TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа ядерных технологий
Направление подготовки 01.03.02. «Прикладная математика и информатика»
Отделение школы (НОЦ) Отделение экспериментальной физики

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка программного обеспечения для составления программы научных конференций

УДК 004.415.2:001.83(063)

Студент

Группа	ФИО	Подпись	Дата
0В8Б	Петров Дмитрий Владимирович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ ИЯТШ	Семенов Михаил Евгеньевич	Кандидат ф-м. наук, доцент		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Концепция стартап-проекта»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ШИП	Калашникова Татьяна Владимировна	Кандидат техн. наук, доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Сечин Андрей Александрович	Кандидат техн. наук, доцент		

ДОПУСТИТЬ К ЗАЩИТЕ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ООП 01.03.02 «Прикладная математика и информатика»	Крицкий Олег Леонидович	Кандидат ф-м. наук, доцент		

Томск – 2022 г.

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение

высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Инженерная школа ядерных технологий
Направление подготовки 01.03.02 «Прикладная математика и информатика»
Отделение экспериментальной физики

УТВЕРЖДАЮ:

Руководитель ООП

_____ _____ Крицкий О.Л.

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
0В8Б	Петрову Дмитрию Владимировичу

Тема работы:

Разработка программного обеспечения для составления программы научных конференций	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Программы конференции “Перспективы развития фундаментальных наук” за 2016-2021 годы.

Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Выбор и программная реализация методов классификации 2. Сбор и подготовка данных для проведения машинного обучения 3. Верификация и тестирование работоспособности *4.* Разработка графического интерфейса
Перечень графического материала	1. Архитектура нейронной сети 2. График зависимости ошибки от количества эпох обучения 3. Матрица неточностей 4. Изображения примеров работы программы *5.* Диаграмма распределение данных по классам

Консультанты по разделам выпускной квалификационной работы

(если необходимо, с указанием разделов)

Раздел	Консультант
Концепция стартап-проекта	Калашникова Татьяна Владимировна
Социальная ответственность	Сечин Андрей Александрович

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ	Семенов Михаил Евгеньевич	к. ф.-м. н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0В8Б	Петров Дмитрий Владимирович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«КОНЦЕПЦИЯ СТАРТАП-ПРОЕКТА»

Студенту:

Группа	ФИО
0В8Б	Петрову Дмитрию Владимировичу

Школа	ИЯТШ	Направление	01.03.02 Прикладная математика и информатика
Уровень образования	Бакалавриат		

Перечень вопросов, подлежащих разработке:	
Проблема конечного потребителя, которую решает продукт, который создается в результате выполнения НИОКР	Сокращение времени формирования плана выступлений на мероприятиях
Способы защиты интеллектуальной собственности	Регистрация исходного кода, патентование алгоритма интерфейса
Объем и емкость рынка	Объем рынка СФО на год - 20 612 644,39 руб.
Современное состояние и перспективы отрасли, к которой принадлежит представленный в ВКР продукт	Прогнозируется рост объема рынка
Себестоимость продукта	52 744,74 руб.
Конкурентные преимущества создаваемого продукта	Автоматизация процесса формирования программы конференции
Сравнение технико-экономических характеристик продукта с отечественными и мировыми аналогами	На основании конкурентных преимуществ
Целевые сегменты потребителей создаваемого продукта	Организаторы научных конференций
Бизнес-модель проекта	Модель по А. Остервальдеру
Производственный план	52 ПО в первый год, далее увеличивающийся
План продаж	В первый год продаж: 3 291 271,88руб.
Перечень графического материала:	
При необходимости представить эскизные графические материалы (например, бизнес-модель)	

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант по разделу «Концепция стартап-проекта»
(со-руководитель ВКР):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ШИП	Калашникова Татьяна Владимировна	канд. техн. наук, доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0В8Б	Петров Дмитрий Владимирович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО		
0В8Б	Петрову Дмитрию Владимировичу		
Школа	**Инженерная школа ядерных технологий**	**Отделение (НОЦ)**	**Экспериментальной физики**
Уровень образования	Бакалавриат	**Направление/ специальность**	01.03.02 Прикладная математика и информатика

Тема ВКР:

Разработка программного обеспечения для составления программы научных конференций	
Исходные данные к разделу «Социальная ответственность»:	
Введение – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации	*Объект исследования:* программное обеспечение для составления программы научных конференций *Область применения:* составление программ научных конференций *Рабочая зона:* офисное помещение *Размеры:* 18 м^2 *Количество и наименование оборудования рабочей зоны:* 1 персональный компьютер *Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:* реализация программного обеспечения на персональном компьютере
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	– Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032-78 – Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиНом 2.2.2/2.4.1340-03 – Трудовой кодекс Российской Федерации: федеральный Закон от 30 дек. 2001 г. №197-ФЗ Раздел 10 – Система стандартов безопасности труда и электробезопасность регулируется ГОСТом 12.1.009-2017
2. Производственная безопасность при разработке проектного решения: – Анализ выявленных вредных и опасных производственных факторов	– Отклонение показателей микроклимата; – Недостаточная освещённость рабочей зоны; – Пониженная световая и цветовая контрастность; – Повышенный уровень шума на рабочем месте; – Повышенный уровень статического электричества; – Повышенная запыленность воздуха рабочей зоны;
3. Экологическая безопасность при разработке проектного решения	**Анализ воздействия на литосферу:** – Утилизация компьютеров, оргтехники и бумаги; **Анализ воздействия на гидросферу:** – Производство компьютерной техники; **Анализ воздействия на атмосферу:** – Выделение вредных веществ при нагреве компонентов персонального компьютера; – Повышенная сухость воздуха при работе компьютера.

4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	– Затопление; – Землетрясение; – Короткое замыкание проводки; – Наиболее типичная ЧС: Пожар;
Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Сечин Андрей Александрович	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0В8Б	Петров Дмитрий Владимирович		

Реферат

Выпускная квалификационная выполнена на 67 страницах, содержит 12 таблиц, 11 рисунков, 26 источников, 2 приложения.

Ключевые слова: машинное обучение, анализ текста, числовые признаки, классификация, оптимизация, семантика.

Объект исследования: 6 сборников трудов международной конференции студентов, аспирантов и молодых ученых «Перспективы развития фундаментальных наук» за 2016 – 2021 гг.

Цель работы: разработать программу для автоматического формирования программы научных конференций.

Актуальность: машинное обучение эффективно используется для автоматизации решения интеллектуальных задач, что позволяет снизить издержки, сократить объем рутинных операций.

Методы проведения работы: теоретические (изучение литературы, обзор методов и моделей классификации) и практические (применение методов для построения модели).

В результате исследования: построены 4 модели классификации: 1) логистическая регрессия, 2) метод опорных векторов, 3) метод К-ближайших соседей, 4) однослойный персептрон. С применением различных методов получения числовых признаков текста: 1) TF-IDF, 2) Doc2Vec. Разработан графический интерфейс, скомпилирована программа.

Бакалаврская работа написана в текстовом редакторе Microsoft Word 2013, программный код написан на языке программирования Python.

Оглавление

Введение

Современная практика организации и проведения научных конференций включает самостоятельную регистрацию потенциальных участников конференции. При регистрации участник заполняет анкетные данные, а также указывает название доклада и выбирает секцию/подсекцию из предложенных, текст статьи может быть прикреплен значительно позднее. Чтобы убедиться в правильности выбора секции участником, необходимо произвести классификацию предоставленной работы.

Машинное обучение – область научного знания, основным преимуществом которого является возможность восстановить неизвестную зависимость по некоторой выборке, что позволяет решать задачи с трудно формализуемыми правилами. Например, такими задачами являются распознавание изображений, речи или некоторых объектов.

Одной из важнейших задач машинного обучения является задача классификации объектов, которая и рассматривается в данной работе. В нашем случае классификации является бинарной, то есть объекты делятся на два класса: 0 (у объекта отсутствует некоторое свойство) и 1 (у объекта присутствует некоторое свойство).

Анализ текста далеко продвинулся вперед благодаря технологиям преобразования текста в численные признаки, которые можно использовать в качестве входных данных для модели. Основными такими технологиями являются: TF-IDF, word2vec и doc2vec. Word2vec особенно хорошо способен понимать смысл слова, например: если из векторного представления слова «король» вычесть вектор слова «мужчина» и прибавить вектор слова «женщина», то результатом будет вектор слова «королева».

1. Обзор литературы

1.1. Постановка задачи классификации

Формальная постановка задачи классификации выглядит следующим образом: Пусть существует множество описаний объектов X и конечное множество меток классов Y, и существует неизвестная целевая зависимость $y^*: X \to Y$, значения которой известны только на объектах конечной обучающей выборки $X^m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Тогда требуется построить алгоритм $a^*: X \to Y$, который будет способен определить класс произвольного объекта $x \in X$ [3].

Чтобы определить к какому классу относится тот или иной объект, следует определить признак этого объекта. Признаком называется отображение $f: X \to D_f$, где D_f – множество допустимых значений признака. Если задано множество D допустимых значений признака $f_1, \ldots, f_n$, то вектор $\boldsymbol{x} = (f_1(x), \ldots, f_n(x))$ называется описанием признака объекта $x \in X$ [3].

1.2. Признаковое пространство

Допустимо отождествлять признаковые описания с самими объектами. При этом множество $X = D_{f1} \times \ldots \times D_{fn}$ называют признаковым пространством – совокупностью признаков, которые с достаточной полнотой отражают свойства объекта. Признаки делятся на следующие типы в зависимости от типа множества D_f [3]:

- Количественный признак: D_f - множество действительных чисел;
- порядковый признак: D_f – конечное упорядоченное множество;
- номинальный признак: D_f – конечное множество;
- бинарный признак: $D_f = \{0, 1\}$.

1.3. Линейный классификатор

Линейные классификаторы – это алгоритмы классификации, которые основаны на построении линейной разделяющей поверхности. В случае, когда множество меток классов имеет всего 2 элемента, разделяющей поверхностью будет являться гиперплоскость, которая делит пространство признаков на 2 полупространства. В других случаях, когда число классов больше двух, разделяющей поверхностью будет являться кусочно-линейная поверхность [4].

Алгоритмы классификации, которые не подходят под определение линейного классификатора, называются нелинейным классификатором.

Пусть описание объекта состоит из n числовых признаков:

$$f_j: X \to R, j = 1, 2, \dots, n$$

Тогда пространство признаковых описаний объектов есть $X = R^n$. Пусть Y – конечное множество меток классов. Положим $Y = \{-1, 1\}$, тогда линейным классификатором будет называться алгоритм $a: X \to Y$ вида [4]:

$$a(x, w) = sign \sum_{j=1}^{n} w_j f_j(x) = sign < x, w > \quad (1)$$

1.4.Метод минимизации эмпирического риска

Обучение линейного классификатора методом минимизации эмпирического риска заключается в построении алгоритма $a: X \to Y$ указанного вида по заданной обучающей выборке пар «объект, ответ» $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$, минимизирующий функционал эмпирического риска [4]:

$$Q(w, X^m) = \sum_{i=1}^{m} [a(x_i, w_i) \neq y_i] \to \min_w. \quad (2)$$

1.5.Функционал через понятие отступа

В случае двух классов Y = {-1, 1}, удобно определить для произвольного обучающего объекта $x_i \in X^m$ величину отступа [4]:

$$M(x_i, w) = y_i < x_i, w >. \quad (3)$$

В случае произвольного числа классов отступ определяется выражением [4]:

$$M(x_i, w) = < x_i, w_i > - \max_{y \in Y, y \neq y_i} < x_i, w_i >. \quad (4)$$

Понятие отступа можно расценивать как степень погруженности объекта в класс, т.е. насколько близок или далек объект к границе классов. Чем больше значение $M(x_i, w)$, тем дальше объект находится от границы классов, и тем ниже вероятность ошибки. Отрицательным отступ может быть тогда и только тогда, когда алгоритм *a(x)* допустил ошибку для объекта x_i.

Это наблюдение помогает записать функционал (2) эмпирического риска в следующем виде [4]:

$$Q(w, X^m) = \sum_{i=1}^{m} [M(x_i, w) < 0] \rightarrow \min_w. \quad (5)$$

1.6.Замена пороговой функции потерь

Минимизация функционала *Q(w)* по вектору весов сводится к поиску максимальной совместной подсистемы в системе неравенств. Эта задача является NP-полной и может иметь очень много решений, поскольку минимальное число ошибок может реализоваться на различных подмножествах объектов. Однако абсолютно точное решение этой задачи, и, тем более, нахождение всех её решений, в большинстве приложений не представляет практического интереса. Обычно вполне устраивает приближённое решение, достаточно близкое к точному. Наиболее известные методы обучения линейного классификатора связаны с заменой пороговой функции потерь её различными непрерывными аппроксимациями [4]:

$$[M < 0] \leq L(M), \quad (6)$$

где L – аппроксимирующая функция.

Если функционалом будем являться непрерывная или гладкая функция, к тому же еще и невозрастающая, то оптимизационную задачу будет решать намного удобнее.

После замены функции потерь минимизируется не сам функционал эмпирического риска, а его верхняя оценка [4]:

$$Q(w, X^m) \leq \tilde{Q}(w, X^m) = \sum_{i=1}^{m} L\big(M(x_i, w)\big). \quad (7)$$

1.7.Регуляризация

Когда на обучающей выборке хорошо объясняется принадлежность объектов к классам, на выборке, не участвующей в обучении, результат становится относительно плохим (переобучение модели). Такое явление происходит из-за выявления закономерностей на обучающей выборке, которые не присущи генеральной совокупности.

Для борьбы со слишком большими значениями нормы вектора весов введем понятие регуляризации. Тогда формулу (7) можно записать в следующем виде:

$$Q(w, X^m) \leq \tilde{Q}(w, X^m) = \sum_{i=1}^{m} L\big(M(x_i, w)\big) + \gamma||w||^{\rho} \to \min_{w}, \quad (8)$$

где ρ – степень регуляризации, которая определяет класс методов оптимизации, а γ – параметр регуляризации, подбираемый исходя из априорных соображений, либо по скользящему контролю [4].

1.8.Логистическая регрессия

По большей части методы обучения линейных классификаторов схожи. Отличие методов заключается в основном реализацией

регуляризации и выбором ее способов, выбором аппроксимирующей функции для замены пороговой функции потерь, а также выбором метода решения численного решения оптимизационной задачи.

Методом линейной классификации является логистическая регрессия. Данный метод помимо классификации позволяет оценить апостериорные вероятности принадлежности объектов к классам.

Для случая двух классов задача обучения классификатора также строится на минимизации эмпирического риска где функция потерь выглядит таким образом [6]:

$$Q(w, X^m) = \sum_{i=1}^{m} \ln(1 + \exp(-y_i < x_i, w >)) \to \min_w. \quad (9)$$

После того, как решение w найдено, становится возможным не только проводить классификацию $a(w) = sign < x, w >$, для произвольного объекта x, но и оценивать апостериорные вероятности его принадлежности классам [6]:

$$P\{y|x\} = \sigma(y < x, w >), \quad (10)$$

где

$$\sigma(z) = \frac{1}{1+e^{-z}}. \quad (11)$$

1.9.Метод к-ближайших соседей

Метод к-ближайших соседей является не линейным методом классификации, а метрическим, однако это один из популярных методов классификации, поэтому рассмотрим и его. Суть метода к-ближайший соседей состоит в том, что для произвольного объекта x вычисляется расстояние между остальными объектами, тогда объекту x присваивается класс, частота которого большая среди k самых близких соседей. Число k соседей принято выбирать нечетным, дабы избежать случаев, когда частоты классов соседей равны.

Пусть на множестве обучающей выборки объектов X^m задана функция расстояния $\rho(x, x')$. Для произвольного объекта u из тестовой выборки вычислим расстояния и обозначим через $x_{i,u}$тот объект, который является i-м соседом объекта u, через $y_{i,u}$ – ответ, для i-го соседа. В общем виде алгоритм ближайших соседей есть [7]:

$$a(u) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{m} [x_{i,u} = y] w(i, u). \quad (12)$$

Проблемой данного метода является выбор метрики. Когда количество признаков слишком большое, и расстоянием является сумма отклонений по отдельным признакам, возникает проблема выбора k ближайших соседей, так как с большой вероятностью значения расстояний имеют схожие значения. Тогда выбор k ближайших соседей становится почти случайным.

Часто в практических задачах необоснованно выбирается евклидова метрика. Тогда не стоит забывать, что признаки объектов должны быть «одной формы», в идеальном случае должно быть произведено нормирование. В таком случае не произойдет доминации признака с наибольшим числовым значением в метрике и игнорирования признаков с малыми числовыми значениями.

В случае нормировки встает другой вопрос об одинаковой важности всех признаков.

1.10. Метод опорных векторов

Метод опорных векторов соответствует кусочно-линейной аппроксимации [2]:

$$[M < 0] \leq (1 - M)_+, \quad (13)$$

где + означает положительные значения. Применяется регуляризация с квадратичной нормой. Задача оптимизации решается как задача квадратичного программирования.

1.11. Метод градиентного спуска

Градиентный спуск – это алгоритм оптимизации, суть которого заключается в минимизации функции потерь посредством нахождения локального минимума при движении вдоль градиента функции.

Пусть есть функция потерь $L(w, X^m)$, тогда задача минимизации выглядит так [5]:

$$Q(w, X^m) = \sum_{i=1}^{m} L(w, x_i) \to \min_w. \quad (14)$$

Далее для решения задачи вычисляется градиент функционала [5]:

$$\nabla Q(w, X^m) = \left(\frac{\partial Q(w, X^m)}{\partial w_j}\right)_{j=0}^{m}, \quad (15)$$

Теперь уточняем веса w и выполняем оценку функционала следующим образом [5]:

$$w_{t+1} = w_t - h \sum_{i=1}^{m} \nabla L(w_t, x_i), \quad (16)$$

$$Q(w_{t+1}, X^m) = \sum_{i=1}^{m} L(w_{t+1}, x_i), \quad (17)$$

где h – градиентный шаг, называемый также темпом обучения. Алгоритм выполняется до тех пор, пока не будет достигнута сходимость Q или w [5].

Отличие стохастического градиентного спуска от обычного заключается в том, что градиент функции, которую необходимо оптимизировать, вычисляется не как сумма градиентов каждого элемента выборки, а как градиент одного случайного элемента. Тогда при вычислении весов будет вычитаться не сумма градиентов, а градиент одного случайного элемента [2]:

$$w^{(t+1)} = w^t - h\nabla L\left(w^{(t)}, x_i\right). \quad (18)$$

Так как теперь направление измерения w будет определяться за *O(1)*, подсчет Q на каждом шаге будет слишком дорогостоящим. Для того, чтобы ускорить оценку Q, будем использовать приближенную рекуррентную формулу. Пример таких формул [2]:

- Среднее арифметическое: $\bar{Q}_t = \frac{1}{t}\varepsilon_t + (1 - \frac{1}{t})\bar{Q}_{t-1}$
- Экспоненциальное скользящее среднее: $\bar{Q}_t = \lambda\varepsilon_t + (1 - \lambda)\bar{Q}_{t-1}$,

где λ – темп забывания предыстории ряда, а $\varepsilon_t = L(w_t, x_i)$ – потеря [2].

1.12. Однослойный персептрон

Однослойный персептрон — это линейный алгоритм классификации, принцип работы которого основан на модели нервной клетки - нейрона. Представляет собой пример нейронной сети с одним скрытым слоем.

Аппроксимация выглядит следующим образом [9]:

$$[M < 0] \leq \frac{2}{1+e^{aM}}, \quad (19)$$

где параметр *a* задается из априорных соображений. Задача оптимизации решается с помощью градиентных методов [9].

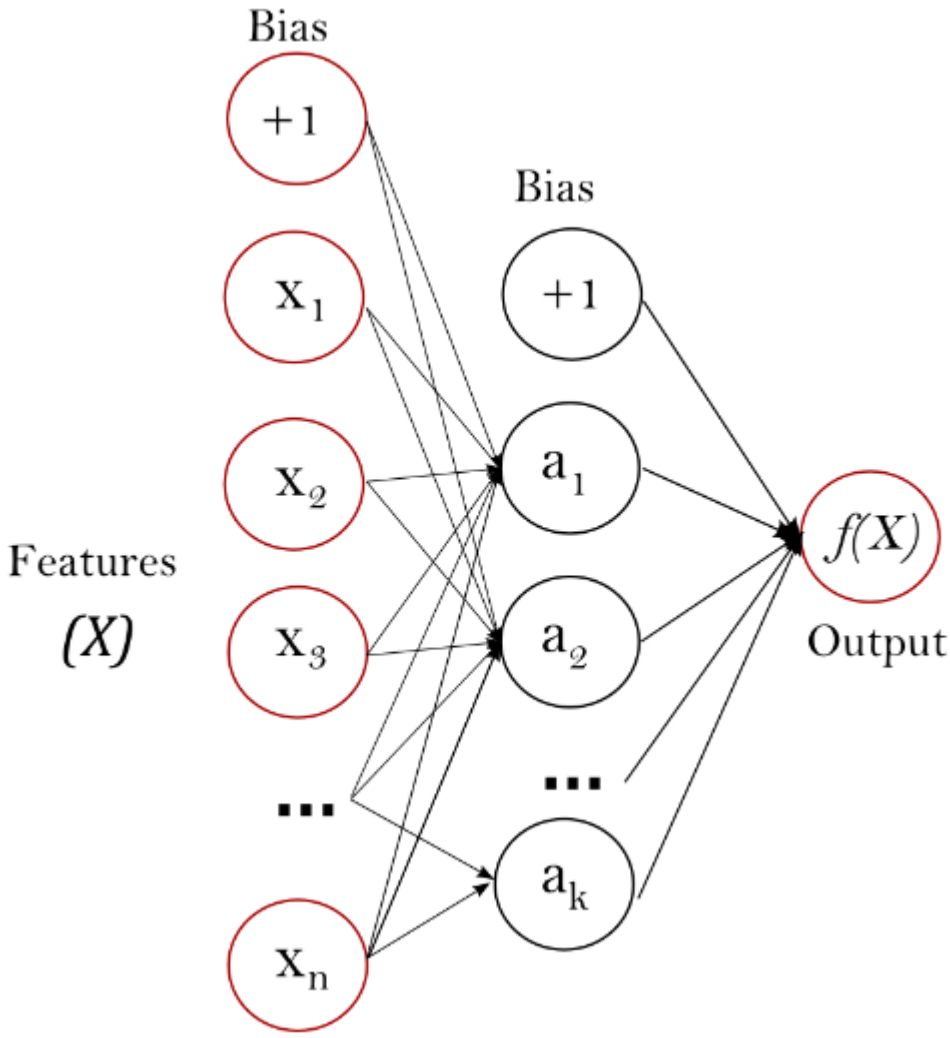

Рисунок 1. Архитектура нейронной сети

1.13. Общая задача преобразования текстовой информации в численные признаки

Имеется набор документов $d_i, i = 1, \dots, m$, где m - количество документов, входящих в корпус документов D. В каждом документе находятся слова $t_j, j = 1, \dots, n_j$, где n_j – число слов, входящих в документ d_i. Тогда следует получить признаки $x_i \epsilon X$ для каждого документа d_i.

1.14. TF-IDF

TF-IDF – статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален частоте

употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции. Мера TF-IDF часто используется в задачах анализа текстов и информационного поиска, например, как один из критериев релевантности документа поисковому запросу, при расчёте меры близости документов при кластеризации.

TF – отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова t_i в пределах отдельного документа:

$$tf(t,d) = \frac{n_t}{\sum_k n_k}, \quad (20)$$

где n_t – число вхождений слова t в документ, а в знаменателе – общее число слов в данном документе.

IDF – инверсия частоты, с которой некоторое слово встречается в документах коллекции. Основоположником данной концепции является Карен Спарк Джонс. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF:

$$idf(t,D) = log\frac{|D|}{|\{d_i \in D | t \in d_i\}|}, \quad (21)$$

где $|D|$ – число документов в коллекции, а знаменатель – число документов из коллекции D, в которых встречается t (когда $n_t \neq 0$).

В конечном итоге, произведение двух сомножителей и является мерой TF-IDF:

$$tf - idf(t,d,D) = tf(td) \times idf(t,D) \quad (22)$$

Мера TF-IDF присваивает большие веса словам с высокой частотой в рамках конкретного документа, но с низкой частотой употреблений в остальных.

1.15. Doc2Vec

Doc2Vec - программный инструмент анализа семантики естественных языков. В отличие от TF-IDF способен понимать семантический смысл слов. Основной идеей Doc2Vec является отказ от подсчета количества слов, заменяя его предсказанием окружающих слов относительно каждого слова [10].

Тогда решим следующую задачу:

Предсказать окружающие слова в окне длиной m каждого слова. Под окном длины m понимается количество слов вокруг слова, которые нужно предсказать. Максимизируем следующий функционал:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=-m,j\neq 0}^{m}\ln(p(w_{t+1}|w_t)), \quad (23)$$

где θ – все переменные, которые мы оптимизируем, T – количество слов в документе, w_t – каждое слово из документа. Оптимизируемые вектора можно увидеть, если расписать следующее выражение:

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{W}\exp(u_0^T v_c)}, \quad (24)$$

где o – предсказываемое окружающее слово, c – слово, относительно которого предсказываются слова (т.н. центральное слово), u–вектор центрального слова, v – вектор окружающего слова. Тогда получаем для каждого слова два векторных представления: одно, в случае если слово центральное, второе, в случае если слово не является центральным.

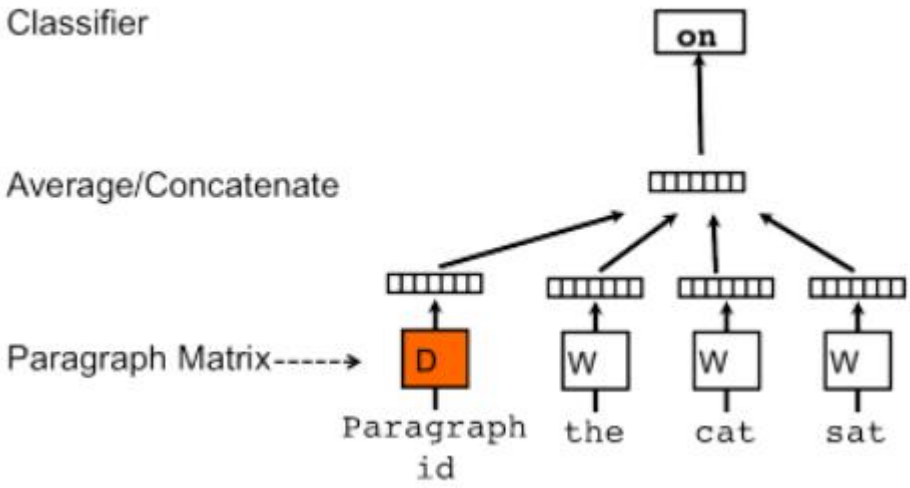

Рисунок 2. Схема Doc2Vec

1.16. Оценка правильности классификации

Построив модель классификации и получив предсказания классов для тестовой выборки, встает вопрос о правильности произведенной классификации. Для оценки качества алгоритмов необходимо ввести численную метрику.

1.17. Точность

Не стоит путать два понятия точности *Accuracy* и *Precision*. Простейшая метрика *Accuracy* показывает долю правильно классифицированных объектов P относительно всех объектов N [8]:

$$Accuracy = \frac{P}{N}, \tag{25}$$

В свою очередь, *Precision* отражает долю правильно классифицированных объектов относительно всех объектов, которые модель отнесла к этому классу [8]:

$$Precision = \frac{TP}{TP+FP}, \tag{26}$$

где *TP (true predict)* – число правильно отнесенных объектов к текущему классу, а *FP (false predict)* – число ложно классифицированных объектов по этому же классу [8].

1.18. Полнота

Метрика полноты или же *Recall* показывает отношение правильно классифицированных объектов к общему числу объектов этого класса:

$$Recall = \frac{TP}{TP+FN}, \quad (27)$$

где FN (false negative) – число ложно-отрицательных объектов.

1.19. F-мера

Легко предположить, что чем выше точность и полнота, тем лучше модель проводит классификацию. Но на практике достичь стопроцентной точности обоих показателей невозможно. Тогда приходится искать некий баланс. *F*-мера – это метрика, которая содержит в себе информацию о точности и полноте и представляет собой их гармоническое среднее:

$$F = 2\frac{Precision \times Recall}{Precision+Recall}. \quad (28)$$

По формуле (28) *Precision* и *Recall* являются равнозначными, т.е. при уменьшении точности или полноты показатель F-меры будет падать одинаково. *F*-мера будет равна 1 при *Precision = 1* и *Recall = 1*, тогда мера будет называться F_1. В случае, когда необходимо определить приоритет между точностью и полнотой, мера F_β выглядит так:

$$F_\beta = (\beta^2 + 1)\frac{Precision \times Recall}{\beta^2 Precision+Recall}, \quad (29)$$

где β принимает значения в диапазоне $0 < \beta < 1$, если точность приоритетнее полноты, или $\beta > 1$, если приоритетнее полнота.

1.20. Confusion matrix

Confusion matrix или матрица неточностей – это матрица, имеющая размерность N на N, где N – количество классов. С помощью такой матрицы, когда количество классов невелико, удобно наглядно анализировать действия классификатора. Строками такой матрицы

являются решения классификатора, а столбцы зарезервированы за экспертными решениями [8].

2. Результаты исследования

2.1. Сбор данных

В качестве данных были приняты 6 сборников трудов, каждый из которых состоит из 7 томов, международной конференции студентов, аспирантов и молодых ученных «Перспективы развития фундаментальных наук» за период с 2016 по 2021 год. Сборники трудов скачаны из архива официального сайта конференции в формате PDF [1]. Тексты сборников были получены с помощью пакета pdfminer для языка программирования Python. Далее из всех файлов были извлечены аннотации к работам. Суммарно получилось 2879 аннотаций. Листинг программы по сбору данных представлен в приложении А

2.2. Подготовка данных к обучению

Перед обучением модели необходимо провести очистку данных от «мусора». Были проведены следующие действия:

- Удалены переносы строк;
- Удалены знаки пунктуации;
- Удалены переносы строк;
- Удалены служебные слова;
- Тексты были переведены на английский язык для однородности.

2.3. Анализ данных

Ниже приведены классы и их числовые описания:

- Физика – 1;
- Химия – 2;
- Математика – 3;

- Биология и фундаментальная медицина – 4;
- Экономика и управление – 5;
- Строительство и архитектура – 6;
- IT-технологии и электроника – 7;

На рисунке 2 приведено распределение объектов по классам:

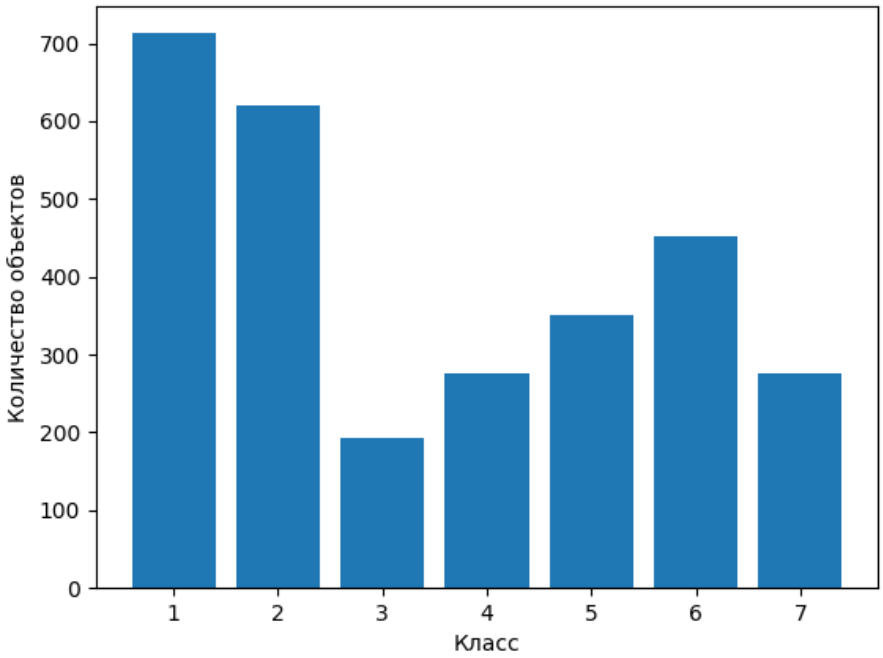

Рисунок 3. Распределение объектов по классам

Можно заметить, что датасет не сбалансирован.

2.4.Генерация признаков TF-IDF

Чтобы сгенерировать признаки TF-IDF, была использована функция TfidVectorizer из библиотеки sklearn для языка программирования Python. Оценка TF была вычислена по формуле (20). Оценка IDF – по формуле (21). Конечная оценка признаков TF-IDF получена по формуле (22). В результате имеем матрицу с размером 2879×14871, где 2879 – число строк – количество используемых аннотаций, а 14871 – число столбцов – размер словаря (количество уникальных слов).

2.5.Генерация признаков Doc2Vec

Для генерации признаков была использована функция Doc2Vec из библиотеки genism для языка программирования Python. Сначала была сформирован корпус документов из всех представленных аннотаций, затем была обучена модель. В итоге имеем матрицу размером 2879×50, где 2879 – число строк – количество используемых аннотаций, а 50 – число столбцов – опытным путем подобранный размер вектора признаков для аннотаций.

2.6.Построение моделей

В данном случае имеем дело с несбалансированной выборкой – присутствуют как и большие классы с 713 элементами, так и малые с 192, при среднем размере класса 411. Тогда сделаем предположение о том, что у нас есть наивный классификатор, который все считает, что все данные относятся к самому большому классу, тогда его точность будет равна:

$$accuracy = \frac{N_{true}}{N} = \frac{713}{2879} = 24{,}77\%$$

Примем это значение как базовое решение, то есть классификаторы, которые будут иметь точность ниже 24,77% будем считать плохими.

Для проверки классификаторов разделим выборку на тренировочную и тестовую части в соотношении 80% – тренировочная выборка, 20% – тестовая выборка.

В качестве модели построения классификатора выберем следующие алгоритмы:

- Метод опорных вектором со стохастическим градиентным спуском;

- Метод к-ближайших соседей;
- Логистическая регрессия;
- Однослойный персептрон.

Построим все четыре модели, которые мы обучим на тренировочной выборке, используя пространства признаков по отдельности. Используя обученные модели, сделаем предсказания для тестовой выборки и установим точность этих предсказаний. Так как выборка не сбалансирована, точность будем вычислять сначала для каждого класса по формулам (26), (27), (28), а итоговое значение как макро среднее по мере F_1, так как все классы важны. Результат приведен в таблице 1.

Таблица 1. Результаты классификаторов на несбалансированной выборке

Пространство признаков	**Классификатор**	**Точность, %**
TF-IDF	Метод опорных вектором со стохастическим градиентным спуском	81,98%
	Метод к-ближайших соседей	75,49%
	Логистическая регрессия	77,46%
	Однослойный персептрон	81,18%
Doc2Vec	Метод опорных вектором со стохастическим градиентным спуском	58,87%
	Метод к-ближайших соседей	53,01%
	Логистическая регрессия	57,83%
	Однослойный персептрон	64,04%

Все четыре классификатора оказались точнее базового решения. Метод к-ближайших соседей оказался худшим решением из предложенных в обоих пространствах признаков, а однозначно лучшего определить не получилось – метод опорных векторов со стохастическим градиентным спуском оказался самым точным с признаки TF-IDF с отрывом в 0,8% от однослойного персептрона, и однослойный персептрон в пространстве Doc2Vec с отрывом в 5,17% от метода опорных векторов. В общем, можно считать, что все полученные классификаторы оказались удачными.

Сравнивая разные методы получения числовых признаков, приходим к выводу, что дальше можно не использовать Doc2Vec.

Посмотрим на матрицу неточностей для каждого классификатора, числовые признаки которых были получены TF-IDF методом:

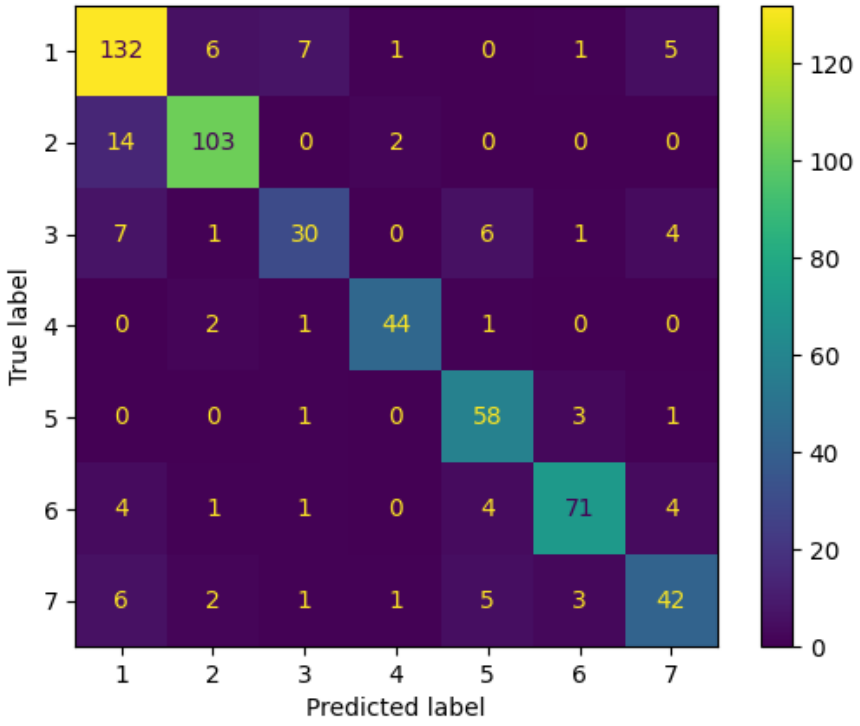

Рисунок 4. Матрица неточностей для метода опорных векторов

По приведенной матрице можно установить, что классификатор плохо справляется с объектами 3 и 7 класса, хорошо с объектами классов

1, 2, 6, и отлично с объектами классов 4, 5. Также можно достаточно просто рассчитать точность и полноту для каждого класса, а именно: точность – это отношение диагонального элемента к сумме элементов этой строки, полнота – отношение диагонального элемента матрицы к сумме элементов этого столбца.

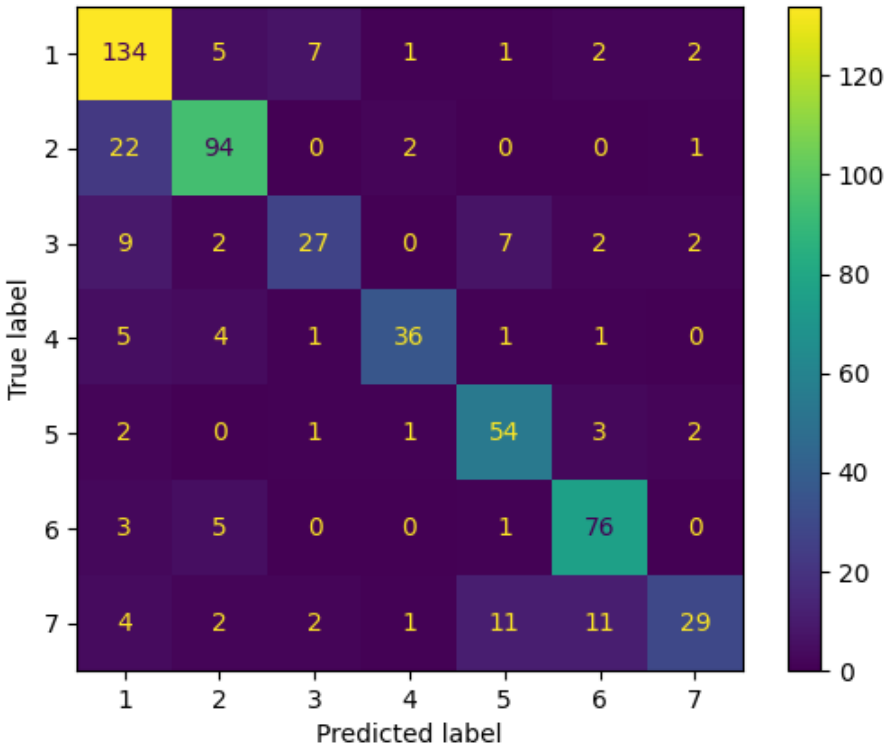

Рисунок 5. Матрица неточностей для метода к-ближайших соседей

Метод к-ближайших соседей еще хуже справляется с теми же классами 3 и 7, но также имеет трудности с 4-м классом. С объектами классов 1, 2, 5, 6 справился хорошо.

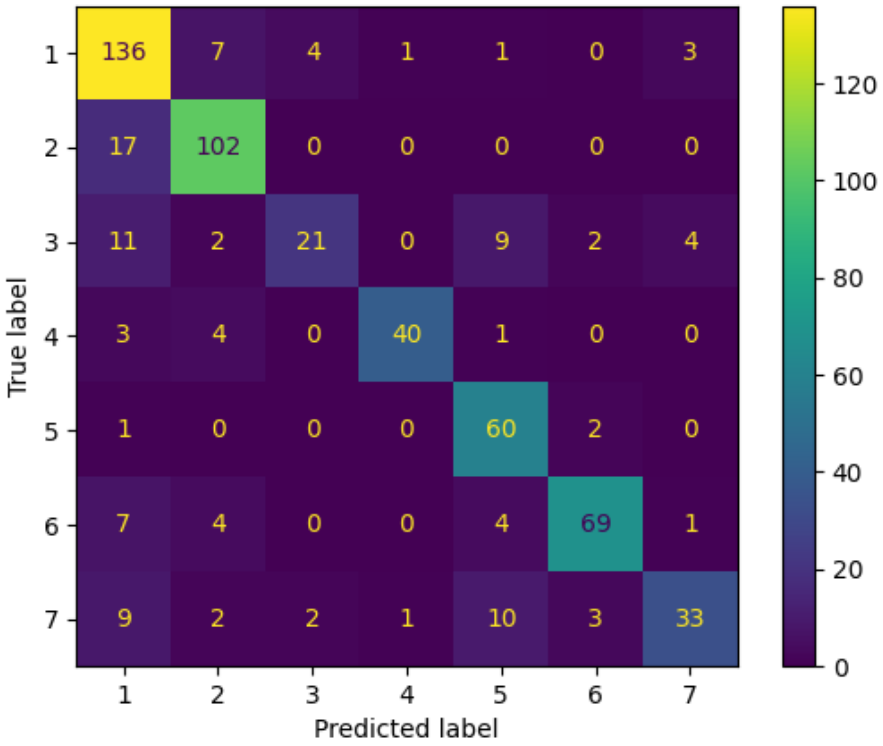

Рисунок 6. Матрица неточностей для логистической регрессии

Модель логистической регрессии плохо определяет объекты классов 3 и 7, и хорошо описывает объекты остальных классов.

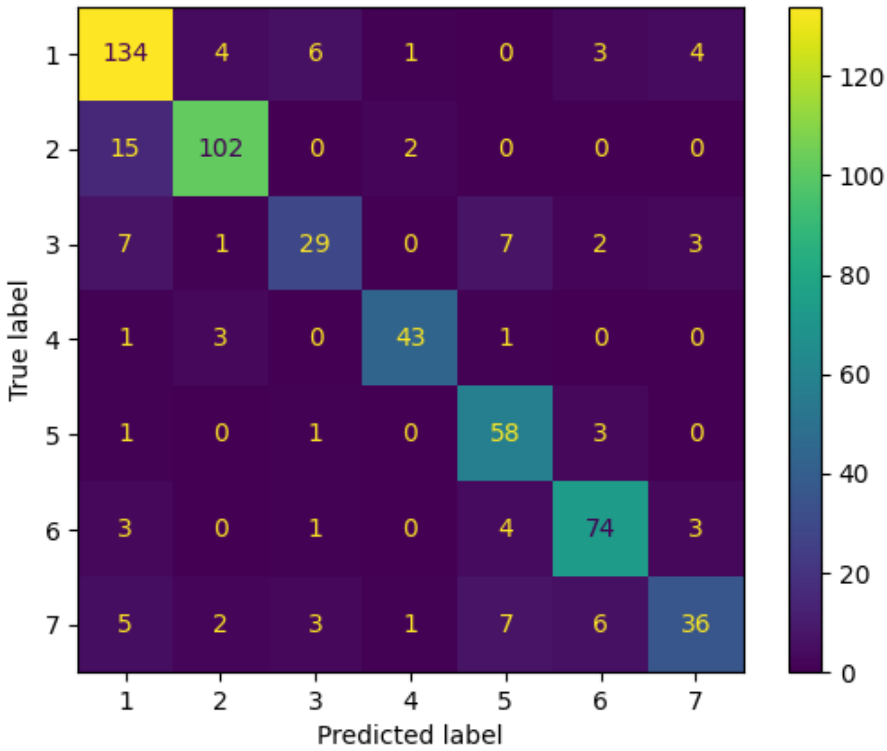

Рисунок 7. Матрица неточностей для однослойного персептрона

Аналогичный результат имеет однослойный персептрон.

Проанализировав 4 матрицы, можно сделать вывод, что классы 3 – математика и 7 – IT-технологии и электроника имеют слабо выраженные признаки, или исходные данные имеют неправильную разметку, и из-за малого количества объектов одна неверная классификация сильнее ухудшает конечную точность.

Ниже приведен график зависимости точности классификации от количества эпох обучения для однослойного персептрона.

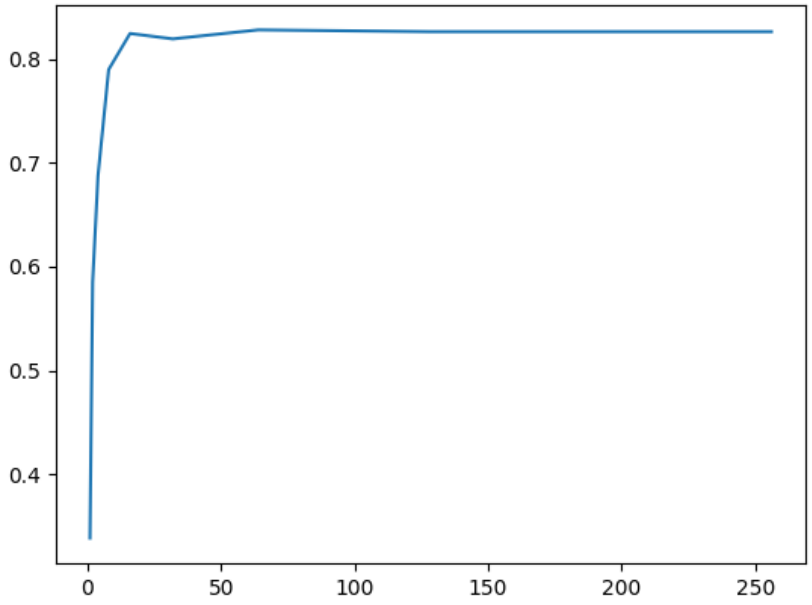

Рисунок 8. График зависимости точности классификации от количества эпох обучения

Теперь приведем выборку в балансное состояние, чтобы более точно оценить обобщающую способность. Так как минимальный размер класса 192 элемента, то сделаем остальные классы такого же размера случайным образом. Тогда базовое решение будет иметь точность:

$$accuracy = \frac{N_{true}}{N} = \frac{192}{1344} = 14{,}29\%$$

Таблица 2. Результаты классификаторов на сбалансированной выборке

Классификатор	**Точность, %**
Метод опорных вектором со стохастическим градиентным спуском	82,01%
Метод к-ближайших соседей	78,92%
Логистическая регрессия	80,99%
Однослойный персептрон	81,54%

Снова все классификаторы оказались лучше базового решения. Метод к-ближайших соседей и логистическая регрессия показали прирост точности 3,43% и 3,53% соответственно, а метод опорных вектором со стохастическим градиентным спуском и однослойный персептрон показали незначительное изменение на 0,03% и 0,36% соответственно.

Так как при случайной балансировке данных часть признаков теряется, то следует провести обучение моделей несколько раз и взять среднее значение точности. Проведем обучение 50 раз:

Таблица 3. Усредненные результаты классификаторов на сбалансированной выборке

Классификатор	**Точность, %**
Метод опорных вектором со стохастическим градиентным спуском	77,99%
Метод к-ближайших соседей	72,49%
Логистическая регрессия	78,78%
Однослойный персептрон	79,81%

Сравнивая полученный результат с первым наблюдением, можно сказать, что для сбалансированной выборки лучше подходит

классификация однослойным персептроном, а для несбалансированной – метод опорных вектором со стохастическим градиентным спуском.

В качестве первого доклада выбирается первая предоставленная работа, остальные сортируются, основываясь на косинусном расстоянии между векторами-признаками.

2.7.Пример работы

Пример полученного набора данных – аннотации и их классы

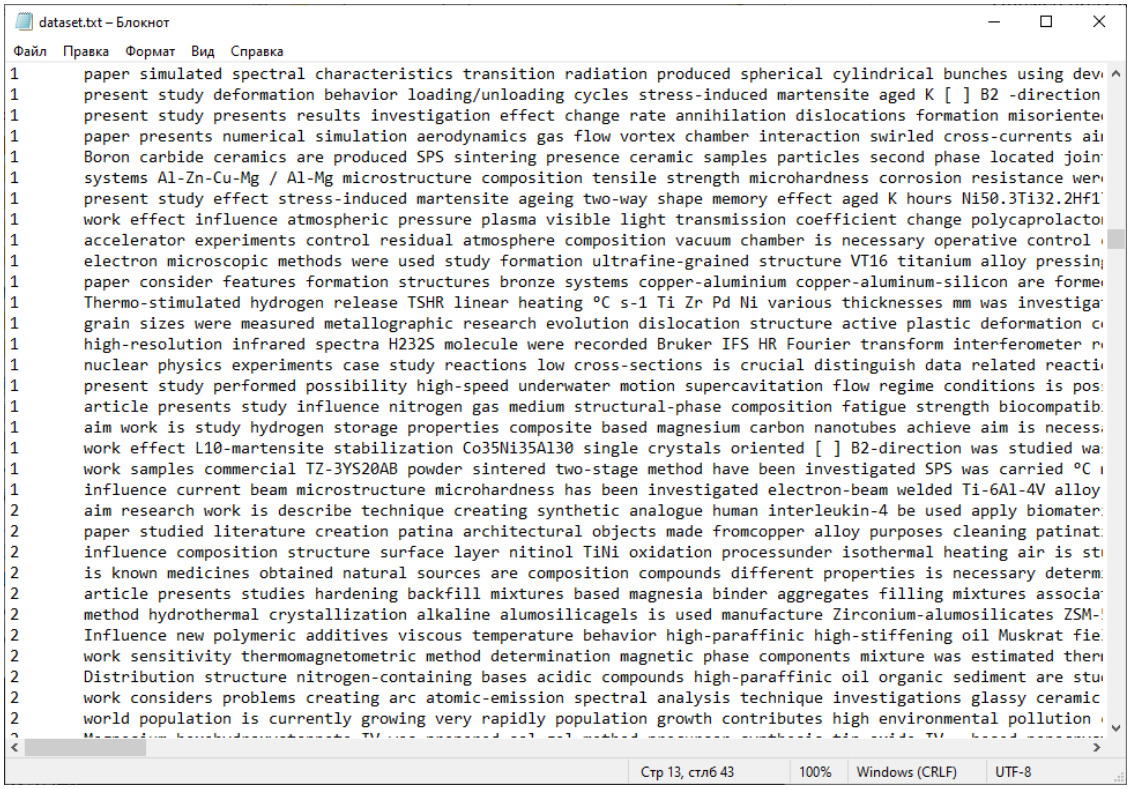

Рисунок 9. Пример полученного набора данных

Графический интерфейс написан при помощи библиотеки PyQt5 для языка программирования Python и выглядит следующим образом:

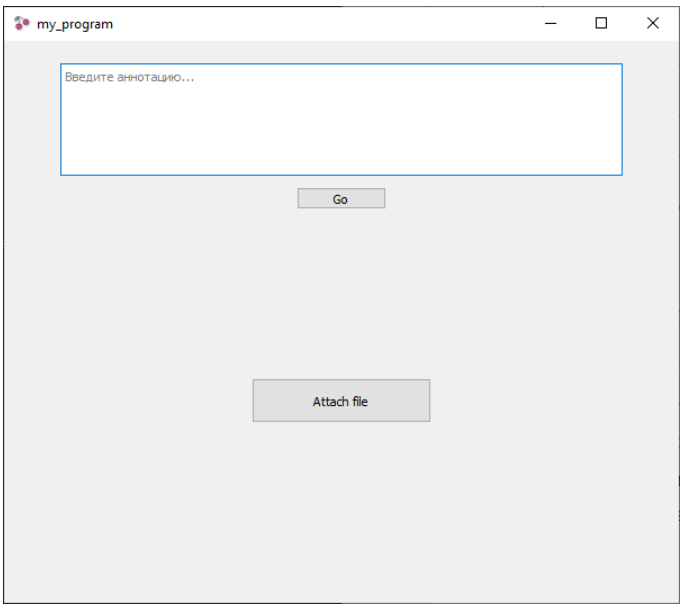

Рисунок 10. Графический интерфейс

Программа может принимать как отдельные аннотации, так и целые файлы с наборами текстов. Листинг графического интерфейса представлен в приложении Б.

После получения данных, вычисляем их признаки, обучаем модель, 20% данных – тестовая выборка сохраняется в текущей директории программы и используется для проверки моделей.

По кнопке «Attach file» загружаем тестовую выборку, получаем всплывающее окно «Done», уведомляющее о успешной загрузке файла. Затем в текущей директории программы появляется 3 файла:

- result_knb.txt – результат классификатора методом ближайших соседей;
- result_sgd.txt – результат классификатора методом опорных векторов;
- result_lr.txt – результат классификатора логистической регрессии;
- result_slp – результат классификатора однослойного персептрона.

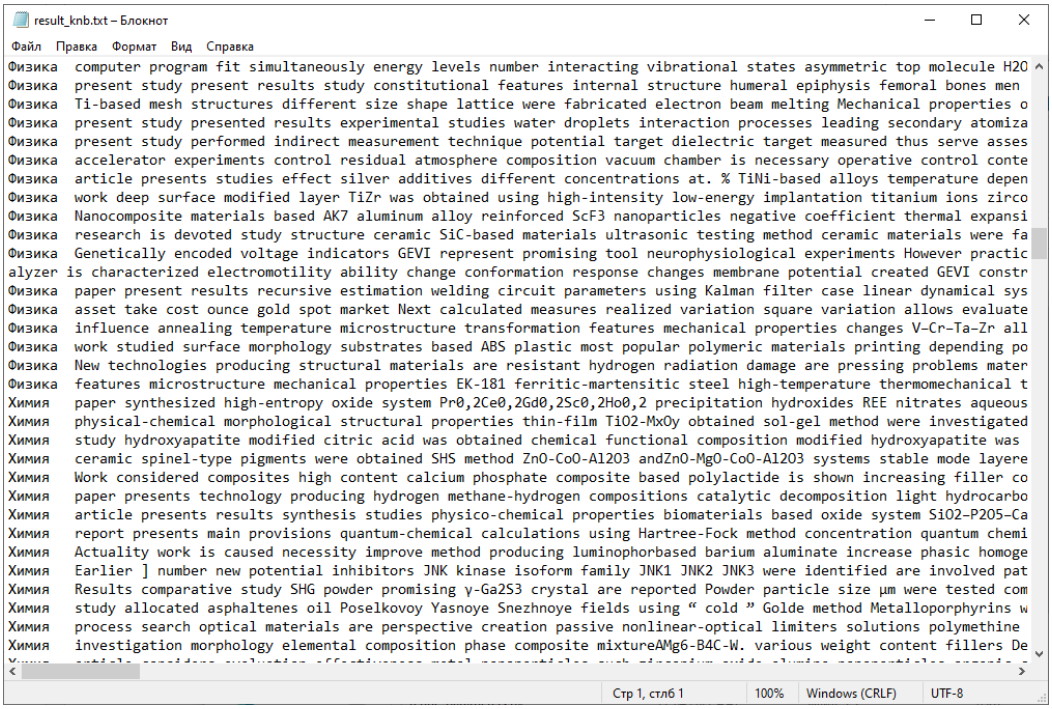

Рисунок 11. Результат работы программы

В файлах-ответах вместо аннотаций должны были быть названия работ, но их не удалось получить.

2.8.Заключение

Были собраны данные из 6 сборников трудов, состоящие из 2879 аннотаций к работам, которые были предварительно обработаны.

Для каждой аннотации были построены признаки TF-IDF и Doc2Vec. TF-IDF метод построения числовых признаков оказался лучшим. В результате каждый объект имеет 14871 признак.

Для несбалансированной и сбалансированной выборок были построены следующие линейные модели: логистическая регрессия, метод к-ближайших соседей, метод опорных вектором со стохастическим градиентным спуском, однослойный персептрон.

По результатам исследования на сбалансированной и несбалансированной выборках была выбраны лучшие модели: для сбалансированной выборки – однослойный персептрон с точностью

79,81%, для несбалансированных данных – метод опорных вектором со стохастическим градиентным спуском с точностью 81,98%.

Ни одним из способов не удалось добиться достаточной точности в 95%. Так как пространство признаков строилось двумя методами, можно сделать вывод о том, что на этапе регистрации участники выбирают секцию неправильно.

Программа, написанная на языке программирования Python, была преобразована в исполняемый файл с расширением «.exe», чтобы ее работоспособность не зависела от наличия интерпретатора языка программирования.

3. Концепция стартап-проекта

Идея разработки программы для ЭВМ для автоматического составления плана выступлений на научных конференциях. Планируется использование программы для ЭВМ организаторами научных конференций с целью оценки правильности выбора секции участником конференции и построения плана выступления участников на конференции таким образом, чтобы тема каждого следующего выступающего была максимально близка к теме предыдущего спикера для лучшего восприятия слушателями. Цель данной разработки – упрощение и ускорение формирования плана выступления участников научных конференций.

3.1. Описание продукта как результата НИР

В настоящее время автоматизация затрагивает все больше различных сфер деятельности человека. Однако, есть сферы, в которых до сих пор все делается вручную. Например, составление плана выступлений на конференциях. Проверка каждой заявки на участие в конференции вручную – очень долгий и утомительный процесс, особенно, когда количество участников исчисляется сотнями.

Современная практика организации и проведения научных конференций включает самостоятельную регистрацию потенциальных участников. При регистрации участник заполняет анкетные данные (персональные данные, вуз, ...), прикрепляет аннотацию к работе, а также указывает название доклада и выбирает секцию/подсекцию из предложенных, текст статьи или тезисы доклада могут быть прикреплен значительно позднее.

На основании предоставленных материалов руководитель секции организует процедуру рецензирования и принимает решение о

включении заявки в программу конференции. Общая программа конференции формируется в два этапа:

1. руководители секций формируют программы секций;
2. руководитель конференции формирует общую программу конференции.

При этом доклады внутри секции распределяются равномерно по дням (например, 1, 2, 3), внутри дня выстраиваются по близким тематикам, в исключительных случаях руководитель конференции может перенести доклад из одной секции в другую.

В рамках данной работы разработана программа для ЭВМ, которая:

1. Не имея основного текста, оценивает релевантность заявки потенциального участника конференции (правильность выбора секции/подсекции для выступления);
2. Составляет план выступлений внутри секций;
3. Составляет общий план выступлений.

3.2. Интеллектуальная собственность

Объектом интеллектуальной собственности является программа для ЭВМ. Отношения, возникающие в связи с правовой охраной и использованием программ для ЭВМ, регулирует Гражданский кодекс РФ, часть 4, ст. 1261 и ст. 1262.

Полностью запатентовать программное обеспечение невозможно, т.к. ПО не является изобретением согласно ст. 1350 п. 5 ГК РФ. Разрабатываемая программа состоит из алгоритма, исходного кода и интерфейса. Только алгоритм и интерфейс ПО признаются патентоспособными. Исходный код охраняются авторским правом. Ключевое различие заключается в том, что авторское право защищает

автора от копирования программы, а патентное право защищает идею программы в целом [12].

Таким образом, для защиты интеллектуальной собственности принято решение о регистрации исходного кода (программы для ЭВМ) и патентование алгоритма и интерфейса программного обеспечения.

3.3. Целевые сегменты потребителей

Целевым сегментом потребителей данного продукта являются организаторы любого рода периодических мероприятий. Концепцией таких мероприятий должны основываться на выступлениях участников. Также для целевой аудитории выдвигается одно ограничение – наличие архивных материалов.

В структуре организаторов конференций, являющихся целевым сегментом, можно выделить несколько заинтересованных лиц. Во-первых, это руководители секций и руководитель конференции, так как им будет проще и быстрее составлять план выступления участников на конференции, во-вторых, это спонсоры, которые выделяют средства на проведение подобных конференций, так как для проведения таких конференций потребуется меньше времени, а, следовательно, меньший расход средств.

3.4. Объем и емкость рынка

Проведём оценку рынка методом снизу-вверх. Для этого метода данные возьмем из открытого каталога научных конференций, выставок и семинаров «Конференции.ru» [11] о количестве проведенных мероприятий на территории России. За 2021 год количество проведенных мероприятий составило 3908. Примем все мероприятия уникальными и периодическими. Тогда TAM составляет 3908 * 52 744,74 = 206 126 443,92 руб., исходя из стоимости разрабатываемого продукта (52 744,74 руб.).

Далее, оценим количество мероприятий, входящих в TAM, у которых количество участников больше 100 человек, чтобы предлагаемое решение имело практический смысл. По результатам исследования среднее количество участников составляет примерно 245 человек, при этом примерно 50% (1954) мероприятий имеют больше чем 100 участников. Тогда SAM составляет 3908 * 0,5 * 52 744,74 = 103 063 221,96 руб.

Исходя из географии проведения мероприятий, примерно с 20% из них проводятся в Сибирском федеральном округе, а значит возможно установить сотрудничество в течении года. Таким образом SOM составляет 3908 * 0,5 * 0,2 * 52 744,74 = 20 612 644,39 руб.

3.5. Анализ современного состояния и перспектив развития отрасли

За последние годы в промышленности наблюдается активное внедрение технологий Индустрии 4.0 (технологии искусственного интеллекта, Интернет вещей, автоматизация бизнес-процессов). В 2020 году было проведено 3379 мероприятий, а в 2021 году уже 3908. Прирост составил 15,66%. Так как самым лучшим типом мероприятия, к которому подходит разрабатываемое ПО, являются научные конференции, упор будем делать на перспективы именно этой отрасли.

На данный момент крупные компании самостоятельно разрабатывают различного рода интеллектуальные системы. В сравнении с прошлыми десятилетиями, статус профессии ученого постепенно повышается. По данным опроса Superjob, проведенного в 2019 году, 57% жителей России считают, что ученым быть престижно. В 2020 году количество выпускников вузов, принятых на работу в научные организации, достигло 14 015 человек, увеличившись по сравнению с предыдущим годом на 25,5%. Самым «неурожайным» в этом плане признан 2017 год — в научные учреждения устроились всего

9 985 молодых специалистов. За последнее десятилетие на 9,1% также выросла численность научных сотрудников в возрасте до 39 лет (включительно) [15].

Позитивные тенденции наблюдаются практически во всех отраслях. Особенно они заметны в тех сферах, которые связаны с получением услуг через интернет — в цифровую среду перешло все, что было возможно перенести.

Подытоживая вышесказанное, можно сделать вывод, что с каждым годом растет количество научных исследований, а, следовательно, и количество научных конференций. Помимо роста количество научных трудов и мероприятий, растет уровень цифровизации и автоматизации. Таким образом, можно сделать вывод о наличии перспектив развития данной отрасли.

3.6. Планируемая стоимость продукта

Для определения планируемой стоимости ПО была рассчитана себестоимость по затратному методу ценообразования.

Разработка программного обеспечения требует привлечения следующих специалистов: специалист в области машинного обучения, backend-разработчик и frontend-разработчик. Минимальная стоимость услуг специалиста в области машинного обучения оценивается в 40 000 руб./месяц, backend и frontend-разработчиков уровня Junior – в 40 000 руб./месяц, специалиста по рекламе – 35 000 руб./месяц [17]. С учетом страховых отчислений суммарные месячные затраты составят (40 000 + 40 000 + 40 000 + 35 000) * 1,302 = 201 810 руб.

Стоимость сервера для хранения данных составляет 110 182 рубля. Срок службы сервера составляет 5 лет и относится ко второй амортизационной группе. Срок полезного использования сервера устанавливается в пределах от 2 лет и одного месяца до 3 лет. Примем

среднее значение срока полезного использования – 2,5 года. Таким образом, ежегодные амортизационные отчисления составляют 44 072,8 руб. Услуги по установке и настройке сервера для хранения данных составляют 7500 руб [19].

Стоимость рекламы в поисковых системах Яндекс и Google одновременно составит 20 000 руб./мес [18].

Разработка программного обеспечения предполагает дистанционную занятость и не предусматривает аренду офисного помещения, а также подразумевает наличие персонального компьютера у разработчиков.

Произведем расчет себестоимости исходя из времени производства одного ПО равному 1 неделе. Тогда траты на содержание сотрудников составят 201 810 * 7 / 30 = 47 089 руб. Стоимость оборудования составит 44 072,8 * 7 / 365 = 845,23 руб. Стоимость услуги настройки сервера составит 7500 * 7 / 365 = 143,84 руб. Стоимость контекстной рекламы составит 20 000 * 7 / 30 = 4 666,67 руб.

Таблица 4 – Себестоимость ПО

Оборудование, тыс. руб.	0,84523
Заработная плата, тыс. руб.	47,089
Услуги сторонних организаций и сервисов, тыс. руб.	4,81051
Себестоимость единицы продукции, тыс. руб.	52,74474

3.7. Конкурентные преимущества создаваемого продукта

Для проведения конкурентного анализа не удалось найти аналогичных решений. Поэтому сравнение проведено с мануальным методом составления программы научных конференций. Результаты сравнения приведены в таблице 5.

Таблица 5 – Конкурентный анализ

Решение / Критерий	Разрабатываемое решение	Мануальное решение
Масштабируемость	+	+
Индивидуализация	+	+
Сокращение времени проведения анализа заявки участника	+	-
Оценка релевантности заявки участника	+	-
Уменьшение количества человек, участвующих в составлении плана мероприятия	+	-

3.7. Бизнес-модель проекта. Производственный план и план продаж

Для разработки бизнес-модели проекта воспользуемся моделью А. Остервальдера, представляющую собой схему из 9 блоков, описывающих различные бизнес-процессы проекта [14].

Таблица 6 – Бизнес-модель по А. Остервальдеру

<table>
<tr>
<td rowspan="2">Ключевые партнёры
Организаторы конференций, выставок, семинаров и т.д.</td>
<td>Ключевые виды деятельности
Разработка программного обеспечения;
Техническая поддержка.</td>
<td rowspan="2">Ценностные предложения
Оценка релевантности заявки на этапе регистрации;
Автоматическое формирование программы выступления участников.</td>
<td>Взаимоотношения с клиентами
Совместное создание;
Персональная техническая поддержка.</td>
<td rowspan="2">Потребительские сегменты
Руководители секций, руководители конференций, спонсоры мероприятий.</td>
</tr>
<tr>
<td>Ключевые ресурсы
Финансы (з/п сотрудникам);
Средства для содержания сервера</td>
<td>Каналы сбыта
Основной канал сбыта: информационный – участие в форумах, конференциях.</td>
</tr>
<tr>
<td colspan="3">Структура издержек
Постоянные: оплата за содержание сервера, заработная плата сотрудников</td>
<td colspan="2">Потоки поступления доходов
Продажа готового решения программного обеспечения.</td>
</tr>
</table>

технической поддержки на начальном этапе реализации ПО. Переменные: заработная плата сотрудников, когда заказов станет достаточно много.	

Длительность создания программного продукта для одного заказчика оценивается в одну неделю. При последовательном и равномерном выполнении заказов (каждую неделю один заказчик) и текущем кадровом составе предполагается совершить 52 продажи в первый год. Данный срок обусловлен наличием базового ПО, разработанного авторами в результате выполнения ВКР, что позволяет довольно быстро адаптировать продукт под нового заказчика.

Примем прибыль от продажи равной 20% в первый год использования, тогда из расчета себестоимости в 52 744,74 руб. планируемая стоимость ПО составляет 52 744,74 * 1,2 = 63 293,69 руб. Таким образом, выручка за продажу 52 ПО без учёта налога составит 52 * 63 293,69 = 3 291 271,88 руб., а прибыль составит 3 291 271,88 / 6 = 548 545,31 руб.

3.8. Стратегия продвижения продукта на рынок

Были выделены следующие способы продвижения продукта на рынок в течении года:

1. Прямые продажи – 1,92% – 1 клиент;
2. E-mail-маркетинг – 69,23% – 36 клиентов;
3. Работа в социальных сетях и соцмедиа (SMM) – 15,38% – 8 клиентов;
4. Реклама в поисковых системах (SEM — Search Engine Marketing, SEO + контекстная реклама) – 13,46% – 7 клиентов.

Доли планируемого количества привлеченных клиентов были определены исходя из статистики [16].

Так как предполагается равномерное поступление заказов, расходы на рекламу и рекламного агента также будут постоянными линейными.

4. Социальная ответственность

4.1. Введение

Объектом исследования данной ВКР являются данные о публикациях и авторах, извлеченные из наукометрической базы данных Scopus. Данная работа направлена на определение структур сотрудничества между исследователями в области «Экономики, Эконометрики и Финансов» (ECON) и выявлении учреждений, направленных на данную тематику. В результате исследования проводится анализ найденных кластеров, строится их визуальное представление, выделяются кластеры, содержащие самое большое количество учёных.

Так как работа производится непосредственно на персональном компьютере (ПК), то в данном разделе целесообразно рассмотреть вопросы анализа опасных и вредных факторов при работе с ПК, влияния этих факторов на окружающую среду и мероприятий по её защите.

Предметом исследования является рабочая зона офисного сотрудника, включая письменный стол, персональный компьютер, клавиатуру, компьютерную мышь и стул, а также помещение в котором эта рабочая зона находится.

4.2. Правовые и организационные вопросы обеспечения безопасности

Разработка программного обеспечения происходит за компьютерным столом. Рабочее место должно удовлетворять требованиям ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя» [20]. РД 153-34.0-03.298-2001 «Типовая конструкция по охране труда для пользователей персональными электронно-вычислительными машинами (ПЭВМ) в электроэнергетике» [22]. Требования к нормам

труда (продолжительность рабочего дня, перерывы в течение рабочего дня, перерывы на обед) регламентируются ТК РФ «Рабочее время» [26].

Выполнение требований на данном рабочем месте отражено ниже в таблице 7, согласно СанПиН 1.2.3685-21 и ГОСТ 12.2.032-78.

Таблица 7 – Требования к организации рабочего места при работе с ПЭВМ

Требование	Требуемое значение	Значение параметров помещения
Высота рабочей поверхности стола	Регулируемая высота(680-800мм) Нерегулируемая высота (725мм)	Нерегулируемая высота (700 мм)
Рабочий стул	Подъемно-поворотный, регулируемый по высоте и углу наклона спинки	Соответствует
Расположение монитора от глаз пользователя	600-700мм	Соответствует

4.3. Производственная безопасность.

Для идентификации потенциальных факторов был использован ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация». При работе с ПК пользователь подвергается воздействию опасных и вредных производственных факторов, представленных в таблице 8.

Таблица 8 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработка	Изготовление	Эксплуатация	
1. Отклонение параметров	+	+		СанПиН 1.2.3685-21 «Гигиенические нормативы и

микроклимата в помещении				требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [23]
2. Недостаточная освещенность рабочей зоны	+	+	+	СП 52.13330.2016 «Естественное и искусственное освещение» [25]
3. Повышенный уровень шума на рабочем месте	+	+		СН 2.2.4/ 2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки» [24]
4. Опасность поражения электрическим током	+	+	+	ГОСТ Р 58698-2019 «Защита от поражения электрическим током» [21]

4.4. Анализ опасных и вредных производственных факторов

4.4.1. Отклонение параметров микроклимата в помещении

Микроклимат производственных помещений – это климат внутренней среды помещений, который определяется действующими на организм человека сочетаниями температур воздуха и поверхностей, относительной влажности воздуха, скорости движения воздуха и интенсивности теплового излучения. Показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма.

Оптимальные микроклиматические при воздействии на человека в течение рабочей смены обеспечивают сохранение теплового состояния организма и не вызывают отклонений в состоянии здоровья. Допустимые микроклиматические условия могут приводить к незначительным дискомфортным тепловым ощущениям. Возможно, временное (в течение рабочей смены) снижение работоспособности, без нарушения здоровья.

Нормы оптимальных и допустимых показателей устанавливает СанПиН 1.2.3685-21. Он регулирует множество параметров, среди которых: температуру воздуха, температуру поверхностей конструкции, относительную влажность воздуха, скорость движения воздуха и интенсивность теплового облучения. Допустимые величины параметров микроклимата на рабочих местах в помещениях оцениваются в зависимости от категории работ по уровню энергозатрат организма. Работа, производимая сидя и сопровождающаяся незначительным физическим напряжением, относится к категории Iа – работа с интенсивностью энергозатрат до 139 Вт. Допустимые нормы микроклимата приведены в таблице 9.

Таблица 9 – Допустимые нормы микроклимата в рабочей зоне производственных помещений.

Период года	Категория работ по уровню энергопотребления, Вт	Температура воздуха °С		Температура воздуха поверхностей	Относительная влажность воздуха	Скорость движения воздуха	
		Диапазон ниже оптимальных величин	Диапазон выше минимальных величин			Для диапазона температур воздуха ниже оптимальных величин, не более	Для диапазона температур воздуха выше оптимальных величин, не более
Холодный	Iа (до 139)	20,0-21,9	24,1-25,0	19,0-26,0	15-75	0,1	0,1
Теплый	Iа (до 139)	21,0-28,0	25,1-28,0	20,0-29,0	15-75	0,1	0,2

В производственных помещениях, где допускаемые нормативные величины локального микроклимата поддерживать не представляется возможным, необходимо проводить мероприятия по защите работников от возможного перегревания и охлаждения. Это достигается разными способами: использование систем местного кондиционирования воздуха; регламентацией периодов работы в неблагоприятном локальном микроклимате и отдыха в помещении с микроклиматом,

нормализующим тепловое состояние; уменьшение длительности рабочей смены и др.

4.4.2. Недостаточная освещённость рабочей зоны

Свет является естественным условием жизни человека. Верно, спроектированное и выполненное освещение обуславливает высокую степень работоспособности, оказывает положительное психологическое воздействие на человека и содействует увеличению производительности труда. На рабочей поверхности обязаны отсутствовать резкие тени, которые создают неравномерное рассредоточивание поверхностей с различной яркостью в поле зрения, искажает размеры и формы объектов различия, в итоге увеличивается утомляемость персонала и понижается производительность труда.

Свет влияет на физиологическое состояние человека, правильно организованное освещение стимулирует протекание процессов высшей нервной деятельности и повышает работоспособность. При недостаточном освещении человек работает менее продуктивно, быстро устает, растет вероятность ошибочных действий, что может привести к травматизму.

Нормы оптимальных и допустимых показателей устанавливает СанПиН 1.2.3685-21.

Таблица 10 – Нормы оптимальных и допустимых значений освещенности.

Рабочая поверхность и плоскость нормирования	Естественное освещение		Совмещённое освещение		Искусственное освещение			
	КЕО, %		КЕО, %		Освещённость, лк		Объединённый показа	Коэффициент пульс
	При верхнем или	При боковом	При верхнем или	При боковом	При комбинированном освещении	При общем		

КЕО и освещённости и высота плоскости над полом, м	комбинированном освещении	освещении	комбинированном освещении	освещении	Всего	От общего	освещении	тель дискомфорта, UGR, не более	ации освещённости, Кп, %, не более
Г-0,8	3,0	1,0	1,8	0,6	400	200	300	21	15

К средствам нормализации освещенности производственных помещений рабочих мест относятся:

- источники света;

- осветительные приборы;

- световые проемы;

- светозащитные устройства.

4.4.3. Повышенный уровень шума на рабочем месте

Основной источник создаваемого шума в помещении – это другие электрические машины.

Повышенный уровень шума может привести к хронической бессоннице, сердечным заболеваниям, нарушениям слуха, повышению в организме гормонов стресса, снижению иммунитета, неврозам.

Может возникнуть шумовая болезнь, которая далеко не всегда поддаётся лечению.

Предельно допустимый уровень (ПДУ) шума – это степень фактора, который при ежедневной (кроме выходных дней) работе, но не более 40 часов в неделю в течение всего рабочего стажа, не должен вызывать заболеваний или же отклонений в состоянии здоровья и

самочувствия. Соблюдение ПДУ шума не исключает нарушения здоровья и самочувствия у сверхчувствительных лиц.

Допустимые значения уровня шума ограничены СанПиН 1.2.3685-21.

Таблица 11 – Допустимые значения уровня шума.

Для источников постоянного шума										Для источников непостоянного шума	
Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука L(A), дБА	Эквивалентные уровни звука, L(Аэкв.), дБА	Максимальные уровни звука, L(Амакс), дБА
31,5	63	125	250	500	1000	2000	4000	80000			
79	63	52	45	39	35	32	30	28	40	40	55

При значениях выше допустимого уровня необходимо предусмотреть средства коллективной защиты (СКЗ). Средства коллективной защиты:

- устранение причин шума или значительное его ослабление в источнике возникновения;

- изоляция источников шума от окружающей среды средствами звуко- и виброизоляции, звуко- и вибропоглощения;

- использование средств, снижающих шум и вибрацию на пути их передачи.

4.4.4. Опасность поражения электрическим током

Электробезопасность подразумевает под собой систему мероприятий, технических и организационных, направленных на защиту людей от опасного воздействия электрического тока,

статического электричества и электромагнитного поля. Значения вышеперечисленных факторов регулируются ГОСТ Р 58698-2019.

Таблица 12 – Пороги напряжения прикосновения для реагирования.

Характер регулирования	Пороги напряжения
Реакция испуга	2 В переменный ток
	8 В постоянный ток
Мышечная реакция	20 В переменный ток
	40 В постоянный ток

Меры предосторожности для основной защиты от повреждения электрическим током:

- использование защитных ограждений или оболочек;

- размещение опасных для жизни и здоровья человека участков электропроводов и приборов вне зоны досягаемости рукой;

- ограничение напряжения или питание должно осуществляться от безопасного источника питания;

- автоматическое отключение питания (защитное устройство, которое будет отключать систему, питающую электрическое оборудование или установку в случае замыкания).

Меры защиты:

- посредством системы безопасного сверхнизкого напряжения (БСНН) и защитного сверхнизкого напряжения (ЗСНН).

4.4.5. Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов на исследователя (работающего)

1. Перед началом работы следует убедиться в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки и провода электропитания, в

отсутствии видимых повреждений аппаратуры и рабочей мебели, в отсутствии повреждений и наличии заземления приэкранного фильтра.

2. При отклонении от нормы предоставить обогреватель, вентилятор или увлажнитель воздуха в зависимости от требуемых условий работы.

3. При отклонении от нормы предоставить дополнительные источники света (например, настольные лампы, точечные светильники и т.п.) зависимости от требуемых условий работы.

4. Монитор компьютера служит источником ЭМП – вредного фактора, который отрицательно влияет на здоровье работника при продолжительной непрерывной работе и приводит к снижению работоспособности. Поэтому во избежание негативного влияния на здоровье необходимо делать перерывы при работе с ЭВМ и проводить специализированные комплексы упражнений для глаз.

4.5. Экологическая безопасность

При выполнении ВКР основными отходами являются: бумага и люминесцентные лампы. Для утилизации бумаги существуют пункты переработки бумаги или специальные мусорные контейнеры. Так же существуют пункты приема для утилизации перегоревших люминесцентных ламп. Объект исследования не оказывает влияния на окружающую среду и не наносит ей ущерб.

4.6. Безопасность в чрезвычайных ситуациях

4.6.1. Затопление

Главная опасность при затоплении помещения, в котором находятся ПК – это способность воды проводить электрический ток, что означает возможность поражения электрическим током человека, находящегося в таком помещении. Ток проводят не сами молекулы

воды, а различные примеси, содержащиеся в ней, такие как ионы различных минеральных солей, которые в достаточных количествах содержат сточные воды.

Затопление может иметь характер техногенной чрезвычайной ситуации, когда возникает по причине наличия сильной изношенности водопровода, свищей, негерметичных соединений водопроводных систем или в следствии аварийной ситуации. Также затопление может являться чрезвычайной ситуацией природного характера, в случаях, когда оно возникает в результате наводнений, паводков и т.д.

4.6.2. Землетрясение

Землетрясение – это подземные толчки и колебания земной поверхности из-за внезапных смещений и разрывов в земной коре или верхней мантии Земли, которые передаются на большие расстояния. Данная чрезвычайная ситуация имеет природный характер, может привести к выходу из строя коммуникаций и энергетических объектов, разрушению зданий, появлению трещин в грунте, возникновению пожаров, значительным людским потерям.

4.6.3. Короткое замыкание

Работа с персональными компьютерами подразумевает постоянное использование электрического тока. При несоблюдении правил электробезопасности возможно возникновение короткого замыкания проводки – резкое и многократное возрастание силы тока, протекающего в цепи, что приводит к значительному тепловыделению, расплавлению электрических проводов с последующим возникновением возгорания. Причиной короткого замыкания является нарушение изоляции и соединения токопроводящих частей электроустановок друг с другом или с заземлёнными поверхностями непосредственно или через токопроводящий материал. К нарушениям изоляции могут привести

перенапряжение, прямые удары молнии, внешние механические повреждения, старение и износ самой изоляции, в том числе возникшие из-за неудовлетворительного ухода.

Если человек находится рядом с участком цепи в котором произошло короткое замыкание, он может получить ожоги, в том числе смертельные. Компьютеры, подключённые в цепь, в которой произошло короткое замыкание могут выйти из строя. Для минимизации перечисленных негативных последствий короткого замыкания следует использовать кабель не распространяющий горение, или помещать кабель в стальные трубы с определённой толщиной стенки, которая не прожжётся при возникновении короткого замыкания.

4.6.4. Пожар

Пожар представляет большую опасность и наносит огромный ущерб, поскольку грозит уничтожением приборов, компьютеров, инструментов и комплектов документов, представляющих значительную ценность. Кроме того, пожар характеризуется опасностью для жизни человека. Возникновение пожара в комнате может быть обусловлено следующими факторами: короткое замыкание или перегрев ПК.

Поэтому во избежание пожаров проводится пожарная профилактика – комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, на предотвращение пожара, ограничение его распространения, а также на создание условий для успешного тушения пожара. Успех борьбы с пожаром во многом зависит от его своевременного обнаружения и быстрого принятия мер по его ограничению и ликвидации. При появлении возгорания необходимо сообщить в службу пожарной охраны адрес и место возникновения пожара.

4.7. Выводы по разделу

Рабочее помещение, где была разработана ВКР, соблюдены все нормы безопасности. Действие вредных и опасных факторов сведено к минимуму. Само помещение и рабочее место удовлетворяет всем требованиям.

Действие вредных и опасных факторов сведено к минимуму, т.е. микроклимат, освещение и электробезопасность соответствуют требованиям, предъявленным в соответствующих нормативных документах. Не стоит забывать, что монитор компьютера служит источником вредного фактора и отрицательно влияет на здоровье офисного сотрудника. Во избежание этого, нужно делать перерывы в работе и проводить специальные комплексы упражнений для разминки тела.

5. Заключение

В результате работы был проведен сбор и предварительная обработка корпуса документов: 6 сборников трудов международной конференции студентов, аспирантов и молодых ученых «Перспективы развития фундаментальных наук» за 2016 – 2021 гг. (приложение А). Автоматизированная обработка документов включала извлечение текстовой информации из pdf файлов Получена выборка из 2879 аннотаций, элементы выборки подготовлены для машинного обучения: удалены переносы строк, удалены знаки пунктуации, удалены переносы строк, удалены служебные слова, тексты были переведены на английский язык для однородности.

С использованием машинных методов обучения: 1) мера важности слова (TF-IDF), 2) векторное представление документа (doc2vec) для элементов выборки сгенерированы числовые признаки для текстов. В результате работы для сбалансированной и несбалансированной выборок оценены параметры для следующих классификаторов: к-ближайших соседей, логистическая регрессия, метод опорных векторов, однослойный персептрон. Достигнутая точность 81,98% методом опорных векторов. Классифицированные данные были отсортированы на основе косинусного расстояния для создания плана выступления.

Список используемых источников

1. XIX Международная конференция студентов, аспирантов и молодых ученых «Перспективы развития фундаментальных наук» [Электронный ресурс]. – Режим доступа: https://conf-prfn.org/archive, свободный (Дата обращения: 05.05.2022).
2. Воронцов К.В. Линейные методы классификации и регрессии: метод стохастического градиента. Курс лекций. – Москва: МФТИ, 2009.
3. Воронцов К.В. Вычислительные методы обучения по прецедентам. Курс лекций. – Москва: МФТИ, 2007.
4. Воронцов К.В. Машинное обучение. Курс лекций. – Москва: МФТИ, 2021.
5. Воронцов К.В. Линейные методы классификации. Курс лекций. – Москва: МФТИ, 2013
6. Логистическая регрессия и ROC-анализ — математический аппарат [Электронный ресурс]. – Режим доступа: https://loginom.ru/blog/logistic-regression-roc-auc, свободный (Дата обращения: 05.05.2022).
7. Лекции по метрическим алгоритмам классификации [Электронный ресурс]. – Режим доступа: http://www.ccas.ru/voron/download/MetricAlgs.pdf, свободный (Дата обращения: 05.05.2022).
8. Tom Fawcett. An introduction to ROC analysis // Pattern Recognition Letters. . 8. 27. С. 861-874.
9. TensorFlow – однослойный персептрон [Электронный ресурс]. – Режим доступа: https://coderlessons.com/tutorials/mashinnoe-obuchenie/vyuchit-tensorflow/tensorflow-odnosloinyi-perseptron, свободный (Дата обращения: 05.05.2022).

10. Q. Le, T. Mikolov. Distributed Representations of Sentences and Documents [Электронный ресурс]. – Режим доступа: https://cs.stanford.edu/~quocle/paragraph_vector.pdf, свободный (Дата обращения: 05.05.2022).
11. Открытый каталог научных конференций, выставок и семинаров [Электронный ресурс]. – Режим доступа: https://konferencii.ru/, свободный (Дата обращения 02.06.2022).
12. Мухаметгалиева К. А. Патентование программного обеспечения / К. А. Мухаметгалиева // Правовая защита, экономика и управление интеллектуальной собственностью: материалы всероссийской научно-практической конференции, Екатеринбург, 21 апреля 2015 г. — Екатеринбург: [УрФУ], 2015. — Т. 1 — С. 84-91.
13. Event.ru [Электронный ресурс]. – Режим доступа: https://event.ru/trips/rossiyskiy-event-ryinok-nastoyashhee-i-budushhee-v-tsifrah/, свободный (Дата обращения 06.06.2022).
14. Vc.ru [Электронный ресурс]. – Режим доступа: https://vc.ru/s/productstar/135102-biznes-model-ostervaldera-chto-eto-takoe#:~:text=Бизнес-модель%20Остервальдера%20(Business%20Model,Александр%20Остервальдер%20и%20Ив%20Пинье., свободный (Дата обращения 06.06.2022).
15. Т. Дьякова. Шесть главных трендов современной науки– исследование НИУ ВШЭ [Электронный ресурс]. – Режим доступа: https://trends.rbc.ru/trends/innovation/61e49a2b9a7947633a0be8e2 свободный (Дата обращения 06.06.2022).
16. Д.Андреева. статистика по e-mail маркетингу [Электронный ресурс]. –

Режим доступа: https://sendpulse.com/ru/blog/statistics-for-email-marketer, свободный (Дата обращения 06.06.2022).

17. hh [Электронный ресурс]. – Режим доступа: https://tomsk.hh.ru/?hhtmFrom=vacancy_search_list, свободный (Дата обращения: 06.06.2022).
18. SEOintellect [Электронный ресурс]. – Режим доступа: https://seointellect.ru/context/stoimost, свободный (Дата обращения: 06.06.2022).
19. ServerMashines [Электронный ресурс]. – Режим доступа: https://servermachines.ru/catalog/servernoe-oborudovanie/dell/servery-i-shassi-dell-emc/cervery-dell-emc-v-korpuse-tower/6097/, свободный (Дата обращения: 06.06.2022).
20. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования от 01.03.1986: дата введения 01.01.1979.
21. ГОСТ Р 58698-2019 (МЭК 61140:2016) Защита от поражения электрическим током. Общие положения для электроустановок и электрооборудования.
22. РД 153-34.0-03.298-2001 «Типовая конструкция по охране труда для пользователей персональными электронно-вычислительными машинами (ПЭВМ) в электроэнергетике».
23. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и безвредности для человека факторов среды обитания».
24. СН 2.2.4/ 2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории милой застройки».
25. СП 52.13330.2016 «Естественное и искусственное освещение».
26. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018).

Приложение А

```
import io
from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from pdfminer.pdfpage import PDFPage
def convert_pdf_to_txt(path, char_margin, line_margin):
    resource_manager = PDFResourceManager()
    retstr = io.StringIO()
    codec = 'utf-8'
    laparams = LAParams(char_margin = char_margin, line_margin = line_margin)
    device = TextConverter(resource_manager, retstr, codec = codec, laparams = laparams)
    fp = open(path, 'rb')
    interpreter = PDFPageInterpreter(resource_manager, device)
    password = ""
    maxpages = 0
    caching = True
    check_extractable = False
    pagenos = set()
    j = 1
    for page in PDFPage.get_pages(fp, pagenos, maxpages = maxpages,
                                  password = password,
                                  caching = caching,
                                  check_extractable = check_extractable):
        try:
            interpreter.process_page(page)
        except:
            print(j)
            print(page)
        j += 1
    fp.close()
    device.close()
    text = retstr.getvalue()
    retstr.close()
    return text
```

Приложение Б

```
import sys
from PyQt5.QtWidgets import (QWidget, QToolTip,
                             QPushButton, QApplication, QTextEdit, QFileDialog, QMessageBox)
from PyQt5.QtGui import QFont, QIcon
from PyQt5.QtCore import QCoreApplication
from classification import classification
import os
from scipy.spatial import distance
from sklearn.feature_extraction.text import TfidfVectorizer

class Example(QWidget):

    def __init__(self):
        super().__init__()

        self.initUI()

    def button1(self):
        self.text = self.textbox.toPlainText()
        self.textbox.clear()
        classes = ['Физика', 'Химия', 'Математика', 'Биология и фундаментальная медицина', 'Экономика и управление', 'Строительство и архитектура', 'IT технологии и электроника']
        path_to_save_sgd = str(os.getcwd()) + '\\result_sgd.txt'
        path_to_save_knb = str(os.getcwd()) + '\\result_knb.txt'
        path_to_save_lr = str(os.getcwd()) + '\\result_lr.txt'
        predicted_sgd = sgd_ppl_clf.predict([self.text])
        predicted_knb = knb_ppl_clf.predict([self.text])
        predicted_lr = lr_ppl_clf.predict([self.text])
        f = open(path_to_save_sgd, 'w', encoding = 'utf-8')
        f.write(str(classes[predicted_sgd[0]-1]) + '\t' + self.text)
        f.close()

        f = open(path_to_save_knb, 'w', encoding = 'utf-8')
```

```
        f.write(str(classes[predicted_knb[0]-1]) + '\t' + self.text)
        f.close()

        f = open(path_to_save_lr, 'w', encoding = 'utf-8')
        f.write(str(classes[predicted_lr[0]-1]) + '\t' + self.text)
        f.close()

        msgBox = QMessageBox()
        msgBox.setIcon(QMessageBox.Information)
        msgBox.setStyleSheet("QLabel{min-width: 60px;}")
        msgBox.setText("Done")
        msgBox.setWindowTitle("Done")
        msgBox.setStandardButtons(QMessageBox.Ok)
        msgBox.exec()

    def button2(self):
        path = QFileDialog.getOpenFileName(self, 'Open File', './', 'Text Files (*.txt)')
        f = open(path[0], 'r', encoding = 'utf-8').readlines()
        predicted_sgd = sgd_ppl_clf.predict(f)
        predicted_knb = knb_ppl_clf.predict(f)
        predicted_lr = lr_ppl_clf.predict(f)

        y = predicted_sgd
        f_by_classes = [[], [], [], [], [], [], []]
        for i in range(len(y)):
            f_by_classes[y[i] - 1].append(f[i])

        vectorizer = TfidfVectorizer()
        for i in range(len(f_by_classes)):
            cur = vectorizer.fit_transform(f_by_classes[i]).toarray()
            for k in range(len(f_by_classes[i]) - 2):
                dist = []
                for j in range(k + 1, len(f_by_classes[i])):
                    dist.append(distance.cosine(cur[k], cur[j]))
```

```
            pos = dist.index(min(dist)) + 1
            f_by_classes[i][pos], f_by_classes[i][k + 1] = f_by_classes[i][k + 1], f_by_classes[i][pos]
            cur[pos], cur[k + 1] = cur[k + 1], cur[pos]

    y = predicted_knb
    f_by_classes_1 = [[], [], [], [], [], [], []]
    for i in range(len(y)):
        f_by_classes_1[y[i] - 1].append(f[i])

    vectorizer = TfidfVectorizer()
    for i in range(len(f_by_classes_1)):
        cur = vectorizer.fit_transform(f_by_classes_1[i]).toarray()
        for k in range(len(f_by_classes_1[i]) - 2):
            dist = []
            for j in range(k + 1, len(f_by_classes_1[i])):
                dist.append(distance.cosine(cur[k], cur[j]))

            pos = dist.index(min(dist)) + 1
            f_by_classes_1[i][pos], f_by_classes_1[i][k + 1] = f_by_classes_1[i][k + 1], f_by_classes_1[i][pos]
            cur[pos], cur[k + 1] = cur[k + 1], cur[pos]

    y = predicted_lr
    f_by_classes_2 = [[], [], [], [], [], [], []]
    for i in range(len(y)):
        f_by_classes_2[y[i] - 1].append(f[i])

    vectorizer = TfidfVectorizer()
    for i in range(len(f_by_classes_2)):
        cur = vectorizer.fit_transform(f_by_classes_2[i]).toarray()
        for k in range(len(f_by_classes_2[i]) - 2):
            dist = []
            for j in range(k + 1, len(f_by_classes_2[i])):
                dist.append(distance.cosine(cur[k], cur[j]))
```

```
            pos = dist.index(min(dist)) + 1
            f_by_classes_2[i][pos], f_by_classes_2[i][k + 1] = f_by_classes_2[i][k + 1], f_by_classes_2[i][pos]
            cur[pos], cur[k + 1] = cur[k + 1], cur[pos]

    path_to_save_sgd = str(os.getcwd()) + '\\result_sgd.txt'
    path_to_save_knb = str(os.getcwd()) + '\\result_knb.txt'
    path_to_save_lr = str(os.getcwd()) + '\\result_lr.txt'
    classes = ['Физика', 'Химия', 'Математика', 'Биология и фундаментальная медицина', 'Экономика и управление', 'Строительство и архитектура', 'IT технологии и электроника']
    f2 = open(path_to_save_sgd, 'w', encoding = 'utf-8')
    for i in range(len(f_by_classes)):
        for j in range(len(f_by_classes[i])):
            f2.write(str(classes[i]) + '\t' + f_by_classes[i][j])

    f2.close()

    f2 = open(path_to_save_knb, 'w', encoding = 'utf-8')
    for i in range(len(f_by_classes_1)):
        for j in range(len(f_by_classes_1[i])):
            f2.write(str(classes[i]) + '\t' + f_by_classes_1[i][j])

    f2.close()
    f2 = open(path_to_save_lr, 'w', encoding = 'utf-8')
    for i in range(len(f_by_classes_2)):
        for j in range(len(f_by_classes_2[i])):
            f2.write(str(classes[i]) + '\t' + f_by_classes_2[i][j])
    f2.close()
    msgBox = QMessageBox()
    msgBox.setIcon(QMessageBox.Information)
    msgBox.setStyleSheet("QLabel{min-width: 60px;}")
    msgBox.setText("Done")
    msgBox.setWindowTitle("Done")
```

```
        msgBox.setStandardButtons(QMessageBox.Ok)
        msgBox.exec()
    def initUI(self):
        QToolTip.setFont(QFont('SansSerif', 10))
        self.setToolTip('This is a <b>QWidget</b> widget')
        self.textbox = QTextEdit(self)
        self.textbox.setPlaceholderText('Введите аннотацию...')
        self.textbox.resize(500, 100)
        self.textbox.move(50, 20)
        self.btn1 = QPushButton('Go', self)
        self.btn1.setToolTip('This is a <b>QPushButton</b> widget')
        self.btn1.resize(80, 20)
        self.btn1.move(260, 130)
        self.btn1.clicked.connect(self.button1)
        self.btn2 = QPushButton('Attach file', self)
        self.btn2.setToolTip('This is a <b>QPushButton</b> widget')
        self.btn2.resize(160, 40)
        self.btn2.move(220, 300)
        self.btn2.clicked.connect(self.button2)
        self.setGeometry(660, 240, 600, 500)
        self.setWindowTitle('my_program')
        self.setWindowIcon(QIcon('icon.png'))
        self.show()
if __name__ == '__main__':
    sgd_ppl_clf, knb_ppl_clf, lr_ppl_clf = classification(str(os.getcwd()) + '\\x_test.txt',
str(os.getcwd()) + '\\y_test.txt')
    app = QApplication(sys.argv)
    ex = Example()
    sys.exit(app.exec_())
```