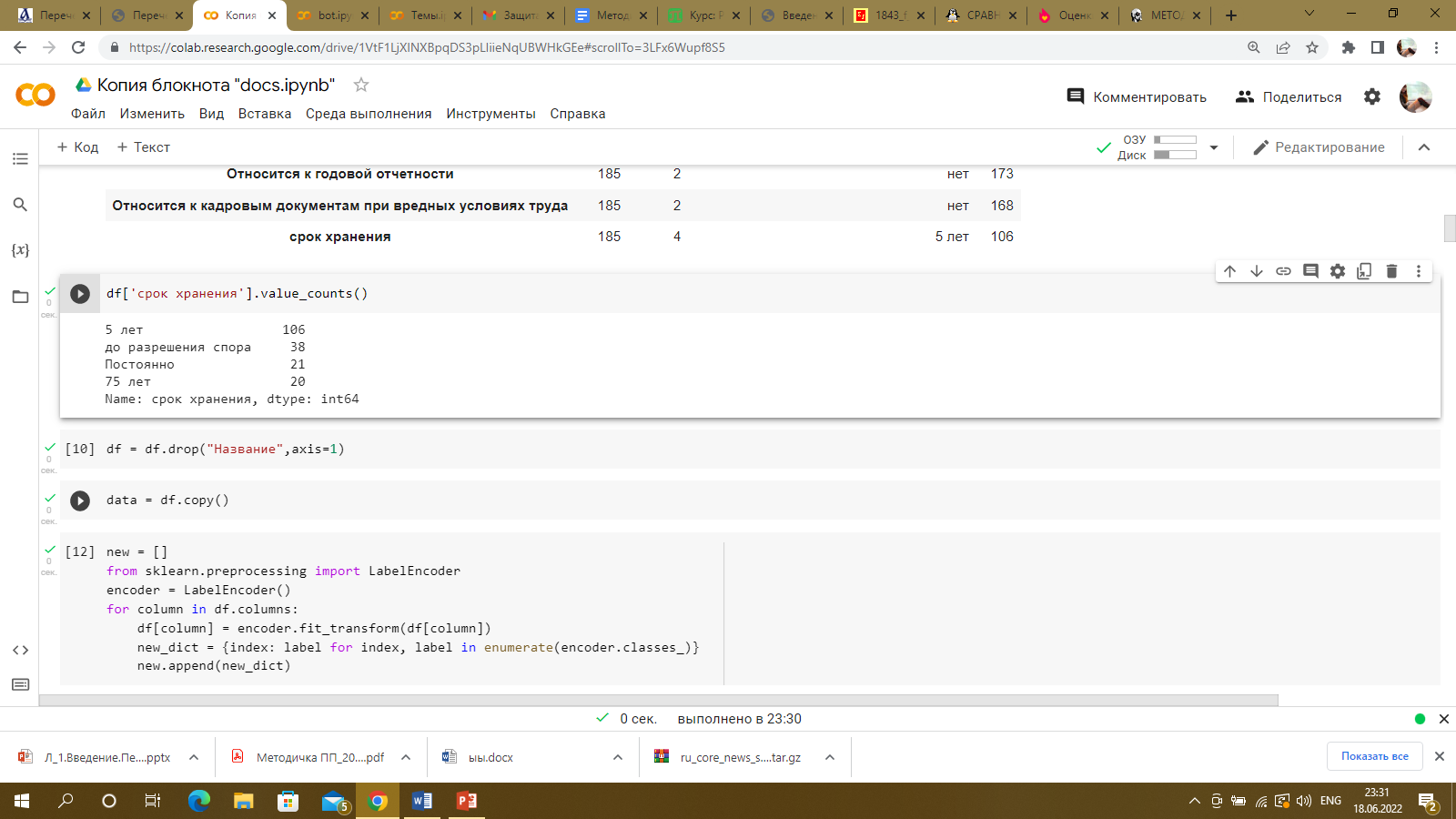


Рисунок 3.1 – Количество документов по срокам хранения

Из рисунка 3.2. видно, что в результативном признаке «Срок хранения» уникальных значений значений 4, соответственно, нужно построить модель относящую документ по ряду номинальных признаков к одной из 4 групп по сроку хранения.

* 1. Лучшая модель

Разделение рассматриваемой совокупности объектов на группы называется классификацией[[6]](#footnote-6). Соответственно, построенная модель должна решать задачу классификации с обучающей выборкой в отнесении документа по ряду номинальных признаков (происхождение, сфера деятельности, наличие споров/разногласий, отдел, относится к годовой отчетности, относится к кадровым документам при вредных условиях труда) к определенной группе по сроку хранения (5 лет, 75 лет, постоянно, до разрешения споров/разногласий):

Чаще всего для решения задач классификации используются следующие методы:

- Метод k-го ближайшего соседа (kNN).

- Логистическая регрессия (LR).

- Линейный метод опорных векторов (SVM).

- Линейный дискриминантный анализ (LDA).

- Наивный байесовский классификатор (Naive Bayes Classifier).

- Деревья решений для классификации (CART).

Используя кросс-валидацию, метод разделения по фолдам (на 7 в данном случае), можно сравнить долю правильных ответов, которую дают эти методы (См. рисунок 3.3).

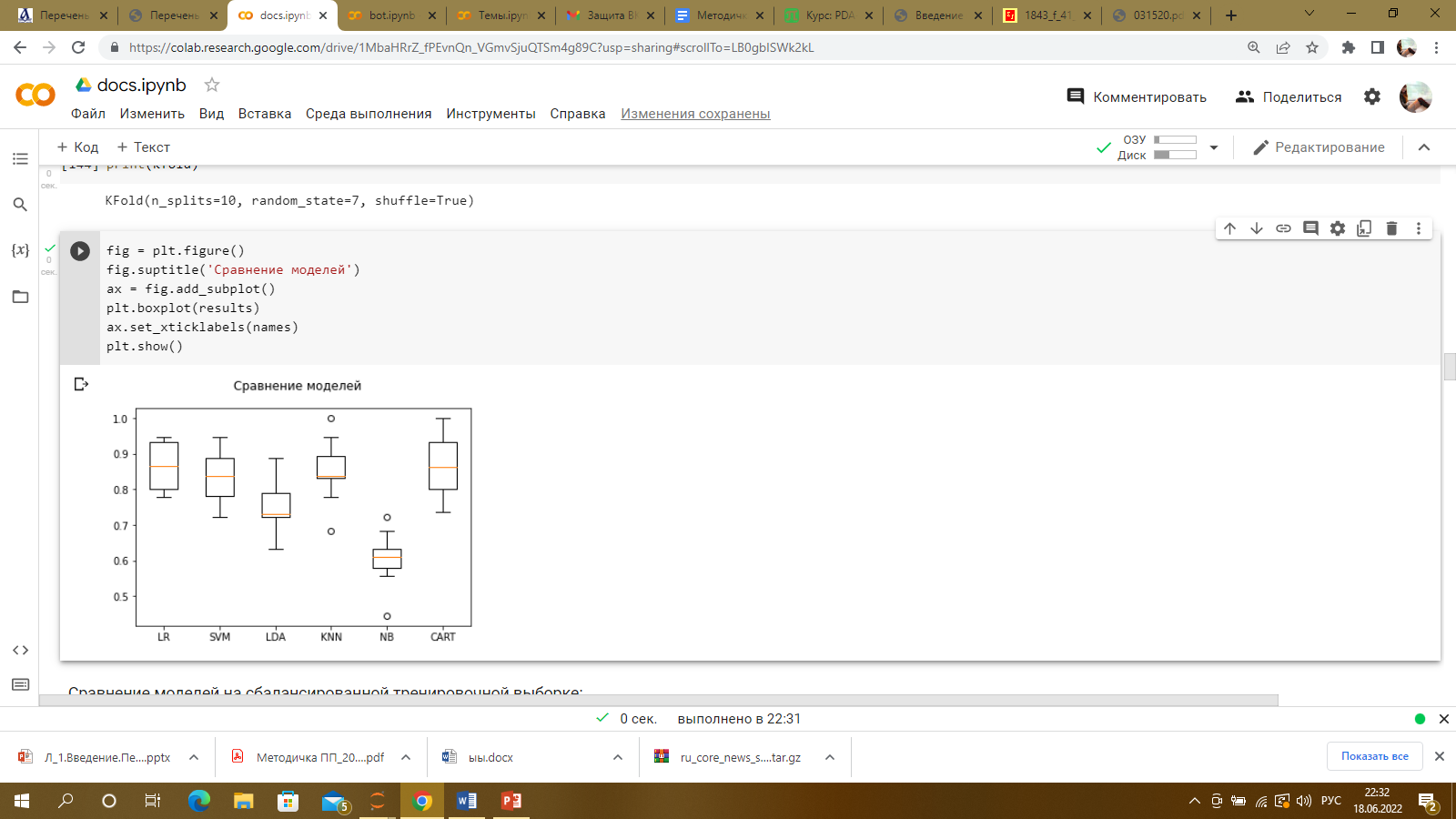


Рисунок 3.2 – Сравнение моделей

Из рисунка 3.3 видно, что лучшие результаты дают метод k-го ближайшего соседа, логистическая регрессия и деревья решений. Однако, учитывая несбалансированность классов, стоит попробовать обучить эти модели на сбалансированной тренировочной выборке, в которой разница между классами нивелирована за счет случайного удаления наблюдений из больших классов.

Если желаемая точность в 80% не будет достигнута после применения этих моделей, необходимо будет использовать ансамблевые методы, объединяющие в себе слабые модели.

Метод k-го ближайшего соседа относится к метрическим методам. Объект относится к тому классу, к которому чаще всего принадлежат соседи данного объекта. Этот метод достаточно прост в реализации и интерпретации, но при большом количестве соседей могут возникнуть сложности в его применении.[[7]](#footnote-7)

Доля правильных ответов при использовании на тестовой выборке метода ближайших соседей, обученного на сбалансированной тренировочной выборке, составляет 55 %.

Логистическая регрессия предсказывает наиболее вероятный класс объекта. Этот метод достаточно точен, поскольку «определяет только условное распределение, но никак не зависит от безусловного распределения в X»[[8]](#footnote-8). Доля правильных ответов при использовании на тестовой выборке метода логистической регрессии, обученного на сбалансированной тренировочной выборке, составляет 46 %.

Деревья решений – метод с высокой производительностью обучения и прогнозирования, его легко визуализировать и интерпретировать. Дерево решений представляет собой иерархическую древовидную структуру, где каждый узел содержит условие дальнейшего ветвления по одному из признаков, а количество ветвлений определяется количеством значений, которые может принять данный признак.[[9]](#footnote-9) Доля правильных ответов при использовании на тестовой выборке метода деревьев решений, обученных на сбалансированной тренировочной выборке, составляет 55 %.

Использованные модели не могут быть выбраны в рамках поставленной в работе задачи, так как их точность ниже 80%. Соответственно нужно обратиться к ансамблевым методам.

В статье, посвященной классификации несбалансированных данных, Y. Sun, A.K. Wong, M.S. Kamel среди прочих методов обработки несбалансированных данных рекомендуют использовать бустинг, так как «бустинговые алгоритмы являются итеративными алгоритмами, которые размещают различные веса в тренировочном распределении на каждой итерации»[[10]](#footnote-10). Иными словами, тем экземплярам, которые улучшают обучение, присваивается больший вес, чем тем, которые не улучшают. Соответственно можно построить градиентный бустинг 100 деревьев решений, чтобы каждое следующее уменьшало функцию потерь. Такой ансамбль, обученный на несбалансированной тренировочной выборке, дает 94% правильных ответов при использовании на тестовой выборке. Этот результат соответствует поставленной задаче.

Таким образом, лучшей моделью является градиентный бустинг - ансамбль слабых предсказывающих моделей (деревьев решений). Его точность выше 80%, как и требовалось в задаче. Код с созданием всех перечисленных в этом параграфе моделей можно увидеть в Приложении 2.

# Реализация

После того, как выбрана наиболее подходящая модель для определения срока хранения документа по ряду номинальных признаков, необходимо перейти к созданию инструмента-справочника.

Согласно сформулированным ранее требованиям, инструмент должен определять срок хранения документа и быть простым и понятным в использовании. В качестве такого инструмента был выбран чат-бот, поскольку это средство коммуникации достаточно доступно и предоставляет мгновенный ответ пользователю.

Открывая чат-бот, пользователь имеет на руках документ, срок хранения которого нужно определить исходя из имеющейся у пользователя информации. Соответственно, после нажатия кнопки начать бот должен предупредить пользователя, что тому придется ответить на несколько вопросов. Далее бот задает пользователю ряд заранее записанных вопросов:

- укажите происхождение документа

- к какой сфере деятельности относится документ?

- ведутся ли споры/разбирательства?

- отнесены ли условия труда работников, указанных в этом документе, к вредным или опасным?

- это годовой отчет?

- из какого вы отдела?

Однако, сценарий диалога с ботом должен строиться по-разному, в зависимости от выбранной сферы деятельности, к которой относится документ. Вопрос о вредности оправдан, только в случае если пользователь определяет срок хранения кадрового документа, равно как вопрос о периодичности отчетности имеет смысл только при рассмотрении документов, относящихся к обязательной отчетности (См. рисунок 4.1).

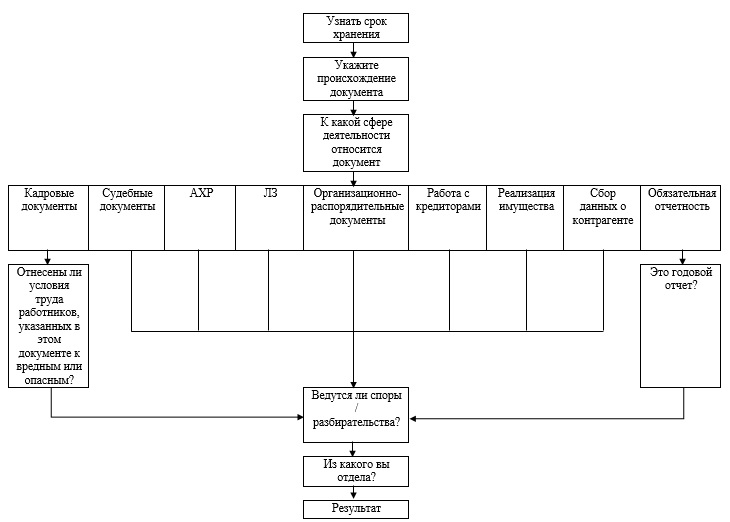


Рисунок 4.1 – Структура чат-бота

Из рисунка видно, что сценарий диалога одинаков для тематик документов: «Судебные документы», «АХР», «ЛЗ», «Организационно-распорядительные документы», «Работа с кредиторами», «Реализация имущества», «Сбор данных о контрагенте». Для «Кадровых документов», как и для «Обязательной отчетности» сценарии уникальные.

На каждом этапе работы с ботом пользователь может нажать кнопку назад, если он передумал определять срок хранения документа или понял, что неправильно ответил на вопросы. В конце диалога с ботом пользователю пишется рекомендуемый срок хранения документа.

Вопросы, задаваемые ботом, соответствуют тем признакам видов документов, которые были определены на этапе сбора данных о фонде организации. При дальнейшем использовании бота или настраивании его для новых отделов и сфер деятельности организации можно будет так же воспользоваться тематическим моделированием.

Для удобства пользования и обработки полученных данных варианты ответов на каждый вопрос для пользователя представлены в виде кнопок. Ответ пользователя на каждый вопрос должен быть сохранен и передан в обученную модель, которая на основе полученных данных определит срок хранения документа.

Бот был создан для кроссплатформенной системы мгновенного обмена сообщениями – Telegram с использованием библиотеки aiogram. Из нее импортировались классы ReplyKeyboardMarkup и KeyboardButton. KeyboardButton нужен для описания отдельных кнопок клавиатуры. ReplyKeyButtonMarkup создает клавиатуру из кнопок с помощью метода add.

Каждый ответ пользователя на вопрос бота сохраняется и затем передается в заранее сохраненную обученную модель, загруженную с помощью библиотеки pickle. Спрогнозированное моделью значение класса документа (его срока хранения) расшифровывается с помощью словаря и передается пользователю. Код бота можно увидеть в Приложении 2, интерфейс бота – на рисунке 4.2.

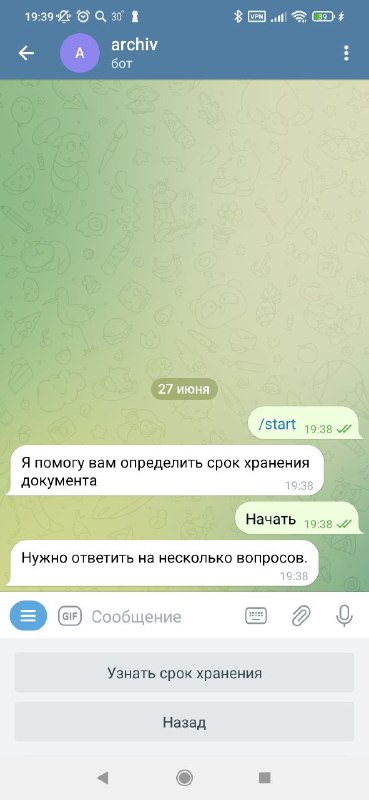
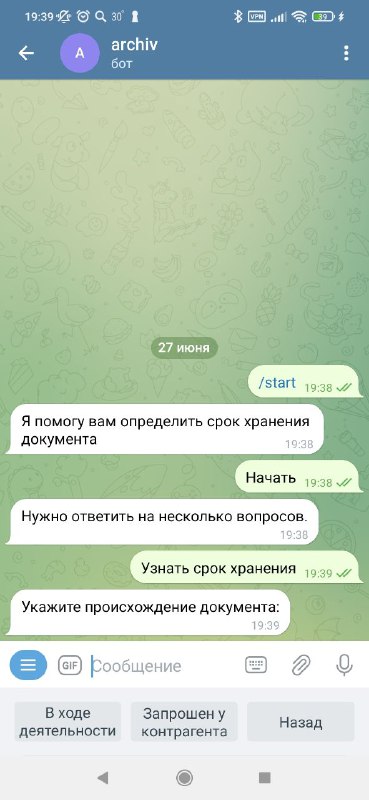


Рисунок 4.2. Интерфейс чат-бота

Как видно из рисунка 4.2, бот очень прост в использовании и не требует от пользователя обращения к специальным нормативным документам.

Таким образом, в качестве инструмента, определяющего срок хранения документа, которым может воспользоваться любой сотрудник организации, был выбран чат-бот для мессенджера Telegram. Бот задает пользователю ряд вопросов, ответы на которые являются значениями признаков, которые затем передаются в модель для определения документа к одному из 4 классов.

# Анализ результатов, тестирование

Для достижения цели работы – создание инструмента-справочника, определяющего сроки хранения документов, был поставлен ряд задач.

Первая задача – сбор данных о фонде организации, была выполнена с использованием книг учета поступлений документов в архив организации, ч.4 ст. 7 № 115-ФЗ и примечаний к статьям Перечня. В результате был составлен список самых часто встречающихся в компании документов с указанием их сроков хранения и ряда признаков, определяющих именно этот срок.

Вторая задача – разведочный анализ собранных данных, проведенный на собранной информации. По итогам выполнения этой задачи были определены 4 класса (срока хранения), к которым относятся документы организации: 5 лет, 75 лет, постоянно, до разрешения споров/разногласий.

Третья задача - выбрать лучшую модель для решения задачи классификации документов по срокам хранения. С учетом того, что пришлось проводить многоклассовую классификацию данных, для которых характерен дисбаланс классов, лучшей моделью стал ансамбль – градиентный бустинг. Точность этой модели, обученной на тренировочной выборке и примененной к тестовой, составила 94%, что соответствует желаемому результату – больше 80%.

Четвертой и последней задачей было создание инструмента, которым может воспользоваться любой сотрудник организации. В качестве такого инструмента был написан чат-бот для Telegram.

Для проверки работоспособности инструмента нужно предложить ему определить сроки хранения документов, сроки которых точно известны.

Первый класс – это документы со сроком хранения 5 лет. Согласно статье 277 Перечня, таким документом, при условии, что по нему не ведется никаких споров, является акт о приеме имущества/материалов, который часто образуется в ходе деятельности административно-хозяйственного отдела любой организации. Соответственно, ответ на вопрос бота о происхождении документа должен быть: «в ходе деятельности организации», о сфере деятельности – «Административно-хозяйственные расходы (АХР)», о спорах – «Споров нет /спор завершен», об отделе – «АХО». Ответ бота должен быть: «Рекомендуемый срок хранения: 5 лет» (см. рисунок 5.1).



Рисунок 5.1 – Чат с ботом по акту о приеме имущества/материалов

Как видно на рисунке 5.1 ответ бота совпадает с ожидаемым.

Второй класс – документы постоянного хранения. Согласно статье 268 Перечня, таким документом, при условии, что по нему не ведется никаких споров, является годовой бухгалтерский баланс, который ежегодно обязательно формируется бухгалтерией организации. Соответственно, ответ на вопрос бота о происхождении документа должен быть: «в ходе деятельности организации», о сфере деятельности – «Обязательная отчетность», о периодичности – «годовой/за несколько лет», о спорах – «Споров нет /спор завершен», об отделе – «бухгалтерия». Ответ бота должен быть: «Рекомендуемый срок хранения: Постоянно».

Так же на этом этапе стоит рассмотреть определенную, сложность, которая может возникнуть при определении срока хранения бухгалтерского баланса, запрошенного у клиента. Его срок хранения – 5 лет согласно ч.4 ст. 7 № 115-ФЗ. Ответы на вопросы бота в этом случае должны быть следующие: о происхождении – «запрошен у контрагента», о сфере деятельности – «Сбор данных о контрагенте», ведутся ли споры – «Споров нет /спор завершен», об отделе – «МКУ». Однако, пользователь может неверно указать сферу, к которой относится документ в данном случае и в качестве ответа выбрать «Обязательная отчетность», а на последующий вопрос о периодичности – дать ответ «годовой/за несколько лет». Но и в том и в другом случает ответ бота должен быть 5 лет (см. рисунок 5.2).

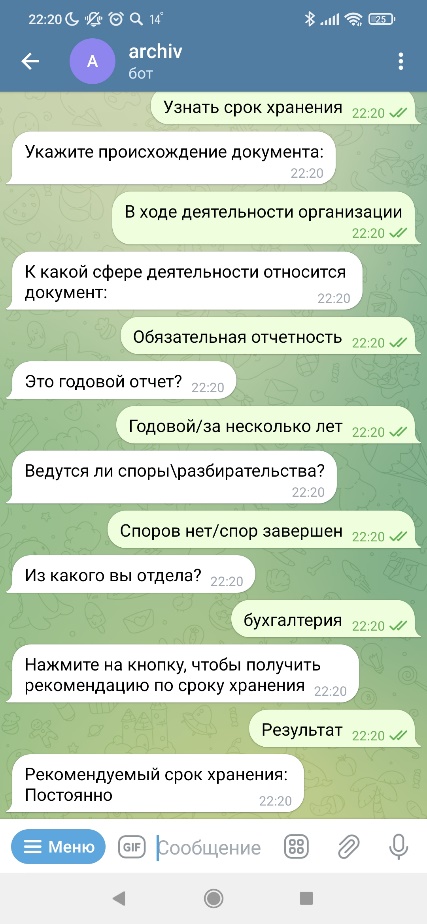


Рисунок 5.2 – Чаты с ботом по бухгалтерскому балансу

Из рисунка 5.2 понятно, что бот справляется с определением срока хранения годового бухгалтерского баланса, созданного в организации, баланса запрошенного у контрагента, и баланса запрошенного у контрагента, где сфера деятельности выбрана не совсем верно.

Третий класс – документы, имеющие срок хранения 75 лет. Согласно статье 435 Перечня, таким документом является трудовой договор, который хранится в отделе кадров. Для трудового договора не принципиально вредные условия труда, или нет, ведутся какие-либо споры или нет, срок хранения у него всегда будет 75 лет. Соответственно, ответ на вопрос бота о происхождении документа должен быть: «в ходе деятельности организации», о сфере деятельности – «Кадровые документы», о вредности – «вредность не определена», о спорах – «Споров нет /спор завершен», об отделе – «кадры». Ответ бота должен быть: «Рекомендуемый срок хранения: 75 лет» (см. рисунок 5.3).



Рисунок 5.3 – Чат с ботом по трудовому договору

Как следует из рисунка 5.3, ответ бота так же совпадает с ожидаемым.

Четвертый класс – документы, имеющие срок хранения до разрешения спора. К таким документам, согласно статье 96 Перечня, можно отнести договор лизинга, если по нему возникли споры и разногласия. Соответственно, ответ на вопрос бота о происхождении документа должен быть: «в ходе деятельности организации», о сфере деятельности – «ЛЗ», о спорах – «Ведется спор», об отделе – «МКУ». Ответ бота должен быть: «Рекомендуемый срок хранения: до разрешения спора» (см. рисунок 5.4).

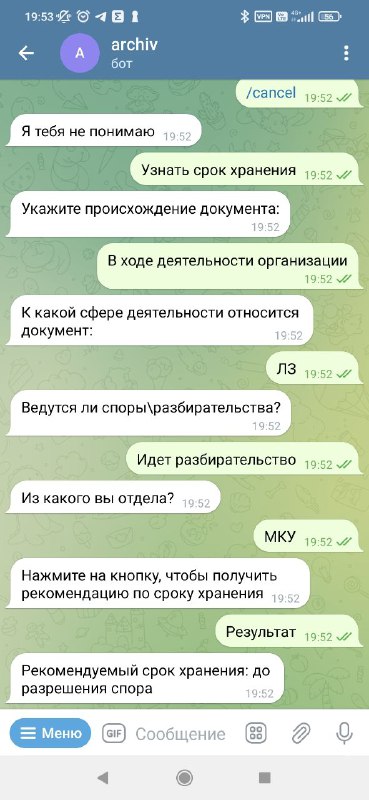


Рисунок 5.4 – Чат с ботом по спорному договору лизинга

Как видно из рисунка 5.4, в этом случае ответ бота так же совпадает с ожидаемым.

Таким образом, бот достаточно успешно выполняет свою функцию: подсказывает пользователю срок хранения документов, не заставляя обращаться к нормативно-правовым актам.

# ЗАКЛЮЧЕНИЕ

В настоящее время, несмотря на постепенный переход к электронному документообороту, количество документов на бумажных носителях продолжает расти. Несмотря на все существующие нормы и правила ведения документооборота, при большом количестве документов у пользователей часто возникают проблемы с их хранением. Существует множество нормативно-правовых документов, к которым приходится обращаться для определения срока хранения документа, что не всегда удобно. В связи с этим, целью данной работы являлось создание вспомогательного инструмента, подсказывающего срок хранения документа пользователю, у которого нет возможности обратиться к нормативным документам.

Для создания такого инструмента было решено собрать информацию о фонде организации, а именно лизинговой компании, определить самые часто встречающиеся виды документов, выделить их признаки и составить набор данных.

При создании набора данных не было получено разрешение использовать текст собираемых документов для тематического моделирования, однако, учитывая, что тема (сфера деятельности) документа является важным признаком для решения задачи классификации документов, возможное использование такого моделирования представляется достаточно перспективным. В связи с чем, для примера был использован метод латентного распределения Дирихле для определения тем ряда документов, найденных в открытом доступе.

На получившемся наборе данных решалась задача многоклассовой классификации несбалансированных данных. Для решения этой задачи использовался ряд простых методов: метод k-го ближайшего соседа, логистическая регрессия и деревья решений. Эти методы не дали желаемого результата в долю правильных ответов не менее 80 %. Поэтому был так же использован ансамблевый метод: градиентный бустинг. Результат работы этого метода, соответствовал желаемому.

В дальнейшем, с использованием библиотеки aiogram был написан простой чат-бот справочник, задающий пользователю ряд вопросов о документе и принимающий ответы в виде выбора определенных кнопок на созданной клавиатуре. В бот для принятия решения о сроке хранения документа, по заданным пользователем признакам, была загружена модель градиентного бустинга, обученная на собранных на первом этапе работе данных.

Результат работы бота можно считать удовлетворительным, он достаточно успешно подсказывает пользователю сколько должны храниться документы, не требуя при этом обращения к нормативно-правовым актам.

Однако этот инструмент-справочник не является универсальным и при дальнейшем использовании будет нуждаться в расширении. Сейчас модель обучена на самых распространенных документах, встречающихся в лизинговой компании, однако при появлении в компании новых отделов или при изменении законодательства, возникнет необходимость в расширении набора данных и добавлении в клавиатуры бота новых отделов и новых сфер, к которым относятся документы. Задачу добавления новых сфер можно будет решить с использованием тематического моделирования, которое использовалось в данной работе, однако все равно понадобиться настройка продукта под нужды конкретной организации в определенный момент времени.

Тем не менее, на настоящий момент, можно считать, что цель работы – разработка вспомогательного инструмента, подсказывающего рекомендуемый срок хранения документа, для сотрудников, не обладающими специальными знаниями и навыками в сфере документооборота, выполнена.

# Список использованных источников

1. О противодействии легализации (отмыванию) доходов, полученных преступным путем, и финансированию терроризма от 07.08.2001 № 115-ФЗ [Текст]. [федер.закон: принят Гос.Думой 13 июля 2001] (<https://base.garant.ru/12123862/>). Проверено 25.06.2022
2. Об утверждении правил организации хранения, комплектования, учета и использования документов Архивного фонда Российской Федерации и других архивных документов в органах государственной власти, органах местного самоуправления и организациях Приказ Минкультуры России от 31.03.2015 № 526 [приказ: зарегистрирован в Минюсте России 07 сентября 2015] (<http://www.consultant.ru/document/cons_doc_LAW_185738/>). Проверено 25.06.2022
3. Перечень типовых управленческих архивных документов, образующихся в процессе деятельности государственных органов, органов местного самоуправления и организаций, с указанием сроков хранения [перечень: утвержден Приказом Росархива № 236 20 декабря 2019 ] (<http://www.consultant.ru/document/cons_doc_LAW_345020/>). Проверено 25.06.2022
4. Айвазян, С.А. Прикладная статистика: Классификация и снижение размерности: Справ.изд. [Текст]/ С.А.Айвазян, В.М.Бухштабер, И.С.Енюков, Л.Д.Мешалкин; под ред. С.А.Айвазяна. – М.: Финансы и статистика, 1989. – 607 с.
5. Булатов, В.Г. Методы оценивания качества и многокритериальной оптимизации тематических моделей в библиотеке TopicNet.[Текст]: дис.... канд. тех. наук: 05.13.18 /Булатов Виктор Геннадьевич. – М., 2020. – 147 с.
6. Гласс, Дж. Статистические методы в педагогике и психологии [Текст] / Гласс Дж., Стэнли Дж. – М.: Прогресс, 1976. – 496 с.
7. Коршунов, А.В. Гомзин А.Г. [Тематическое моделирование текстов на естественном языке.](https://www.ispras.ru/proceedings/isp_23_2012/isp_23_2012_215/)[Текст] / А.В. Коршунов, А.Г. Гомзин // Труды Института системного программирования РАН. – 2012. – Том 23. – С. 215-244.
8. Краснянский, М.Н. Сравнительный анализ методов машинного обучения для решения задачи классификации документов научно-образовательного учреждения [Текст] / М. Н. Краснянский, А. Д. Обухов, Е. М. Соломатина, А. А. Воякина // Вестник ВГУ, СЕРИЯ: Системный анализ и информационные технологии. – 2018. – № 3. – С. 173-182
9. Наследов, А.Д. Математические методы психологического исследования. Анализ и интерпретация данных. Учебное пособие [Текст]. – СПб: Речь, 2004. – 392 с.
10. Неделько, В.М. Регрессионные модели в задаче классификации [Текст] / В.М. Неделько // Сибирский журнал индустриальной математики. – 2014. № 1. – том 17. – С. 86–98
11. López, Victoria An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics [Текст]/ Victoria López, Alberto Fernández, Salvador García, Vasile Palade, Francisco Herrera // Information Sciences –2013. –Volume 250. – С. 113-141 (<https://www.sciencedirect.com/science/article/pii/S0020025513005124>) Проверено 25.06.2022

Приложение 1

Построение тематической модели

1. Установка библиотек

!pip install beautifulsoup4

from bs4 import BeautifulSoup

import requests

import pandas as pd

!pip install python-docx

!pip install pdf2docx==0.5.3

!pip uninstall PyMuPDF==1.20.0

!pip install PyMuPDF==1.19.6

from pdf2docx import parse

import docx

import os

!pip install pyLDAvis

!pip install https://github.com/explosion/spacy-models/releases/download/ru\_core\_news\_sm-3.1.0/ru\_core\_news\_sm-3.1.0.tar.gz

import nltk; nltk.download('stopwords')

import re

import numpy as np

import pandas as pd

from pprint import pprint

from nltk.corpus import stopwords

import gensim

import gensim.corpora as corpora

from gensim.utils import simple\_preprocess

from gensim.models import CoherenceModel

import spacy

import pyLDAvis

import pyLDAvis.gensim\_models as gensimvis

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings("ignore",category=DeprecationWarning)

1. Загрузка документов с сайта:

url\_list = ['https://unecon.ru/doc/arhiv-dokumentov',

            'https://unecon.ru/doc/ustav',

            'https://unecon.ru/document1',

            'https://unecon.ru/page/programma-strategicheskogo-razvitiya-spbgeu-na-period-2018-2025-gg',

            'https://unecon.ru/doc/organizacionnye-dokumenty/deyatelnost-universiteta',

            'https://unecon.ru/doc/organizacionnye-dokumenty/organizaciya-obr-deyatelnosti',

            'https://unecon.ru/doc/rabotnikam',

            'https://unecon.ru/doc/obuchayushchimsya',

            'https://unecon.ru/corruption',

            'https://unecon.ru/distant/normativnye-akty']

alldocs = []

for url in url\_list:

  page = requests.get(url)

  soup = BeautifulSoup(page.text, "html.parser")

  alldocs.extend(soup.findAll(attrs={'target':"\_blank", 'title':''}))

new\_alldocs = []

for i in alldocs:

  new\_alldocs.extend(i)

allpdfdoc = []

for doc in alldocs:

  html = str(doc)

  start = html.index('"')

  fin = html[start+1:].index('"')

  allpdfdoc.append(html[start+1:fin+start+1])

allpdfdoc = [url for url in allpdfdoc if '.pdf' in str(url)]

with open("list.txt", "w") as file:

    print(\*allpdfdoc, file=file, sep = '\n')

!wget -i /content/list.txt

1. Преобразование из формата .pdf в формат .txt

def getText(filename):

    doc = docx.Document(filename)

    fullText = []

    for para in doc.paragraphs:

        fullText.append(para.text)

    return '\n'.join(fullText)

path = "/content/"

dirs = os.listdir(path)

for file in dirs:

  if '.pdf' in file:

    path\_file = path+file

    new\_docx\_file = file.replace('.pdf','.docx')

    parse(path\_file, new\_docx\_file)

    text=getText(new\_docx\_file).replace('\n','')

    new\_text\_file=file.replace('.pdf','.txt')

    new\_path="/content/d/"

    with open(new\_path+new\_text\_file, 'w', encoding='utf-8') as file:

        file.write(text)

1. Создание датасета из названий документов и их текстового содержания

file\_list = [files for files in os.listdir('/content/d')]

path = '/content/d/'

lines=[]

for file in file\_list:

    f = open(path+file, "r")

    lines.append(f.read())

    f.close()

df = pd.DataFrame({'name':file\_list, 'text':lines})

df = df.query("text!=''")

1. Преобразование документа в список токенов без пунктуации

data = df.text.values.tolist()

data = [re.sub(r'\s+', ' ', sent) for sent in data]

data = [re.sub("\'", "", sent) for sent in data]

pprint(data[:1])

def sent\_to\_words(sentences):

    for sentence in sentences:

        yield(gensim.utils.simple\_preprocess(str(sentence), deacc=True))

data\_words = list(sent\_to\_words(data))

1. Выделение устойчивых оборотов, отбрасывание запрещенных слов, лемматизация (приведение каждого слова к его нормальной форме)

bigram = gensim.models.Phrases(data\_words, min\_count=5, threshold=100) # higher threshold fewer phrases.

trigram = gensim.models.Phrases(bigram[data\_words], threshold=100)

bigram\_mod = gensim.models.phrases.Phraser(bigram)

trigram\_mod = gensim.models.phrases.Phraser(trigram)

stop\_words = stopwords.words("russian")

def remove\_stopwords(texts):

    return [[word for word in simple\_preprocess(str(doc)) if word not in stop\_words] for doc in texts]

def make\_bigrams(texts):

    return [bigram\_mod[doc] for doc in texts]

def make\_trigrams(texts):

    return [trigram\_mod[bigram\_mod[doc]] for doc in texts]

def lemmatization(texts, allowed\_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):

    texts\_out = []

    for sent in texts:

        doc = nlp(" ".join(sent))

        texts\_out.append([token.lemma\_ for token in doc if token.pos\_ in allowed\_postags])

    return texts\_out

data\_words\_nostops = remove\_stopwords(data\_words)

data\_words\_bigrams = make\_bigrams(data\_words\_nostops)

nlp = spacy.load("ru\_core\_news\_sm", disable=['parser', 'ner'])

data\_lemmatized = lemmatization(data\_words\_bigrams, allowed\_postags=['NOUN', 'ADJ', 'VERB', 'ADV'])

id2word = corpora.Dictionary(data\_lemmatized)

texts = data\_lemmatized

corpus = [id2word.doc2bow(text) for text in texts]

1. Выбор лучшей модели (Когерентность)

def compute\_coherence\_values(dictionary, corpus, texts, limit, start=1, step=1):

coherence\_values = []

model\_list = []

for num\_topics in range(start, limit, step):

model=gensim.models.ldamodel.LdaModel(corpus=corpus,

id2word=id2word,

num\_topics=num\_topics,

random\_state=100,

update\_every=1,

chunksize=100,

passes=10,

alpha='auto',

per\_word\_topics=True)

model\_list.append(model)

coherencemodel = CoherenceModel(model=model, texts=texts, dictionary=dictionary)

coherence\_values.append(coherencemodel.get\_coherence())

return model\_list, coherence\_values

model\_list, coherence\_values = compute\_coherence\_values(dictionary=id2word, corpus=corpus, texts=data\_lemmatized, start=1, limit=10, step=1)

limit=10; start=1; step=1;

x = range(start, limit, step)

plt.plot(x, coherence\_values)

plt.xlabel("Num Topics")

plt.ylabel("Coherence score")

plt.legend(("coherence\_values"), loc='best')

plt.show()

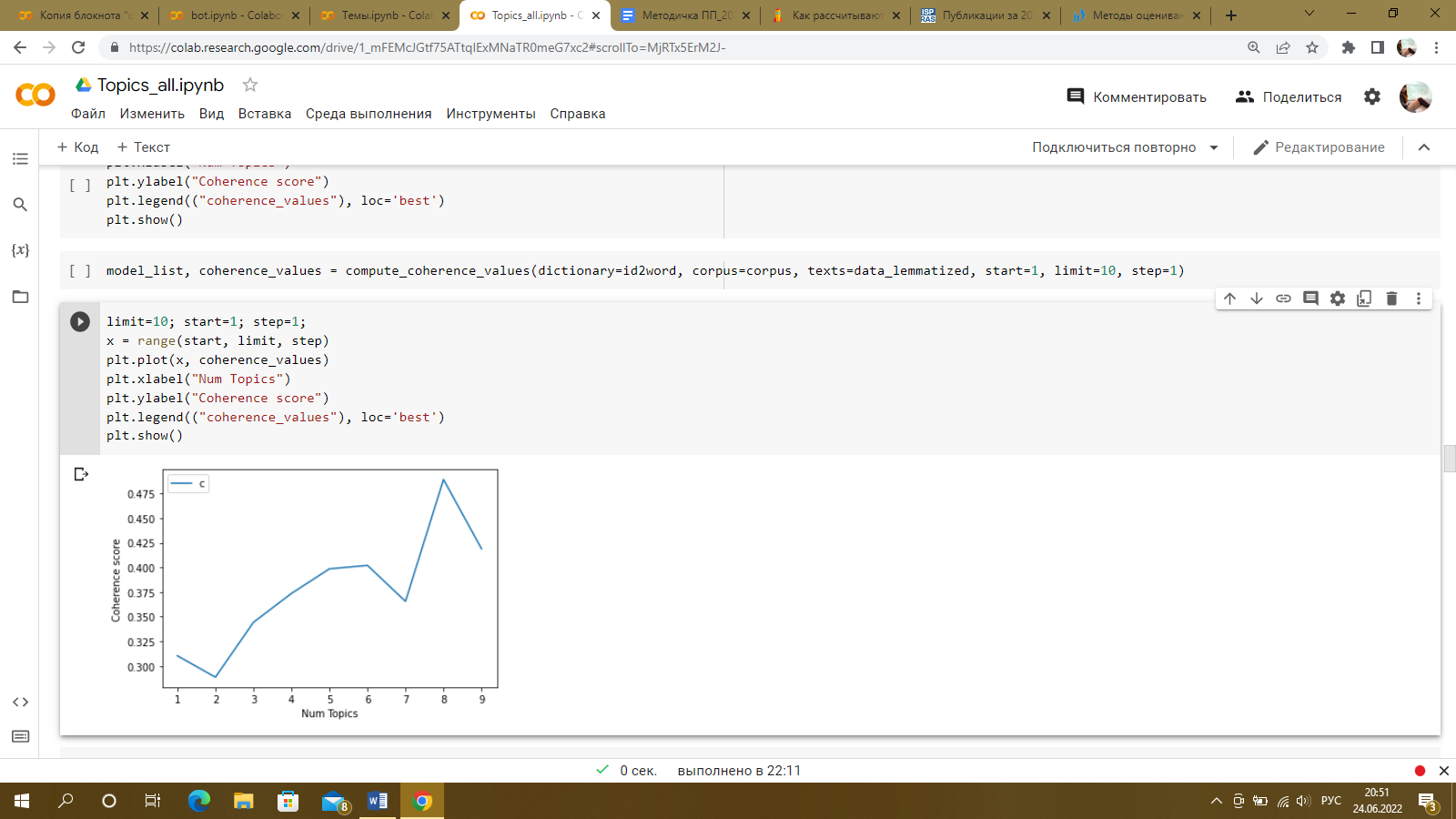


Рисунок 1.1 Показатель когерентности в зависимости от числа тем

for m, cv in zip(x, coherence\_values):

    print("Num Topics =", m, " has Coherence Value of", round(cv, 4))

Num Topics = 1 has Coherence Value of 0.3104

Num Topics = 2 has Coherence Value of 0.2887

Num Topics = 3 has Coherence Value of 0.3443

Num Topics = 4 has Coherence Value of 0.3736

Num Topics = 5 has Coherence Value of 0.3986

Num Topics = 6 has Coherence Value of 0.402

Num Topics = 7 has Coherence Value of 0.3656

Num Topics = 8 has Coherence Value of 0.4892

Num Topics = 9 has Coherence Value of 0.4189

1. Создание модели и визуализация:

lda\_model = gensim.models.ldamodel.LdaModel(corpus=corpus,

                                           id2word=id2word,

                                           num\_topics=8,

                                           random\_state=100,

                                           update\_every=1,

                                           chunksize=100,

                                           passes=10,

                                           alpha='auto',

                                           per\_word\_topics=True)

pyLDAvis.enable\_notebook()

vis = gensimvis.prepare(lda\_model, corpus, id2word)

vis

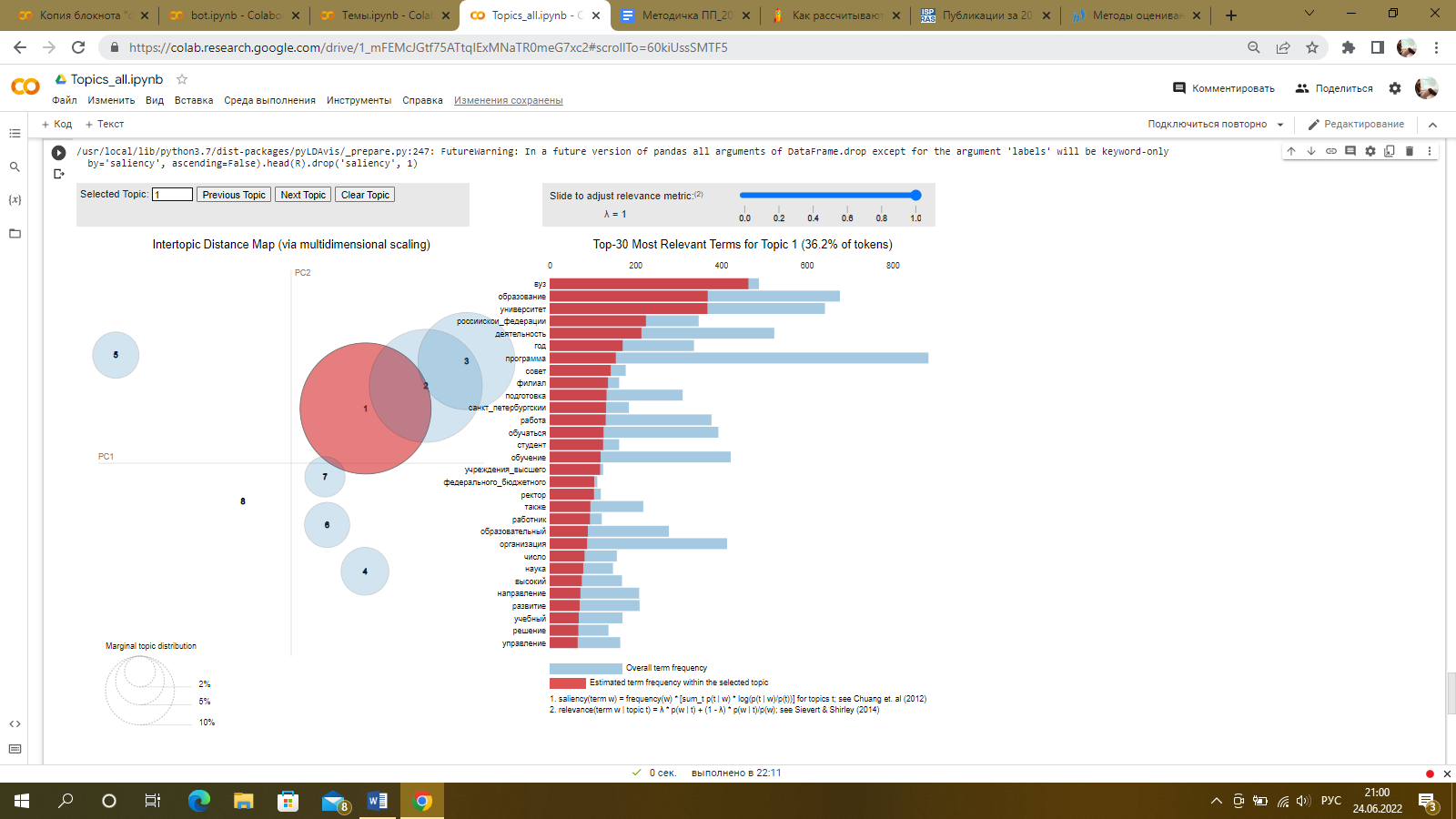


Рисунок 1.2 Визуализация результата работы модели

Приложение 2

Построение модели, классифицирующей документы по срокам хранения

1. Импорт библиотек

import pandas as pd

import os

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

from google.colab import drive

from sklearn.preprocessing import LabelEncoder

from imblearn.under\_sampling import NearMiss

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

from sklearn.model\_selection import KFold

from sklearn.model\_selection import cross\_val\_score

from pickle import dump

from pickle import load

1. Разведочный анализ и кодирование

drive.mount('/content/gdrive')

os.chdir('/content/gdrive/MyDrive/datasets/')

df = pd.read\_excel('docs\_2.xlsx')

df = df.apply(lambda x: x.str.strip())

df.info()

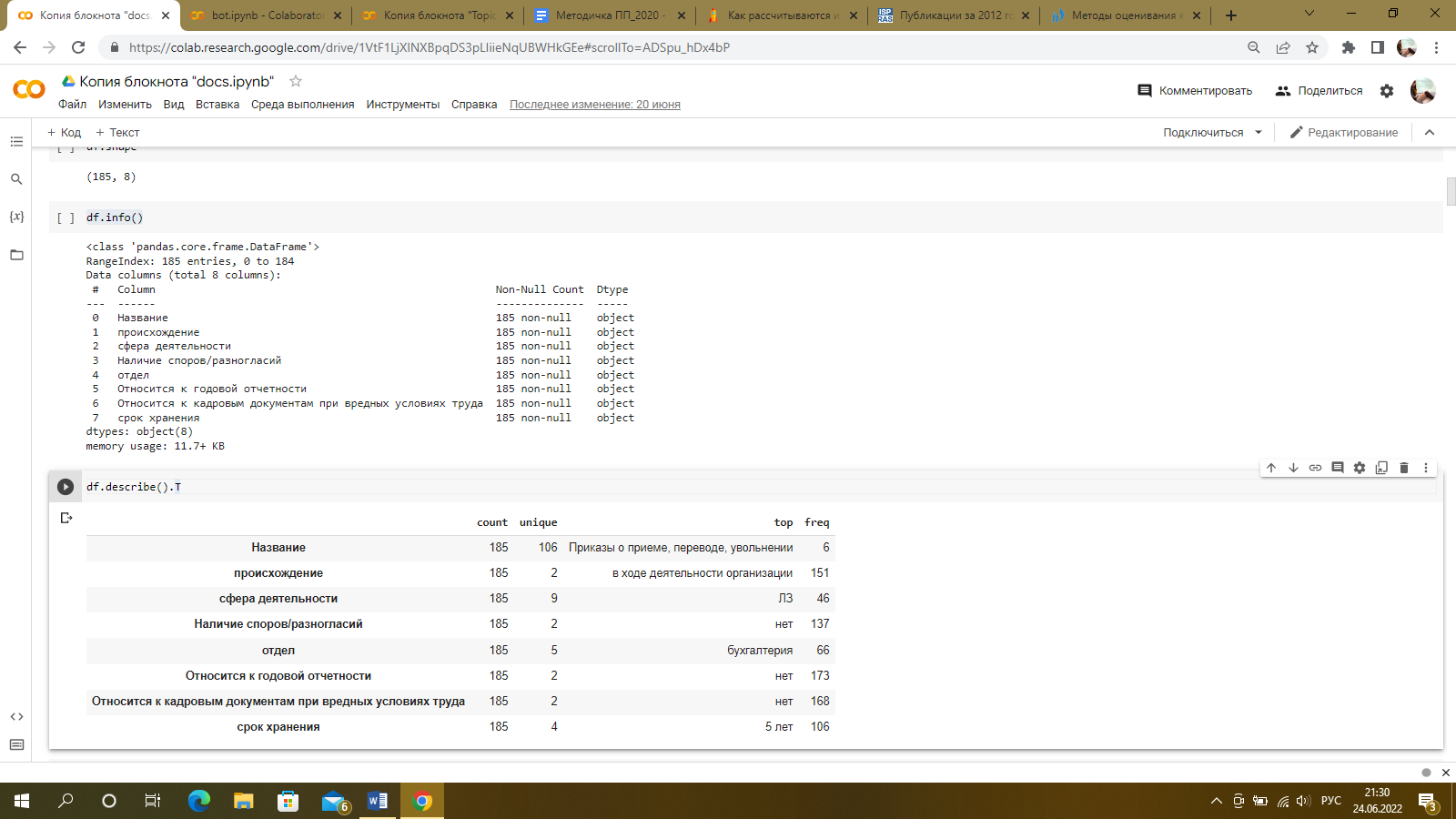


Рисунок 2.1 Информация о столбцах и количестве строк

df.describe().T

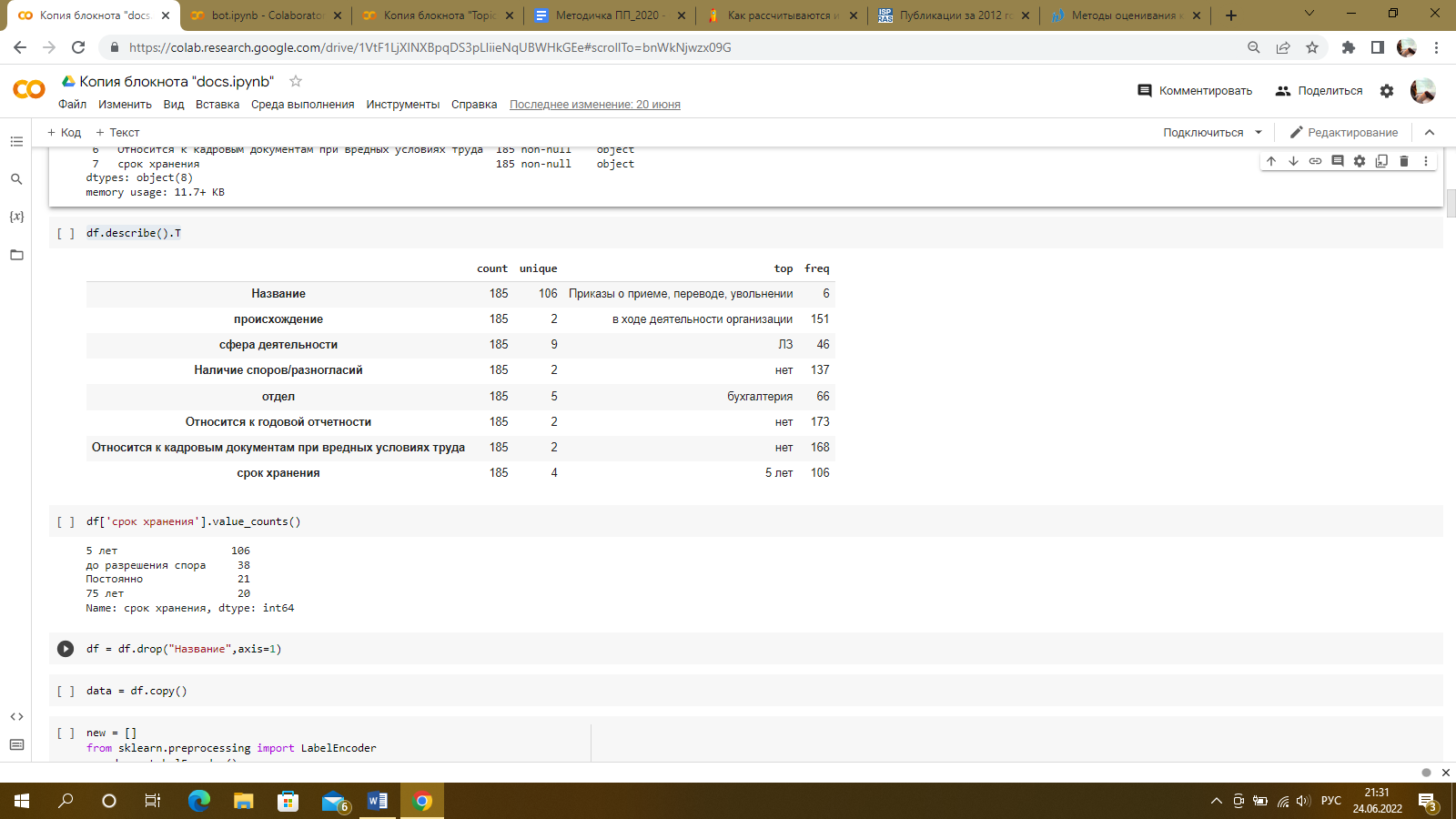


Рисунок 2.2 Информация о количестве значений

df['срок хранения'].value\_counts()

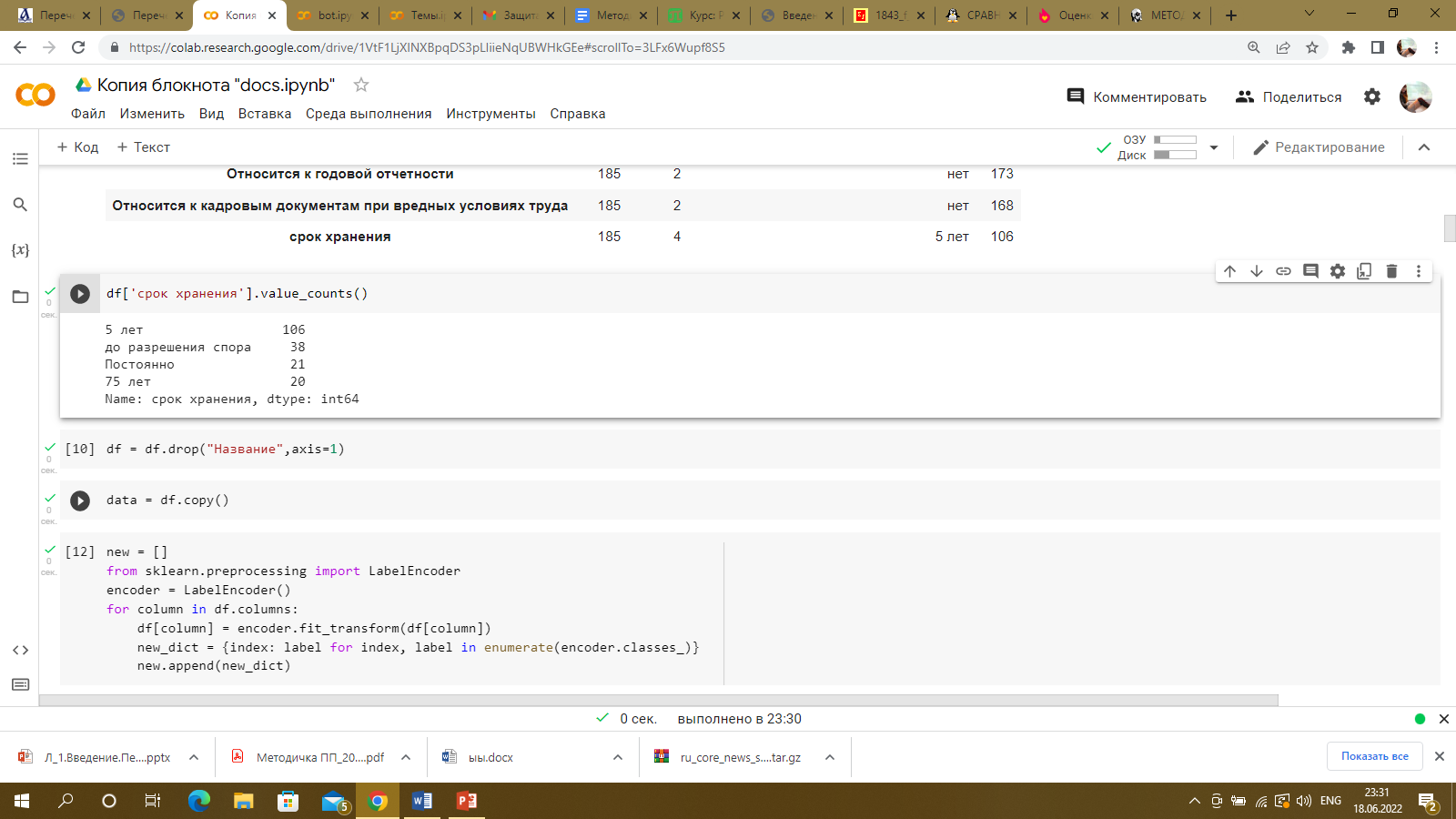


Рисунок 2.3 Количество документов по срокам хранения

df = df.drop("Название",axis=1)

data = df.copy()

new = []

encoder = LabelEncoder()

for column in df.columns:

df[column] = encoder.fit\_transform(df[column])

new\_dict = {index: label for index, label in enumerate(encoder.classes\_)}

new.append(new\_dict)

1. Сравнение популярных методов и балансирование выборки

array = df.values

Y = array[:,6]

X = array[:,:6]

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.3, random\_state = 7, stratify = Y)

print('Кол-во меток со значением 5 лет: {}'.format(sum(Y\_train == 0)))

print('Кол-во меток со значением 75 лет: {}'.format(sum(Y\_train == 1)))

print('Кол-во меток со значением Постоянно: {}'.format(sum(Y\_train == 2)))

print('Кол-во меток со значением До разрешения спора: {}'.format(sum(Y\_train == 3)))

Кол-во меток со значением 5 лет: 74

Кол-во меток со значением 75 лет: 14

Кол-во меток со значением Постоянно: 15

Кол-во меток со значением До разрешения спора: 26

nm = NearMiss()

X\_train\_miss, Y\_train\_miss = nm.fit\_resample(X\_train, Y\_train.ravel())

X\_test\_miss, Y\_test\_miss = nm.fit\_resample(X\_test, Y\_test.ravel())

print('После применения метода кол-во меток со значением 5 лет: {}'.format(sum(Y\_train\_miss == 0)))

print('После применения метода кол-во меток со значением 75 лет: {}'.format(sum(Y\_train\_miss == 1)))

print('После применения метода кол-во меток со значением Постоянно: {}'.format(sum(Y\_train\_miss == 2)))

print('После применения метода кол-во меток со значением До разрешения спора: {}'.format(sum(Y\_train\_miss == 3)))

После применения метода кол-во меток со значением 5 лет: 14

После применения метода кол-во меток со значением 75 лет: 14

После применения метода кол-во меток со значением Постоянно: 14

После применения метода кол-во меток со значением До разрешения спора: 14

models = [('LR', LogisticRegression(max\_iter=1000)), ('SVM', SVC()), ('LDA', LinearDiscriminantAnalysis()), ('KNN', KNeighborsClassifier()), ('NB', GaussianNB()), ('CART', DecisionTreeClassifier())]

results = []

names = []

for name, model in models:

  kfold = KFold(n\_splits=10, random\_state=7, shuffle=True)

  cv\_results = cross\_val\_score(model, X, Y, cv=kfold, scoring='accuracy')

  results.append(cv\_results)

  names.append(name)

fig = plt.figure()

fig.suptitle('Сравнение моделей')

ax = fig.add\_subplot()

plt.boxplot(results)

ax.set\_xticklabels(names)

plt.show()

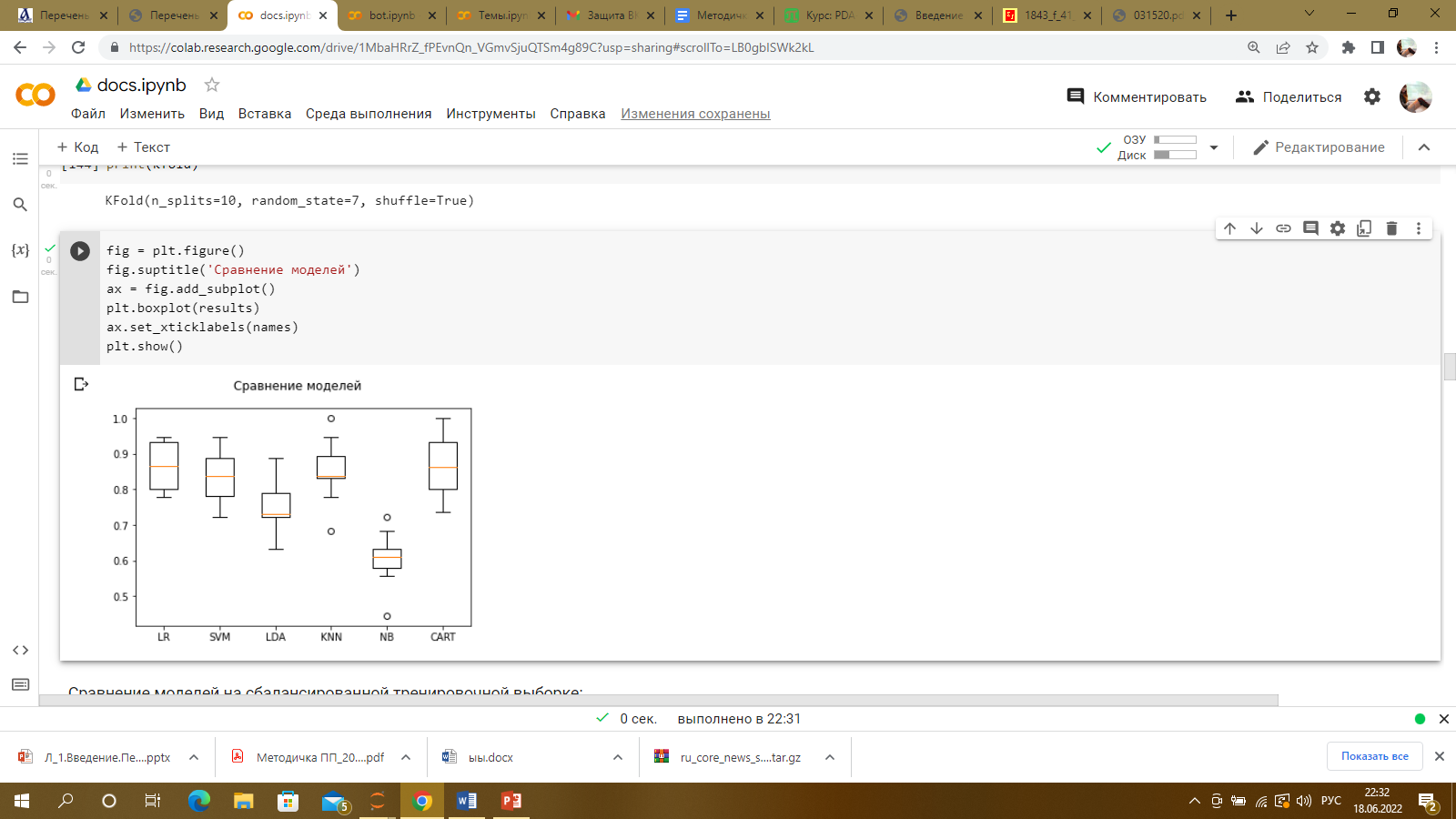


Рисунок 2.4 – Сравнение моделей

1. Логистическая регрессия

model = LogisticRegression(max\_iter=1000)

model.fit(X\_train\_miss,Y\_train\_miss)

predictions = model.predict(X\_test)

accuracy\_score(Y\_test, predictions)

1. Метод ближайших соседей

model=KNeighborsClassifier()

model.fit(X\_train\_miss,Y\_train\_miss)

predictions = model.predict(X\_test)

accuracy\_score(Y\_test, predictions)

print(classification\_report(Y\_test, predictions))

1. Решающие деревья

model = DecisionTreeClassifier()

model.fit(X\_train\_miss, Y\_train\_miss)

predictions = model.predict(X\_test)

accuracy\_score(Y\_test, predictions)

print(classification\_report(Y\_test, predictions))

1. Метод Gradient Boosting Machine (GBM)

model = GradientBoostingClassifier(n\_estimators = 100, random\_state = 7)

results = cross\_val\_score(model, X\_train, Y\_train, cv=kfold)

results.mean()

results.std()

model.fit(X\_train, Y\_train)

predictions = model.predict(X\_test)

accuracy\_score(Y\_test, predictions)

1. Сохранение модели для использования в чат-боте

model = GradientBoostingClassifier(n\_estimators = 100, random\_state = 7)

model.fit(X\_train, Y\_train)

filename = 'finalized\_model.sav'

dump(model, open(filename, 'wb'))

Приложение 3

Создание чат-бота

1. Импорт библиотек

!pip install aiogram

from pickle import load

import os

import numpy as np

from google.colab import drive

from aiogram.types import ReplyKeyboardRemove, ReplyKeyboardMarkup, KeyboardButton

from aiogram import Bot, types

from aiogram.dispatcher import Dispatcher

from aiogram.utils import executor

!pip install nest\_asyncio

import nest\_asyncio

nest\_asyncio.apply()

1. Словарь на основе данных разведочного анализа

origin\_dict = {0: 'В ходе деятельности организации', 1: 'Запрошен у контрагента'}

field\_dict = {0: 'Административно-хозяйственные расходы (АХР)',

              1: 'Кадровые документы',

              2: 'ЛЗ',

              3: 'Обязательная отчетность',

              4: 'Организационно - распорядительные документы',

              5: 'Работа с кредиторами и заемщиками',

              6: 'Реализация имущества',

              7: 'Сбор данных о контрагенте',

              8: 'Судебные документы'}

danger\_dict = {0: 'Да, вредность есть', 1: 'Нет, вредность не определена'}

trial\_dict = {0: 'Идет разбирательство', 1: 'Споров нет/спор завершен'}

report\_dict = {0: 'Годовой/за несколько лет', 1: 'Нет'}

department\_dict = {0: 'АХО', 1: 'МКУ', 2: 'ЮО', 3: 'бухгалтерия', 4: 'кадры'}

1. Создание кнопок

button\_main = KeyboardButton('Начать')

button\_back = KeyboardButton('Назад')

button\_term = KeyboardButton('Узнать срок хранения')

button\_result = KeyboardButton('Результат')

origin\_button\_0 = KeyboardButton(list(origin\_dict.values())[0])

origin\_button\_1 = KeyboardButton(list(origin\_dict.values())[1])

field\_button\_0 = KeyboardButton(list(field\_dict.values())[0])

field\_button\_1 = KeyboardButton(list(field\_dict.values())[1])

field\_button\_2 = KeyboardButton(list(field\_dict.values())[2])

field\_button\_3 = KeyboardButton(list(field\_dict.values())[3])

field\_button\_4 = KeyboardButton(list(field\_dict.values())[4])

field\_button\_5 = KeyboardButton(list(field\_dict.values())[5])

field\_button\_6 = KeyboardButton(list(field\_dict.values())[6])

field\_button\_7 = KeyboardButton(list(field\_dict.values())[7])

field\_button\_8 = KeyboardButton(list(field\_dict.values())[8])

danger\_button\_0 = KeyboardButton(list(danger\_dict.values())[0])

danger\_button\_1 = KeyboardButton(list(danger\_dict.values())[1])

trial\_button\_0 = KeyboardButton(list(trial\_dict.values())[0])

trial\_button\_1 = KeyboardButton(list(trial\_dict.values())[1])

report\_button\_0 = KeyboardButton(list(report\_dict.values())[0])

report\_button\_1 = KeyboardButton(list(report\_dict.values())[1])

department\_button\_0 = KeyboardButton(list(department\_dict.values())[0])

department\_button\_1 = KeyboardButton(list(department\_dict.values())[1])

department\_button\_2 = KeyboardButton(list(department\_dict.values())[2])

department\_button\_3 = KeyboardButton(list(department\_dict.values())[3])

department\_button\_4 = KeyboardButton(list(department\_dict.values())[4])

1. Создание клавиатур

main\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).add(button\_main)

start\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).add(button\_term).add(button\_back)

back\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).add(button\_back)

result\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).add(button\_result).add(button\_back)

origin\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).insert(origin\_button\_0).insert(origin\_button\_1).insert(button\_back)

field\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).insert(field\_button\_0)\

          .insert(field\_button\_1)\

          .insert(field\_button\_2)\

          .insert(field\_button\_3)\

          .insert(field\_button\_4)\

          .insert(field\_button\_5)\

          .insert(field\_button\_6)\

          .insert(field\_button\_7)\

.insert(field\_button\_8)\

          .insert(button\_back)

danger\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).insert(danger\_button\_0).insert(danger\_button\_1).insert(button\_back)

trial\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).insert(trial\_button\_0).insert(trial\_button\_1).insert(button\_back)

report\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).insert(report\_button\_0).insert(report\_button\_1).insert(button\_back)

department\_kb = ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True).insert(department\_button\_0)\

          .insert(department\_button\_1)\

          .insert(department\_button\_2)\

          .insert(department\_button\_3)\

          .insert(department\_button\_4)\

          .insert(button\_back)

1. Загрузка модели и функция для ее применения

drive.mount('/content/gdrive')

os.chdir('/content/gdrive/MyDrive/datasets/')

result\_dict = {0: '5 лет', 1: '75 лет', 2: 'Постоянно', 3: 'до разрешения спора'}

loaded\_model = load(open('finalized\_model.sav', 'rb'))

def result(user\_dict):

  lst = []

  lst.append(user\_dict['origin'])

  lst.append(user\_dict['field'])

  lst.append(user\_dict['trial'])

  lst.append(user\_dict['department'])

  lst.append(user\_dict['report'])

  lst.append(user\_dict['danger'])

  lst = np.array(lst)

  lst = lst.reshape(1, -1)

  res = loaded\_model.predict(lst)

  return result\_dict[int(res)]

1. Создание бота

Token = '5417278260:AAHXoCB0XweQiUgj-tfyk63k6oI5cYmMm2M'

bot = Bot(token = Token)

dp = Dispatcher(bot

data = {}

#Команда start

@dp.message\_handler(commands=['start'])

async def start (message: types.Message):

  await bot.send\_message(message.from\_user.id, 'Я помогу вам определить срок хранения документа', reply\_markup = main\_kb)

#Обработка ответов

@dp.message\_handler()

async def text (message: types.Message):

  answer=message.text

  #кнопки без получения данных:

  if answer == 'Начать':

   await bot.send\_message(message.from\_user.id, 'Нужно ответить на несколько вопросов.',reply\_markup = start\_kb)

  elif answer == 'Назад':

   await bot.send\_message(message.from\_user.id, 'Нужно ответить на несколько вопросов',reply\_markup = start\_kb)

  elif answer == 'Узнать срок хранения':

    await bot.send\_message(message.from\_user.id, 'Укажите происхождение документа:',reply\_markup = origin\_kb)

  #кнопки, данные из которых нужно записать:

  #Происхождение:

  elif answer == list(origin\_dict.values())[0]:

    data.update({'origin':0})

    await bot.send\_message(message.from\_user.id, 'К какой сфере деятельности относится документ:',reply\_markup = field\_kb)

  elif answer == list(origin\_dict.values())[1]:

    data.update({'origin':1})

    await bot.send\_message(message.from\_user.id, 'К какой сфере деятельности относится документ:',reply\_markup = field\_kb)

  #Сфера деятельности:

  elif answer == list(field\_dict.values())[0]:

    data.update({'field':0})

    data.update({'danger':1})

    data.update({'report':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  elif answer == list(field\_dict.values())[1]:

    data.update({'field':1})

    data.update({'report':1})

    #вопрос о вредности принципиален только для кадровых документов, остальным видам документов можно сразу писать "нет"

    await bot.send\_message(message.from\_user.id, 'Отнесены ли условия труда работников, указанных в этом документе, к вредным или опасным?',reply\_markup = danger\_kb)

  elif answer == list(field\_dict.values())[2]:

    data.update({'field':2})

    data.update({'danger':1})

    data.update({'report':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  elif answer == list(field\_dict.values())[3]:

    data.update({'field':3})

    data.update({'danger':1})

    #вопрос о периодичности отчетности актуален только для вида документа "обязательная отчетность", остальным можно сразу писать - "нет"

    await bot.send\_message(message.from\_user.id, 'Это годовой отчет?',reply\_markup = report\_kb)

  elif answer == list(field\_dict.values())[4]:

    data.update({'field':4})

    data.update({'danger':1})

    data.update({'report':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  elif answer == list(field\_dict.values())[5]:

    data.update({'field':5})

    data.update({'danger':1})

    data.update({'report':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  elif answer == list(field\_dict.values())[6]:

    data.update({'field':6})

    data.update({'danger':1})

    data.update({'report':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  elif answer == list(field\_dict.values())[7]:

    data.update({'field':7})

    data.update({'danger':1})

    data.update({'report':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  elif answer == list(field\_dict.values())[8]:

    data.update({'field':8})

    data.update({'danger':1})

    data.update({'report':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  #Вредность

  elif answer == list(danger\_dict.values())[0]:

    data.update({'danger':0})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  elif answer == list(danger\_dict.values())[1]:

    data.update({'danger':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  #Отчетность

  elif answer == list(report\_dict.values())[0]:

    data.update({'report':0})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  elif answer == list(report\_dict.values())[1]:

    data.update({'report':1})

    await bot.send\_message(message.from\_user.id, 'Ведутся ли споры\разбирательства?',reply\_markup = trial\_kb)

  #Споры

  elif answer == list(trial\_dict.values())[0]:

    data.update({'trial':0})

    await bot.send\_message(message.from\_user.id, 'Из какого вы отдела?',reply\_markup = department\_kb)

  elif answer == list(trial\_dict.values())[1]:

    data.update({'trial':1})

    await bot.send\_message(message.from\_user.id, 'Из какого вы отдела?',reply\_markup = department\_kb)

  #Отдел

  elif answer == list(department\_dict.values())[0]:

    data.update({'department':0})

    await bot.send\_message(message.from\_user.id, 'Нажмите на кнопку, чтобы получить рекомендацию по сроку хранения',reply\_markup = result\_kb)

  elif answer == list(department\_dict.values())[1]:

    data.update({'department':1})

    await bot.send\_message(message.from\_user.id, 'Нажмите на кнопку, чтобы получить рекомендацию по сроку хранения',reply\_markup = result\_kb)

  elif answer == list(department\_dict.values())[2]:

    data.update({'department':2})

    await bot.send\_message(message.from\_user.id, 'Нажмите на кнопку, чтобы получить рекомендацию по сроку хранения',reply\_markup = result\_kb)

  elif answer == list(department\_dict.values())[3]:

    data.update({'department':3})

    await bot.send\_message(message.from\_user.id, 'Нажмите на кнопку, чтобы получить рекомендацию по сроку хранения',reply\_markup = result\_kb)

  elif answer == list(department\_dict.values())[4]:

    data.update({'department':4})

    await bot.send\_message(message.from\_user.id, 'Нажмите на кнопку, чтобы получить рекомендацию по сроку хранения',reply\_markup = result\_kb)

  #Результат

  elif answer == 'Результат':

    await bot.send\_message(message.from\_user.id, f'Рекомендуемый срок хранения: {result(data)}', reply\_markup = start\_kb)

  #На случай непредвиденных ситуаций

  else:

    await bot.send\_message(message.from\_user.id, 'Я тебя не понимаю',reply\_markup= start\_kb)

if \_\_name\_\_ == '\_\_main\_\_':

  executor.start\_polling(dp)

1. П.4.18 Приказ Минкультуры России от 31.03.2015 № 526 Об утверждении правил организации хранения, комплектования, учета и использования документов Архивного фонда Российской Федерации и других архивных документов в органах государственной власти, органах местного самоуправления и организациях [↑](#footnote-ref-1)
2. Коршунов, А.В. Гомзин А.Г. Тематическое моделирование текстов на естественном языке. / А.В. Коршунов, А.Г. Гомзин // Труды Института системного программирования РАН. – 2012. – Том 23. – С. 215-244. [↑](#footnote-ref-2)
3. 5. Булатов, В.Г. Методы оценивания качества и многокритериальной оптимизации тематических моделей в библиотеке TopicNet: дис. канд. тех. наук: 05.13.18 /Булатов Виктор Геннадьевич. – М., 2020. – 147 с. [↑](#footnote-ref-3)
4. Наследов, А.Д. Математические методы психологического исследования. Анализ и интерпретация данных. Учебное пособие [Текст]. – СПб: Речь, 2004. – 392 с. [↑](#footnote-ref-4)
5. Гласс, Дж. Статистические методы в педагогике и психологии / Гласс Дж., Стэнли Дж. – М.: Прогресс, 1976. – 496 с. [↑](#footnote-ref-5)
6. Айвазян, С.А. Прикладная статистика: Классификация и снижение размерности: Справ.изд. / С.А.Айвазян, В.М.Бухштабер, И.С.Енюков, Л.Д.Мешалкин; под ред. С.А.Айвазяна. – М.: Финансы и статистика, 1989. – 607 с. [↑](#footnote-ref-6)
7. Краснянский, М.Н. Сравнительный анализ методов машинного обучения для решения задачи классификации документов научно-образовательного учреждения / М. Н. Краснянский, А. Д. Обухов, Е. М. Соломатина, А. А. Воякина // Вестник ВГУ, СЕРИЯ: Системный анализ и информационные технологии. – 2018. – № 3. – С. 173-182 [↑](#footnote-ref-7)
8. Неделько, В.М. Регрессионные модели в задаче классификации / В.М. Неделько // Сибирский журнал индустриальной математики. – 2014. № 1. – том 17. – С. 86–98 [↑](#footnote-ref-8)
9. [↑](#footnote-ref-9)
10. López, Victoria An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics [Текст]/ Victoria López, Alberto Fernández, Salvador García, Vasile Palade, Francisco Herrera // Information Sciences –2013. –Volume 250. – С. 113-141 (<https://www.sciencedirect.com/science/article/pii/S0020025513005124>) Проверено 25.06.2022 [↑](#footnote-ref-10)