



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное
учреждение высшего образования

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Информационных технологий и систем безопасности

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(Специалитет)

На тему «**Построение психологического портрета по технологии
TextMining**»

Исполнитель

Кравченко Егор Владимирович

(фамилия, имя, отчество)

Руководитель

Доктор технических наук

(ученая степень, ученое звание)

Лепешкин Олег Михайлович

(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой

(Подпись)

Доктор технических наук, доцент

(ученая степень, ученое звание)

Лепешкин Олег Михайлович

(фамилия, имя, отчество)

« ____ » _____ 20 ____ г.

Санкт-Петербург 2026 год

МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Информационных технологий и систем безопасности
«УТВЕРЖДАЮ»
Заведующий кафедрой

(подпись) Лепешкин Олег Михайлович
(фамилия, имя, отчество)

«__» _____ 20__ года

Задание

на выпускную квалификационную работу

студенту Кравченко Егору Владимировичу

(фамилия, имя, отчество)

1. Тема «Построение психологического портрета по технологии TextMining»
закреплена приказом ректора Университета от «09» 09 2020 года, № 684-с
2. Срок сдачи законченной работы «__» _____ 20__ года
3. Исходные данные к выпускной квалификационной работе:

Среда разработки: python

Конфигурационные параметры нейронных сетей : архитектура (8 входных нейронов, 4 выходных), инициализированные весовые коэффициенты (weights, bias), параметры обучения (количество эпох, скорость обучения) и типы активационных функций (сигмоида для скрытых преобразований, softmax для классификации).

Входная информация — это исходные данные для анализа, поступающие в систему через пользовательский интерфейс. Основным источником является текстовое поле (ScrolledText) в GUI приложения

4. Перечень вопросов, подлежащих разработке (краткое содержание работы):
Введение. Актуальность темы, цели и задачи ВКР

Глава 1. Психологический портрет личности

(наименование главы)

Глава 2. Технология Text Mining

(наименование главы)

Глава 3. Разработка программного обеспечения для создания психологического портрета личности

(наименование главы)

Заключение. Выводы по работе в целом. Оценка степени решения поставленных задач. Практические рекомендации.

5. Перечень материалов, представляемых к защите:

–Пояснительная записка;

- Листинг программы

6. Консультанты по работе

6.1. _____

7. Дата выдачи задания: «09» 09 2020 года

Руководитель выпускной квалификационной работы

и.о. зав. кафедры, доктор технических наук, доцент Лепешкин Олег Михайлович

(должность, ученая степень, ученое звание, фамилия, имя, отчество)

(подпись)

Задание принял к исполнению «09» 09 2020 года

Студент Кравченко Егор Владимирович

РЕФЕРАТ

Дипломная работа: 93 с., 9 рис., 1 табл., 1 приложения,
34 источников литературы.

СИСТЕМА УПРАВЛЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТЬЮ,
СТАНДАРТИЗАЦИЯ В ОБЛАСТИ СУИБ, ИНФОРМАЦИОННАЯ
БЕЗОПАСНОСТЬ, ПОЛИТИКА ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ, ОБРАБОТКА РИСКОВ.

Объект исследования: Текстовые данные, технология Text Mining.

Предмет исследования: Использование машинного обучения для построения психологического портрета личности

Цель работы: разработка программного обеспечения для построения психологического портрета личности по технологии TextMining.

В дипломной работе проводится анализ технологии получения информации из неструктурированных текстовых данных путем их преобразования в пригодный для дальнейшей работы набор структурированных данных, представленных в удобном для машинной обработки виде.

Разработана программа для анализа психологического портрета личности, являющаяся комплексным решением, включающим в себя технологии искусственного интеллекта (в частности, искусственные нейронные сети) и методы обработки естественного языка.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. ПСИХОЛОГИЧЕСКИЙ ПОРТРЕТ ЛИЧНОСТИ	7
1.1 Понятие портрета личности	7
1.2 Построение портрета личности	19
ГЛАВА 2. ТЕХНОЛОГИЯ TEXT MINING	23
2.1 Основные понятия и терминология Text Mining	23
2.2 Задачи, решаемые Text Mining	33
2.3 Построение психологического портрета личности с помощью технологии TextMining.....	34
ГЛАВА 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СОЗДАНИЯ ПСИХОЛОГИЧЕСКОГО ПОРТРЕТА ЛИЧНОСТИ.....	49
3.1 Характеристика нормативно-справочной, входной, оперативной и результатной информации	49
3.2 Разработка общего алгоритма программного продукта	53
3.3. Выбор и обоснование программной реализации.....	60
3.4 Создание интерфейса.....	62
3.5 Тестирование программы.....	66
ЗАКЛЮЧЕНИЕ	70
СПИСОК ЛИТЕРАТУРЫ.....	72
ПРИЛОЖЕНИЕ 1	76

ВВЕДЕНИЕ

В современных реалиях наблюдается устойчивый рост популярности различных тестов по всему миру. Психодиагностика представляет собой метод психологического исследования, который обычно применяется для проверки гипотез относительно взаимосвязей среди различных психологических особенностей. Когда удаётся выявить характеристики у достаточного числа участников, наступает момент, когда с помощью соответствующих математических процедур можно установить их взаимосвязь. Для достижения этих целей применяются психодиагностические методики, которые помогают определить и оценить индивидуальные черты.

Из-за огромного потока информации ручная обработка и систематизация данных становятся неэффективными и дорогими. Поэтому существует необходимость либо в полной автоматизации этого процесса, либо в использовании человеческого ресурса лишь на тех этапах, когда нужные данные уже были отобраны автоматически. Для автоматизации извлечения информации из текстов используют интеллектуальный анализ текста, также известный как *text mining*.

Таким образом, для формирования психологического профиля будет удобно применять интеллектуальный анализ данных, полученных в результате психологических исследований конкретной личности. Цель данной работы заключается в исследовании необходимых материалов для приобретения навыков создания детального психологического портрета и последующего интеллектуального анализа собранной информации с использованием технологий *Text Mining*.

В эпоху цифровой трансформации объемы генерируемых текстовых данных растут экспоненциально. Эти данные, часто неструктурированные или слабо структурированные, содержат ценную информацию, способную стимулировать инновации, улучшить процессы и повысить эффективность. Анализ таких данных на естественных языках становится ключевым инструментом для извлечения знаний.

Одним из основных методов анализа является обработка естественного языка (NLP), область искусственного интеллекта, занимающаяся пониманием и генерацией человеческого языка. NLP позволяет компьютерам интерпретировать смысл текста, выявлять закономерности и извлекать полезные данные.

Алгоритмы машинного обучения играют важную роль в анализе текстовых данных. Методы классификации, кластеризации и тематического моделирования помогают организовать и структурировать информацию. Классификация позволяет определить категорию текста (например, положительный отзыв или новостная статья), кластеризация объединяет похожие тексты в группы, а тематическое моделирование выявляет ключевые темы и тренды.

Применение анализа текстовых данных охватывает широкий спектр областей. В маркетинге это позволяет анализировать отзывы клиентов, выявлять потребности и оптимизировать рекламные кампании. В финансах – выявлять мошеннические операции и оценивать кредитные риски. В здравоохранении – анализировать медицинские записи для улучшения диагностики и лечения.

Однако, работа с неструктурированными данными представляет собой вызов. Требуется предварительная обработка текста, включающая удаление стоп-слов, приведение слов к нормальной форме и токенизацию. Выбор правильных алгоритмов и параметров имеет решающее значение для достижения высокой точности и эффективности анализа.

В заключение, анализ неструктурированных текстовых данных на естественных языках открывает огромные возможности для извлечения знаний и принятия обоснованных решений. Развитие NLP и машинного обучения продолжает расширять границы возможного, делая анализ текста еще более мощным и доступным инструментом.

Частотная функция, или функция распределения частот, является краеугольным камнем статистического анализа, предоставляя возможность количественно оценить, как часто встречаются различные значения переменной в заданном наборе данных. В основе своей, частотная функция представляет

собой табличное или графическое представление, показывающее количество (или долю) наблюдений, попадающих в определенный интервал или категорию.

На практике, частотная функция может быть построена различными способами. Для дискретных данных, таких как количество бракованных изделий в партии, она принимает форму таблицы, перечисляющей каждое возможное значение и соответствующую ему частоту. Для непрерывных данных, таких как рост людей, данные обычно группируются в интервалы, а затем строится гистограмма, где высота столбцов соответствует частоте в каждом интервале.

Независимо от конкретной формы, частотная функция является мощным инструментом для визуализации и интерпретации данных, позволяющим исследователям принимать обоснованные решения в самых разных областях, от маркетинга до медицины и инженерии. Она помогает увидеть скрытые закономерности и понять, как данные распределены, что является ключом к успешному анализу и прогнозированию.

В отличие от простого поиска по ключевым словам, Text Mining идет дальше поверхностного анализа, позволяя выявлять скрытые взаимосвязи, контекст и семантический смысл текста. Этот процесс включает в себя несколько ключевых этапов: предварительную обработку текста, включающую удаление стоп-слов, приведение слов к нормальной форме и токенизацию; извлечение признаков, таких как частота слов, наличие определенных фраз или эмоциональная окраска; и, наконец, применение алгоритмов машинного обучения для классификации, кластеризации, тематического моделирования и других задач.

Применение Text Mining охватывает широкий спектр областей. В бизнесе эта технология используется для анализа отзывов клиентов, выявления потребностей рынка и мониторинга бренда. В медицине она помогает анализировать медицинские записи для улучшения диагностики и лечения заболеваний. В сфере безопасности Text Mining используется для выявления террористических угроз и мошеннических операций.

Однако освоение Text Mining требует не только технических знаний, но и понимания предметной области. Успешное применение этой технологии зависит от правильного выбора методов анализа, настройки параметров и интерпретации полученных результатов. В будущем, с развитием искусственного интеллекта, Text Mining станет еще более мощным инструментом для извлечения знаний и принятия обоснованных решений на основе текстовых данных.

Тема работы: «Построение психологического портрета по технологии TextMining».

Целью данной работы является разработка программного обеспечения для построения психологического портрета личности на основе TextMining.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить предметную область: исследовать технологию TextMining и оценить возможность построения психологического портрета с её применением;
2. Разработать алгоритм;
3. Выбрать среду программирования;
4. Создать программное обеспечение;
5. Провести тестирование разработанного ПО;
6. Сформулировать выводы.

ГЛАВА 1. ПСИХОЛОГИЧЕСКИЙ ПОРТРЕТ ЛИЧНОСТИ

1.1 Понятие портрета личности

Психологический портрет личности представляет собой комплексную характеристику внутреннего мира человека, включающую совокупность индивидуальных особенностей восприятия, эмоциональных реакций, мотиваций, поведения и взаимодействия с окружающей средой. Это понятие возникло на стыке психологии, социологии и философии, стремясь отразить глубинные структуры сознания и подсознания индивида, проявляющиеся в конкретных жизненных ситуациях.

История изучения психологического портрета уходит корнями в античность, когда философы древности пытались осмыслить природу человеческой души и её проявления в поступках и мыслях. Платон, Аристотель и Эпикур закладывали основы понимания индивидуальности и характерологических черт, влияющих на поведение и развитие личности. Позднее психоанализ Фрейда и Юнга привнёс идею бессознательного, которое играет ключевую роль в формировании психики и определяет многие особенности поведения человека.

Современные исследования подчёркивают важность интеграции различных подходов для составления целостного представления о личности. Например, гуманистическая психология Карла Роджерса фокусируется на самореализации и внутреннем потенциале каждого человека, тогда как когнитивная психология исследует механизмы обработки информации и принятия решений. Эти подходы позволяют учитывать широкий спектр факторов, формирующих личность, включая генетические предрасположенности, воспитание, культурный контекст и жизненный опыт.

Создание психологического портрета начинается с анализа биографических данных, позволяющих выявить ключевые события, повлиявшие на формирование личности. Важную роль играют также личностные опросники и тесты, направленные на выявление определённых качеств и склонностей. Одним из наиболее известных инструментов является Миннесотский

многофазный личностный опросник (ММРІ), позволяющий оценить степень выраженности тревожности, депрессии, интроверсии-экстраверсии и других характеристик.

Однако важно понимать, что психологический портрет не является статичным образованием. Под влиянием социальных изменений, личного опыта и новых обстоятельств черты характера и модели поведения могут трансформироваться. Поэтому создание точного и полного описания требует постоянного обновления и дополнения новыми данными.

Особое внимание уделяется изучению влияния культурных норм и традиций на формирование личности. Культурные различия определяют восприятие мира, нормы морали и этики, что находит отражение в поведении и мышлении людей разных народов. Так, восточная философия ориентирована на коллективизм и гармонию с окружающим миром, в то время как западная культура подчеркивает индивидуализм и стремление к успеху и достижениям.

Кроме того, важную роль в создании психологического портрета играют социальные роли и статусные позиции, занимаемые человеком в обществе. Социальные ожидания и требования оказывают значительное влияние на самооценку и самоидентификацию личности, определяя её цели и приоритеты. Люди стремятся соответствовать принятым нормам и стандартам, что отражается в их действиях и решениях.

Ещё одним аспектом формирования психологического портрета являются межличностные отношения. Близкое окружение оказывает огромное воздействие на процесс социализации и становления личности. Семья, друзья, коллеги формируют систему ценностей и убеждений, которыми руководствуется человек в повседневной жизни. Конфликты и кризисы в отношениях часто становятся катализаторами внутренних перемен, способствуя развитию или деградации личности.

Таким образом, психологический портрет личности представляет собой динамическое образование, отражающее взаимодействие множества факторов, влияющих на развитие и функционирование человека. Создание подробного и

глубокого описания требует комплексного подхода, учитывающего индивидуальные особенности, социальное окружение и культурный контекст. Это позволяет лучше понять внутренние мотивы и потребности человека, что способствует эффективному взаимодействию и решению возникающих проблем.

Важно отметить, что каждый человек уникален и неповторим, поэтому универсальных шаблонов для построения психологического портрета не существует. Каждый случай требует индивидуального подхода и тщательного анализа всех доступных данных. Только таким образом можно достичь объективного и всестороннего понимания личности, необходимого для успешной профессиональной деятельности, психологической помощи и общего благополучия человека.

Формирование психологического портрета включает ряд этапов, начиная с сбора первичной информации и заканчивая интерпретацией полученных результатов. Первым этапом является сбор анамнеза, включающего семейную историю, образовательный уровень, профессиональный опыт и личные предпочтения. Далее проводится диагностика с использованием специальных методик, направленных на выявление особенностей темперамента, уровня интеллекта, мотивации и эмоциональной устойчивости. Полученные данные интегрируются в единый образ, который отражает специфику каждой отдельной личности.

При составлении психологического портрета учитываются как позитивные, так и негативные стороны личности. Важно избегать стереотипов и предвзятых оценок, стремиться к максимальной объективности и точности. Необходимо помнить, что даже незначительные детали могут иметь важное значение для понимания сути личности и её потенциальных возможностей.

Например, изучение детских воспоминаний помогает раскрыть скрытые травмы и переживания, оказавшие существенное влияние на дальнейшее развитие. Анализ семейных взаимоотношений выявляет характерные паттерны поведения, усвоенные в раннем возрасте. Изучение профессиональных

достижений даёт представление о целях и стремлениях личности, а также о степени удовлетворённости своей жизнью.

Психологические портреты широко применяются в различных сферах деятельности, таких как подбор персонала, консультирование, коучинг и терапия. Они помогают определить совместимость сотрудников, спрогнозировать успешность адаптации новичков, разработать стратегии развития карьеры и разрешить конфликты внутри коллектива. Психологи используют полученные знания для разработки эффективных методов коррекции нежелательных поведенческих моделей и улучшения качества жизни клиентов.

Таким образом, психологический портрет личности является важным инструментом познания человеческого существа, необходимым для успешного функционирования в современном мире. Его значимость возрастает по мере усложнения социальной среды и увеличения требований к профессиональному росту и личной эффективности. Умение составлять точные и глубокие психологические портреты становится залогом успеха в любых областях деятельности, будь то бизнес, наука или искусство.

Психология изучает личность как сложную структуру, состоящую из взаимосвязанных элементов. Одной из ключевых составляющих является темперамент, представляющий собой устойчивые индивидуально-психологические характеристики, обусловленные биологическими факторами. Темперамент проявляется в особенностях нервной системы, таких как скорость реакции, чувствительность и устойчивость к стрессовым ситуациям. Выделяют четыре основных типа темперамента: холерик, меланхолик, флегматик и сангвиник. Холерики отличаются энергичностью и вспыльчивостью, меланхолики склонны к депрессивным состояниям и повышенной чувствительности, флегматики характеризуются спокойствием и уравновешенностью, а сангвиники обладают жизнерадостностью и общительностью.

Другой важной составляющей личности является характер, представляющий собой совокупность приобретённых черт, сформировавшихся в процессе воспитания и социального окружения. Характер определяется системой ценностных ориентаций, моральными принципами и привычными способами реагирования на внешние стимулы. Он формируется под воздействием семейной атмосферы, школьного образования, культурного контекста и личных переживаний. Особенности характера проявляются в стиле общения, выборе профессии, предпочтениях в искусстве и литературе.

Третью составляющую составляют способности, понимаемые как потенциальные возможности человека достигать высоких результатов в той или иной области деятельности. Способности зависят от природных задатков, условий развития и целенаправленных усилий по их совершенствованию. Среди способностей выделяют интеллектуальные, творческие, коммуникативные и профессиональные. Интеллектуальные способности связаны с уровнем умственного развития, творческими — с креативностью и оригинальностью мышления, коммуникативными — с умением устанавливать контакты и эффективно общаться, профессиональными — с компетенциями в конкретной сфере деятельности.

Четвёртая составляющая — направленность личности, определяющая отношение человека к миру и себе самому. Направленность выражается в интересах, убеждениях, мировоззрении и жизненной позиции. Она формирует внутренний стержень личности, придаёт смысл действиям и выборам. Человек стремится реализовать себя в тех видах деятельности, которые соответствуют его направленности, реализуя тем самым свою жизненную миссию.

Наконец, пятую составляющую составляет самосознание, понимаемое как осознание собственной личности, своего места в мире и отношения к другим людям. Самосознание развивается постепенно, проходя стадии идентификации, рефлексии и интеграции. Оно обеспечивает человеку возможность контролировать своё поведение, планировать будущее и оценивать собственные поступки.

Таким образом, личность представляет собой сложное и многослойное явление, зависящее от множества факторов. Её исследование имеет большое значение для науки и практики, позволяя глубже понять человеческие проблемы и находить эффективные пути их решения. Понимание природы личности открывает новые перспективы для совершенствования методов диагностики, терапии и профессионального отбора, обеспечивая каждому человеку возможность полноценно развиваться и раскрывать свой потенциал.

Одной из центральных категорий психологии является структура личности, отражающая организацию и взаимодействие отдельных компонентов, входящих в состав личности. Структура личности состоит из нескольких уровней, каждый из которых обладает своими особенностями и функциями. Рассмотрим подробнее эти уровни.

Первый уровень — базовый, включает врожденные свойства организма, такие как физиологическая конституция, биохимия мозга и гормональный баланс. Этот уровень задаёт общие рамки для дальнейшего развития личности, определяя физические и психофизиологические предпосылки.

Второй уровень — темперамент, характеризующий активность, эмоциональность и адаптивность индивида. Темперамент зависит от наследственности и влияет на выбор профессий, друзей и хобби. Например, экстраверты предпочитают активную социальную жизнь, интроверты же находят удовольствие в одиночестве и внутренней работе над собой.

Третий уровень — характер, формирующийся в результате воспитания и жизненного опыта. Характер определяет способы взаимодействия с внешним миром, привычки и манеру поведения. В отличие от темперамента, характер может меняться под влиянием внешних обстоятельств и сознательных усилий.

Четвертый уровень — способности, заключающиеся в природных задатках и развитых талантах. Способности обеспечивают эффективность деятельности и служат основой для достижения успехов в выбранной сфере. Развитие способностей возможно благодаря обучению, практике и созданию благоприятных условий.

Пятый уровень — направленность, охватывающая ценности, установки и смыслы, организующие деятельность и общение человека. Направленность связана с потребностями и интересами личности, регулируя её желания и намерения. Именно этот уровень определяет общую линию поведения и устремлённость человека.

Шестой уровень — самосознание, представляющее собой способность воспринимать самого себя как объект размышлений и наблюдений. Самосознание позволяет человеку критически относиться к своим поступкам, анализировать собственное состояние и улучшать качество жизни. Уровень самосознания повышается с возрастом и опытом, обогащаясь новыми знаниями и впечатлениями.

Все перечисленные компоненты находятся в постоянном взаимодействии друг с другом, создавая уникальную картину личности. Индивидуальность проявляется именно в сочетании свойств, присущих разным уровням структуры. Каждое свойство дополняет другое, придавая особую окраску всему образу личности.

Для наглядности рассмотрим пример. Представьте молодого инженера с высоким IQ, сильным характером и ярко выраженной направленностью на достижение карьерных высот. Такой человек обладает отличным потенциалом для быстрого продвижения по службе, однако, если его амбициозность сочетается с недостатком эмпатии и уважения к коллегам, его успехи могут оказаться кратковременными. Без должного внимания к социальным аспектам карьера рискует закончиться конфликтами и разочарованиями.

Анализируя подобные случаи, психологи приходят к выводу, что успешность человека определяется не только его талантами и возможностями, но и умением гармонично сочетать разные элементы своей личности. Оптимальное сочетание базовых качеств с осознанным выбором целей и форм поведения создаёт основу для полноценной реализации потенциала.

Чтобы составить точный психологический портрет личности, специалисты проводят специальные исследования, используя разнообразные методы

диагностики. Наиболее распространёнными методами являются наблюдение, тестирование, интервью и анкетирование. Наблюдение позволяет зафиксировать поведение человека в естественных условиях, тестирование — измерить конкретные показатели, интервью — выяснить мнение субъекта относительно собственного образа, анкеты — собрать подробную информацию о прошлом опыте и текущих обстоятельствах.

Применение перечисленных методов даёт возможность сформировать полное представление о личности клиента, учитывая как положительные, так и отрицательные стороны. Однако стоит помнить, что любая методика имеет ограничения и погрешности, поэтому важно применять их совместно, комбинируя достоинства каждого метода.

Итак, мы рассмотрели основные составляющие структуры личности и познакомились с методиками, используемыми для их выявления. Осознавая сложность и многообразие процессов, происходящих внутри человека, можно уверенно утверждать, что познание собственной личности — увлекательное путешествие длиной в жизнь. Чем больше мы узнаём о самих себе, тем легче нам справляться с трудностями и радоваться победам.

Одним из важнейших направлений современной психологии является изучение типов личности, классификация которых основана на различных критериях и теориях. Типология позволяет систематизировать разнообразие человеческих проявлений, облегчая понимание индивидуальных различий и повышая точность предсказания поведения. Существует множество классификационных схем, каждая из которых выделяет определённые признаки и характеристики.

Одна из первых типологий была предложена Гиппократом ещё в Древней Греции. Согласно его теории, существуют четыре основных типа темперамента: сангвиник, холерик, флегматик и меланхолик. Сангвиники активны, подвижны и легко адаптируются к новым условиям. Холерики импульсивны, решительны и способны быстро реагировать на изменения ситуации. Флегматики стабильны,

спокойны и медленно принимают решения. Меланхолики чувствительны, впечатлительны и склонны к глубоким переживаниям.

Позже Юнг предложил свою теорию, основанную на доминировании функций сознания. Он выделил два противоположных полюса: экстравертированный и интровертированный типы. Экстраверты получают энергию извне, активно взаимодействуют с людьми и открыты новому опыту. Интроверты черпают силы изнутри, предпочитают уединение и глубокий внутренний мир. Эта дихотомия стала отправной точкой для многих последующих исследований.

Карл Густав Юнг развивал идеи архетипов, считая, что в каждом человеке заложены универсальные символы и образы, существующие независимо от культуры и исторического периода. Архетипы влияют на мышление, эмоции и поведение, играя значительную роль в формировании личности. Среди архетипов выделяются такие фигуры, как герой, мать-земля, мудрец и трикстер. Герой символизирует борьбу за свободу и независимость, мать-земля олицетворяет заботу и защиту, мудрец ассоциируется с знанием и пониманием, а трикстер воплощает игривость и непредсказуемость.

Альберт Адлер разработал концепцию комплекса неполноценности, утверждая, что чувство недостаточности возникает у ребёнка вследствие физических недостатков или неблагоприятных условий воспитания. Компенсируя свою слабость, ребёнок вырабатывает стратегию преодоления трудностей, формирующую уникальный стиль жизни. Таким образом, комплексы становятся движущими силами развития личности, заставляя искать выходы из сложных ситуаций.

Ганс Айзенк продолжил развивать тему типологии, предложив трёхфакторную модель личности. Он выделял три измерения: экстраверсию—интроверсию, стабильность—невротичность и психотизм—нормальность. Модель Айзенка позволила создать стандартизированные шкалы оценки, применяемые в клинической практике и профессиональном отборе.

Рассматривая современные классификации, нельзя обойти стороной пятифакторную модель («Большая Пятёрка»), разработанную Полом Коста и Робертом Макре. Данная модель описывает пять широких измерений личности: открытость опыту, добросовестность, экстраверсия, доброжелательность и невротизм. Открытые люди любопытны и любят новшества, добросовестные — ответственные и организованы, экстраверты коммуникабельны и оптимистичны, доброжелательные — отзывчивы и щедры, невротичные подвержены тревоге и негативным эмоциям.

Типология необходима для лучшего понимания индивидуальных особенностей и предвидения возможных последствий. Правильная оценка типа личности облегчает диагностику, терапию и обучение, помогая каждому человеку обрести уверенность в собственных силах и двигаться вперёд.

Теперь обратимся к рассмотрению механизма формирования личности, являющегося предметом пристального интереса учёных и практических работников. Формирование личности представляет собой сложный и длительный процесс, протекающий на протяжении всей жизни человека. Начало этого процесса связано с ранним детством, когда ребенок впервые сталкивается с окружающим миром и усваивает первичный опыт. Здесь огромную роль играют родители и ближайшее окружение, создающие условия для первоначального этапа социализации.

Далее следуют школьные годы, когда дети начинают осваивать общественные правила и нормы, учиться строить взаимоотношения с ровесниками и взрослыми. В подростковом периоде происходят значительные изменения в физическом развитии, формируются первые самостоятельные взгляды и убеждения. Молодёжь пытается найти своё место в обществе, экспериментирует с разными ролями и моделями поведения.

Следующим этапом является зрелый возраст, характеризующийся достижением определённого статуса, принятием ответственности за собственную судьбу и участие в общественных процессах. Взрослые

продолжают приобретать новые знания и умения, решают важные жизненные задачи, строят карьеру и создают семью.

Завершающим периодом является старость, сопровождающаяся переосмыслением пройденного пути, подведением итогов и передачей накопленного опыта следующим поколениям. Старея, человек учится жить с ограниченными ресурсами здоровья и энергии, сохраняя при этом достоинство и уважение к самим себе.

Процесс формирования личности непрерывен и проходит в тесной связи с культурой, обществом и индивидуальным развитием. Каждая стадия сопровождается особыми задачами и проблемами, решение которых либо укрепляет, либо ослабляет позицию личности в обществе. В конечном итоге личность формируется как результат взаимопроникновения биологических и социальных факторов, создавая уникальные комбинации признаков и качеств.

Индивидуальная траектория развития личности обусловлена множеством переменных, среди которых важнейшими являются наследственность, среда обитания, система воспитания и индивидуальный опыт. Наследственность передаёт ребёнку исходные характеристики, служащие базой для последующего роста. Окружающая среда предоставляет условия для приобретения необходимых навыков и способностей. Система воспитания воспитывает необходимые социальные компетенции и формирует нравственные принципы. Наконец, личный опыт служит источником уникальности и своеобразия каждой отдельно взятой личности.

Воздействие этих факторов неравномерно распределено во времени и пространстве, что порождает бесконечное разнообразие вариантов развития. Одни люди рождаются в богатых семьях и имеют доступ к лучшим ресурсам, другие растут в бедности и вынуждены бороться за выживание. Кто-то получает хорошее образование и достигает значительных успехов, кто-то ограничивается рамками ограниченного круга знакомств и узкого кругозора. Все эти обстоятельства накладывают отпечаток на дальнейшую жизнь человека, предопределяя направления его развития.

Вместе с тем, несмотря на сильное влияние внешних условий, решающее слово остаётся за внутренним фактором — свободой воли и способностью самостоятельно выбирать путь развития. Многие известные исторические деятели начинали свою жизнь в крайне неблагоприятной среде, но благодаря упорству и целеустремленности добились выдающихся результатов. Другие, обладавшие всеми необходимыми условиями, теряли шансы на реализацию своего потенциала, останавливались на полпути или вовсе опускались до посредственности.

Таким образом, формирование личности — это активный творческий процесс, осуществляемый каждым человеком в течение всей жизни. Степень активности и продуктивности этого процесса зависит от совокупности внутренних и внешних условий, складывающихся в каждый конкретный период существования. Исследователи подчеркивают необходимость гармоничного сочетания природной одарённости и приобретаемых навыков, обеспечивающих оптимальный рост и раскрытие творческого потенциала личности.

Исследовательская работа в области психологии направлена на глубокое проникновение в тайны личности, раскрытие механизмов её функционирования и разработку рекомендаций по улучшению качества жизни. Для научного сообщества важны фундаментальные открытия, позволяющие пролить свет на неизвестные ранее закономерности и тенденции. Практическое применение научных разработок способствует повышению эффективности управления персоналом, оптимизации образовательных технологий и расширению спектра услуг в сфере здравоохранения.

Развитие психологии продолжается быстрыми темпами, открывая всё новые горизонты познания человека. Современные технологии предоставляют дополнительные инструменты для анализа и моделирования психических явлений, углубляют наши знания о природе личности и повышают нашу готовность помогать людям преодолевать трудности и достигать поставленных целей.

Основой любого психологического портрета являются факты, добытые путём наблюдения, тестирования и бесед. Отчётливо структурированные и проверенные данные делают возможным составление точной карты внутреннего мира личности, доступной для анализа и коррекции. Вместе с тем, создание полноценного психологического портрета требует не только технических навыков, но и глубокой интуиции, чуткости и внимательности исследователя.

Полученная картина должна служить основой для выработки оптимальной стратегии действий, направленной на повышение эффективности деятельности и улучшение качества жизни. Целью специалиста является содействие раскрытию потенциала личности, устранение препятствий на пути к счастью и благополучию. Таким образом, психологический портрет становится мощным инструментом поддержки и развития человека, открывающим перед ним широкие перспективы для самовыражения и творчества.

Таким образом, изучая личность, мы получаем ключ к разгадке тайн человеческой природы, познанию законов, управляющих поведением и сознанием. Результаты наших изысканий применимы в самых разных областях жизни — от педагогической деятельности до бизнеса и медицины. Благодаря этому знание о личности становится неотъемлемой частью нашего повседневного бытия, превращаясь в важный фактор прогресса и процветания человечества.

1.2 Построение портрета личности

Построение психологического портрета личности — задача одновременно интересная и сложная, поскольку охватывает множество аспектов человеческой натуры. Чтобы успешно справиться с ней, важно учесть целый ряд факторов, каждый из которых по-своему важен и информативен.

Начнём с характера, который представляет собой совокупный набор устойчивых индивидуальных черт личности. Данные черты подразделяются на несколько групп, каждая из которых относится к различным сферам жизни и деятельности человека. Во-первых, к работе относятся такие качества, как

добросовестность, трудолюбие, аккуратность, склонность к творчеству, пассивность, лень, нечестность, инициативность. Эти черты показывают, каким образом человек подходит к выполнению своих обязанностей и какую ценность он вкладывает в рабочий процесс.

Во-вторых, есть группа черт, относящихся к поведению в обществе и отношениям с другими людьми. Сюда входят уважение к другим, чуткость и отзывчивость, замкнутость, черствость, общительность, грубость. То, как человек воспринимает и взаимодействует с окружающими, сильно влияет на его репутацию и положение в коллективе.

Отношения к материальным благам также заслуживают отдельного внимания. Небрежность или опрятность, халатное или бережливое отношение к вещам демонстрируют степень заботы и ответственности, которую проявляет человек в повседневной жизни.

Что касается четвертой группы черт, касающейся самовосприятия, сюда включают высокомерие, скромность, эгоизм, чувство собственного достоинства, самокритику, тщеславие. Самоощущение человека определяет его реакцию на критику, похвалу и признание.

Затем следует обратить внимание на темперамент — одну из важнейших черт личности, значительно влияющую на общий психологический портрет. Некоторые люди медлительны, спокойны и невозмутимы, их внутреннее состояние практически не подвергается резким изменениям. Другие же стремительно реагируют на происходящее, проявляя яркие эмоциональные вспышки. Как правило, темперамент легко распознать путем простого наблюдения за поведением человека в течение короткого промежутка времени.

Традиционно выделяют четыре классических типа темперамента: флегматик, холерик, меланхолик и сангвиник. Флегматики известны своей спокойностью и размеренностью, тогда как холерики склонны к быстрой смене настроений и бурным эмоциональным реакциям. Меланхолики переживают каждую ситуацию глубоко и болезненно, а сангвиники отличаются живостью и легкостью восприятия действительности.

Еще одной ключевой чертой является интеллектуальность — система психических процессов, обеспечивающих оценку ситуации, принятие решений и регуляцию поведения. Данный параметр сложно однозначно определить, ведь интеллект бывает эстетическим, телесно-кинестетическим, пространственным, логико-математическим, музыкальным, натуралистическим и внутриличностным. Иногда невербально-глупый человек неожиданно демонстрирует высокий уровень интеллекта в другой сфере жизнедеятельности.

Немаловажную роль играет мотивация — психофизиологический механизм, контролирующий поведение, устанавливающий его стабильность, направление, активность и организацию. Следует установить причину, почему человек совершает те или иные действия, какая цель заставляет его действовать определенным образом. Помимо прочего, полезно разобраться, преобладает внутренняя или внешняя мотивация у вашего испытуемого.

Способности представляют собой личностные качества, выступающие условием успешного выполнения определенной деятельности. Они определяются не только имеющимися навыками и знаниями, но и степенью глубины, скорости и прочности освоения приёмов и способов деятельности. Стоит упомянуть понятия привязанности и таланта, первое из которых отражает мотивационную сторону деятельности, второе — уникальное сочетание навыков, полученных естественным путем или в детстве.

Самоконтроль — это умение управлять собственными мыслями, действиями и чувствами. Вам предстоит выяснить, способен ли человек отказываться от немедленного удовлетворения потребностей ради долгосрочного результата. Эмоциональность — это показатель того, как человек выражает чувства, характер и настроение, его реакция на внешний мир. Эмоции тесно переплетаются с темпераментом, например, холерики мгновенно переключаются между эмоциями, а флегматики остаются невозмутимыми долгое время.

Коммуникативные способности особенно интересны, поскольку одна и та же личность способна совершенно иначе проявить себя в зависимости от

обстоятельств и участников разговора. Внутренние установки, убеждения и эмоциональное состояние постоянно варьируются, поэтому рассматривать данную категорию нужно с учётом разных аспектов.

Волевые качества отображают способность принимать решения, основанные на анализе и доводах, и последовательно придерживаться выбранных действий. Особое внимание уделяйте способности принимать трудные решения, соблюдению дисциплины и выдержке.

Самооценка — это представление человека о своем месте среди других, оценка себя и своих достоинств, своих сильных и слабых сторон. После исчерпывающей детализации каждого аспекта, наступает этап написания собственно психологического портрета, для чего можно воспользоваться специальными опросниками, такими как тест на уровень субъективного контроля Дж. Роттера, индивидуально-типологический тест Л.Н. Собчик, опросник Р. Кеттела, характерологический опросник Леонгарда. Однако это лишь вспомогательные средства, помимо которых можно полагаться на собственный взгляд и ощущения, подкрепленные достаточным количеством собранной информации.

Подводя итог, можно сказать, что составление психологического портрета — это искусственное воссоздание сложной мозаики, состоящей из множества мелких деталей, каждая из которых важна и информативна. Тщательно проанализировав каждую грань личности, вы получите полную и точную картину, позволяющую сделать вывод о том, как тот или иной человек воспримет определенную ситуацию и какое поведение выберет в дальнейшем.

ГЛАВА 2. ТЕХНОЛОГИЯ TEXT MINING

2.1 Основные понятия и терминология Text Mining

Текстовые данные присутствуют везде. На самом деле, интернет представляет собой гигантскую неструктурированную коллекцию текстов самых различных форматов. С помощью текстов можно выполнять множество задач. К примеру, используя содержание блога, можно прогнозировать рейтинг записи и число людей, которым она понравится. Это позволит вам оценить ее привлекательность еще до публикации. Также можно проанализировать текст, чтобы определить его эмоциональную окраску — положительную или отрицательную. Это весьма полезно, когда речь идет о клиентах банка, оставляющих отзывы: если встречаете отрицательный отзыв, легко можно быстро предпринять меры для улучшения лояльности клиента.

Кроме того, возможно пытаться выявить тематику научных статей, загружаемых в электронные библиотеки, чтобы определить, к какому предмету они относятся. Статья может быть автоматически отнесена к соответствующей категории или же автору могут быть предложены варианты выборов. Можно также заниматься задачами кластеризации, например, группировать новости по общему сюжету. Если несколько новостных сайтов пишут об одном и том же событии, разумно объединить их в одну группу. Вдобавок, можно искать синонимы. Например, слова «кресло» и «стул» отличаются по значению, но все же обсуждают одно и то же. Для компьютера эти слова — просто набор символов, и простые алгоритмы не позволяют понять, что они практически являются синонимами.

Существуют и более сложные задачи, например, выделение именованных сущностей в тексте. Либо создание краткой аннотации, которая охватывает основные моменты, обсуждаемые в большом тексте. Вопросно-ответные системы, которые набирают популярность, также относятся к сложным задачам. Их цель — разработка модели, которая принимает вопрос в свободной форме и находит ответ, основываясь на обширном неструктурированном корпусе, например, Википедии.

Главная трудность заключается в том, что текст представляет собой некоторый набор данных, состоящий из букв переменной длины. Каждый текст может быть описан разным количеством символов, и необходимо каким-то образом преобразовать этот текст в фиксированное количество признаков.

Рассмотрим два ключевых этапа обработки текста: токенизацию и нормализацию. Токенизация заключается в разбивке текста — длинной строки — на отдельные слова, с которыми уже можно работать. Нормализация же предполагает приведение каждого слова к его начальной форме. Токенизация может включать множество нюансов, и важно учитывать специфику задачи: нужно ли приводить слова к нижнему регистру, удалять ли знаки препинания и т.д. В нормализации применяются два основных подхода — стемминг и лемматизация. Лемматизация обычно более эффективна, но и занимает больше времени.

Современные системы интеллектуального анализа способны за несколько шагов извлекать ключевые смыслы из "корпуса", определяя, пригоден ли текст для поставленной задачи, и выявляя детали его содержания. Под "корпусом" понимается набор текстов, соответствующих заранее установленным параметрам: сначала формируются критерии, затем подбираются подходящие тексты.

Машинное обучение - это область обучения, которая дает компьютерам возможность учиться без явного программирования. ML - одна из самых захватывающих технологий, которые когда-либо встречались. Как видно из названия, она дает компьютеру способность к обучению, делает его более похожим на людей. Машинное обучение активно используется сегодня, возможно, в большем количестве приложений, чем можно было бы ожидать.

Типы проблем машинного обучения.

Существуют различные способы классификации проблем машинного обучения.

1. На основании характера обучающего «сигнала» или «обратной связи», доступной для системы обучения. Процесс обучения продолжается до тех пор,

пока модель не достигнет желаемого уровня точности данных тренировки. Вот некоторые примеры из жизни.

Классификация изображений. Проводится тренировка с изображениями или метками. Затем в будущем дается новое изображение, ожидая, что компьютер распознает новый объект.

Кластеризация. Необходимо научить компьютер разделить похожие данные на кластеры, это важно в научных исследованиях.

Визуализация больших данных. Использование компьютера для визуализации данных больших размеров.

Генеративные модели. После того, как модель уловит распределение вероятностей входных данных, она сможет генерировать больше данных. Это может быть очень полезным для повышения надежности классификатора.

Функция извлечения текста, которая извлекает текстовую информацию, является основой обработки текста. Базовая единица называется текстовой функцией. Выбор набора признаков из нескольких эффективных способов уменьшить размерность пространства признаков, цель этого процесса состоит в извлечении признаков. Во время извлечения объектов некоррелированные или лишние объекты будут удалены. Как метод предварительной обработки данных алгоритма обучения, извлечение признаков может улучшить точность алгоритма обучения и сократить время. Выбор из части документа может отражать информацию о содержании слов, а вычисление веса называется извлечением текстового признака. Распространенные методы извлечения текстовых объектов включают в себя методы фильтрации, слияния, отображения и кластеризации. Традиционные методы извлечения признаков требуют ручных функций. Создание вручную эффективной функции - это длительный процесс, и глубокое обучение может быть нацелено на новые приложения и быстро получить новое эффективное представление характеристик из обучающих данных.

Ключевой аспект глубокого обучения заключается в том, что эти слои функций не разрабатываются людьми, а извлекаются из данных с использованием процедуры обучения общего назначения. Глубокое обучение

требует очень мало ручной разработки, поэтому оно может легко воспользоваться увеличением объема доступных вычислений и данных. Преимущество глубокого обучения заключается в выявлении модели неструктурированных данных, и большинство людей знакомы со средствами массовой информации, такими как изображения, звук, видео и текст, все из которых принадлежат таким данным. Глубокое обучение дало чрезвычайно многообещающие результаты для различных задач в понимании естественного языка, в частности, классификации тем, анализа настроений, ответов на вопросы языкового перевода. Его глубокая архитектура обеспечивает глубокое изучение возможности решения гораздо более сложных задач ИИ. В настоящее время представление функции глубокого обучения включает в себя авто-кодер, ограниченную модель Больцмана, сеть глубокого убеждения, сверхточную нейронную сеть и рекуррентную нейронную сеть и т. д.

Качество обучающих данных имеет решающее значение для производительности модели машинного обучения. Качество измеряется как последовательностью, так и точностью помеченных данных. Стандартные методы расчета качества данных обучения - это эталонные тесты (золотой стандарт), консенсус и анализ.

Качество состоит из последовательности и точности. Дело не только в том, насколько точны данные. Согласованность - это степень, в которой данные согласуются друг с другом. Согласованность предотвращает случайный шум, обеспечивая правильную или неправильную маркировку в согласованном порядке. Согласованность измеряется с помощью согласованного алгоритма. Без автоматизации передовых инструментов искусственного интеллекта этот процесс является ручным, требует много времени и требует безопасности. Поскольку маркеры могут быть постоянно правильными или постоянно неправильными, одной лишь высокой последовательности недостаточно для полного объяснения качества.

Обзор является еще одним методом обеспечения точности. После того, как метки были завершены, доверенный эксперт проверяет точность меток. Обзор

обычно проводится путем визуальной выборочной проверки меток, но некоторые проекты проверяют все метки. Обзор часто используется для выявления низкой точности и несоответствий в процессе маркировки, в то время как эталонные тесты часто используются, чтобы получить представление о производительности.

Проблема проведения анализа текстовой информации Text Mining на основании машинного обучения представляется достаточно актуальной в настоящий момент времени. Новые механизмы сравнения текстовых данных из разных источников могут применяться в различных сферах жизнедеятельности человека. Использование различных математических методов к проблеме анализа текста берет начало еще в прошлом столетии. Тогда предпринимаются первые попытки статистической обработки текстовой информации, которые состояли в проведении подсчета словоупотреблений в работах разных авторов или в текстовых образцах одного автора и сравнении наборов полученных частот. Для сравнения различных источников обычно выбирают совокупность каких-либо критериев, которые отражают изучаемые переменные текста. Объединение различных текстов в группы можно осуществлять по совпадению вероятности появления одного или многих признаков, но эта вероятность, чаще всего, неизвестна.

Кластеризацией называют разбивку фраз из текста на группы, каждую из которых называют кластером. Внутри группы должны быть размещены «похожие» выражения, а фразы в разных группах должны как можно более сильным образом отличаться. Например, набор слов и словосочетания, как «молоко», «iphone 6», «iphone 10», «йогурты», «масло» и «mac apple» может быть разбит на такие кластеры: {mac apple}, {iphone 6, iphone10}, {молоко, йогурты, масло}. Кластеризация необходима для того, чтобы:

- проводить объединение фраз по смыслу;
- анализировать семантическое ядро;
- создавать группы фраз для размещения в конкретном месте;

- распределять слова по страницам и организовать полноценную структуру необходимого массива текстовых данных;

- находить некластеризованные ключевые словосочетания, которые нельзя отнести ни к одной из групп.

У документа есть векторные модели, которые мы можем использовать для поиска решения довольно большого количества задач информационного поиска, в пример можно привести кластеризацию каких-либо документов [28].

Кластеризацию документов с текстом можно определить, как процесс поиска различных наборов групп сочетаний слов, имеющих примерно одинаковое значение по смыслу (это однородный класс в данной предметной области). Когда вы проводите кластеризацию может появиться две проблемы. Первая проблема — это то, что заранее все кластеры, которые можно отнести к истинным, будут неизвестны вам. Вторая проблема — это то, что количество данных кластеров, на которые можно разделить конкретную коллекцию, тоже является неизвестным.

Проводя анализ документов с текстом, возникает необходимость, чтобы в кластере распределенные данные текстовые имели схожесть по смыслу, задающуюся численным образом, когда мы используем такие методы:

1. Выяснение какой является величина близости между одним и другим элементом, принадлежащих кластеру, попарно. Её значение ниже определенного порога;

2. Определение величины значения близости между центроидами данных кластеров и каким-либо из перечисленных документов множества. Такая величина, как правило, ниже заданного определенного порога. Центроиды конкретных кластеров можно рассчитать как среднее арифметическое между документами в этом кластере.

В процессе анализа кластер формируется на основе схожести описаний каждого из документов, что отличается от проведения классификации. В данной группе заранее характеристики не задаются.

Можно обратить внимание на то, что задачи проведения смыслового анализа документов с текстом — одни из основных задач при анализе существенного количества слабоструктурированной или неструктурированной текстовой информации на языках естественных, а также при выполнении какого-либо информационного поиска. Кроме того, когда точное представление о составе данных и их структуре отсутствует, кластеризацию рассматривают как единственный метод для решения задач [31].

Когда кластеризацию проводят на основе семантического анализа могут появиться некоторые проблемы, связанные с: выбором кластера, который будет первым; зависимостью объема текстовой информации от уровня качества кластеризации, нахождения общего количества этих кластеров, связь будет отсутствовать между похожими по контенту разными текстами, где обычно используют неодинаковую лексику и т.п. Чтобы решить проблемы, связанные с разной лексикой, необходимо использовать методы, основывающиеся не только на сходстве по используемой лексике, но ещё и на принципах семантической смежности или принципах ассоциативности.

Разберём основные направления современных исследований в области кластеризации полуструктурированной текстовой информации.

Сегодня применяют большое количество методов для кластеризации документов. Например, такие, как:

1. Scatter/Gathering, K-mean, Concepts Indexing (CI);
2. Self-Organizing Map (SOM)
3. Single Links, Complete Links, Groups Average;
4. Custom Search Folder, Latent Semantics Analysis/Indexing (LCA/LSI);
5. Suffix Trees Clusterings (STCS);

и многие другие.

Впрочем, несмотря даже на большие успехи в этой области, все еще присутствуют следующие насущные нерешённые задачи кластеризации:

1. повышение производительности вычисления и ускорение процесса кластеризации

2. выбор начального расположения ядра кластера
 3. увеличение точности проведения кластеризации
 4. масштабирование и управление размерностью массива информации
- и т.д.

Большое количество исследователей посвящает свои работы многим актуальным проблемам кластеризации на основе семантического анализа полуструктурированных текстов. [19].

Предложенные в данных работах модели кластеризации весьма разные и нередко представляют собой гибридные решения на основании нынешних алгоритмов.

Так, в работе [29] разбирают алгоритм «искусственных пчелиных колоний» (ABC), являющийся методом стохастической оптимизации. Те, кто написал работу, используют гибридный алгоритм (MABKM) на основании использования модифицированных алгоритмов ABC и K-Mean. Решение, которое создается ABC, берут как первоначальное решение для другого алгоритма K-Mean. Производительность такого алгоритма оценивают в терминах разных критериев на шести обыкновенных выборках данных из алгоритма UCI MLR и проводят сравнение с результатами работы алгоритмов ABC и K-Means. Подтверждает превосходство использования алгоритма экспериментальный результат MABKM для кластеризации полуструктурированных данных с текстом и отображает, что данный метод предлагает возможность сторониться локального оптимума и выявлять лучшее значение целевой функции с более низкой величиной отклонения, если сравнивать с иными алгоритмами. Данный метод ещё превосходит иные алгоритмы по значению нескольких показателей (F-мера, CH, ARI и DB) и предоставляет возможность лучшего ранжирования среди других трех методов, которые были рассмотрены. Результат же отображает, что использование модифицированного алгоритма предоставляет возможность наилучшего разделения кластеров и мы можем прийти к выводу, что MABKM — это метод кластеризации данных, который ещё можно использовать. Но для такого метода

нужно сильное улучшение для автоматической кластеризации без каких-либо начальных знаний о количестве кластеров.

В работе [32] дают некоторые алгоритмы кластеризации текстовой информации: алгоритмы с функциями весовых схем и динамического уменьшения размера и алгоритмы с наличием выбора функции. В процессе проведения кластеризации документ с текстовой информацией делят на несколько кластеров (когерентных) в соответствии с уникальным образом, который подобрал информативный коэффициент, используя необходимую функцию для оценки, которая обычно зависит от частоты. Для каждого документа информативную функцию выбирают, используя метод выбора объектов. Проверенными методами кластеризации можно назвать: генетические алгоритмы (GA), алгоритмы оптимизации (PSO) и алгоритмы гармонического поиска (HS). Данные алгоритмы могут работать на основании выбора признаков, устанавливаемых с применением новых схем взвешивания: весового коэффициента длины (LFW), имеющего зависимость от частот и появления признака в иных документах, которые анализируются. В нескольких работах даются новые методы уменьшения динамических размеров (DDR). Эти алгоритмы применяются для уменьшения количества функций, которые используют в кластеризации, и, следовательно, для увеличения эффективности работы. K-Mean применяют для кластеризации наборов текстовых документов на основе признаков, полученных при динамическом сокращении. Дают оценку нескольким тестовым наборам данных с различным размером и сложностью. Проведение анализа результата исследования отображает, что оптимизация с динамическим снижением и весами длин предоставляет оптимальный результат почти для любого набора данных, который тестируют.

В исследовании [30] представлен масштабируемый алгоритм для выполнения расширенной кластеризации, который основывается на реализации n-грамм для уменьшения высокой размерности и достижения качественного объединения тестовых документов в кластеры. Также в работе представлен сравнительный анализ, который продемонстрировал, что предложенные

алгоритмы показывают лучшие результаты при использовании текстовых данных с удалением стоп-слов, по сравнению с случаями, когда стоп-слова не исключаются.

Авторы работы [30] предлагают метод, который улучшает работу систем в части извлечения значимых событий из социальных сетей. В предыдущих исследованиях, таких как в [31], события обрабатывались по отдельности, что приводило к созданию множества бессмысленных сообщений из-за отсутствия данных, рассредоточенных по большому количеству текстовых сегментов. Кроме того, обрабатывалось значительное количество несущественных текстов, что увеличивало время обработки и снижало производительность системы. В данном исследовании предложен метод кластеризации, который группирует семантически связанные текстовые фрагменты, фильтрует шум и снижает объем информации для анализа, сосредоточиваясь только на релевантных сегментах текста для извлечения данных. Алгоритм кластеризации интегрирован в потоковую обработку данных в рамках инфраструктуры под названием Storms, что позволяет создавать решения для кластеризации потоковых данных и масштабирования быстро растущих объемов информации.

Существует множество других алгоритмов кластеризации. Их основная концепция заключается в объединении схожих последовательностей в один кластер или класс на основе общих характеристик. Выбор алгоритма, как правило, определяется конкретной задачей. В контексте текстовой информации в качестве сравниваемых элементов выступает последовательность слов или ее атрибуты (например, веса слов в тексте, виды именованных сущностей, тональность и т.д.). Следовательно, текст изначально преобразуется в вектор, с которым затем проводятся различные операции. При этом возникают ряд сложностей, связанных с определением начального кластера, зависимостью качества кластеризации от длины текстов, расчетом общего количества кластеров и другими аспектами. Однако самой трудной задачей остается отсутствие связи между близкими по смыслу наборами текстов, использующими разную лексику. В таких случаях объединение следует осуществлять не только

на основе сходства, но и с учетом семантической ассоциативности или смежности [32].

Первый путь состоит в получении тезаурусов. Он является достаточно трудоемким и полностью будет определяться поставленными задачами. В связи с этим, разработка всеобъемлющего и универсального тезауруса на данный момент представляется задачей, не имеющей реального решения.

А путь второй - это алгоритмический путь, также обладает своими недостатками. Прежде всего, это проблема «туманности» самого метода и неочевидность его применения для текстовой информации. Например, для алгоритма LDA требуется выполнение условий нормальности распределения, что при нахождении решения лингвистической задачи не всегда будет удовлетворяться. Как правило, подобные алгоритмы обладают большим количеством параметров, определение которых возможно только эмпирически и может оказать существенное влияние на качество решения (например, уменьшение сингулярного значения диагональной матрицы алгоритма LSA, что сильно нелинейно влияет на результаты).

2.2 Задачи, решаемые Text Mining

Работа с большими объемами неструктурированной или слабо структурированной текстовой информации часто требует реализации процесса информационного поиска, включая обнаружение скрытых закономерностей, извлечение знаний и классификацию. Это, в свою очередь, предполагает необходимость разработки модели для автоматизации указанных процессов. Важно отметить, что в данном случае отсутствует физически измеримая величина, что затрудняет выполнение математических операций и построение формальных моделей.

Текст представляет собой набор универсальных лингвистических переменных (единиц), которые соответствуют словам и словосочетаниям. Каждая лингвистическая переменная сама по себе не имеет числового или

логического значения, что делает невозможным построение математической модели без предварительной формализации текста [24].

Формализация анализа текстовой информации на естественном языке основывается на лексических подходах с целью поиска и классификации, и в данной области ключевую роль играют статистические семантические гипотезы (statistical semantic hypothesis). Эти гипотезы предполагают, что статистические взаимосвязи в употреблении терминов и слов могут быть использованы для выявления смыслов, заложенных в тексте [28]. Согласно гипотезе о скрытых связях, слова, термины и словосочетания, которые встречаются в схожих документах, имеют тенденцию к близкой семантической зависимости [33].

Статистическая семантическая гипотеза позволяет представлять текстовые документы в виде векторной модели. Основная идея векторной модели семантики (vector space model, VSM) заключается в том, что каждый текстовой документ может быть представлен как точка в многомерном пространстве (вектор в пространстве размерности RM). Близко расположенные друг к другу точки соответствуют документам, которые имеют схожую семантику.

2.3 Построение психологического портрета личности с помощью технологии TextMining.

Создание функций принадлежности для нечетких множеств, описывающих ущерб, нанесенный информационной системе, является ключевым аспектом. В основе теории нечетких множеств лежит понятие функции принадлежности. Следовательно, установление уровней принадлежности элементов к конкретному множеству и разработка соответствующей функции принадлежности – это фундаментальная задача при практической реализации, вне зависимости от конкретной предметной области применения.

Быстрый алгоритм обучения и интерпретируемость накопленной информации – данные факторы делают в настоящий момент нечеткую

нейронную сеть одним из наиболее эффективных и перспективных инструментов для мягких вычислений.

Основным инструментом нечетких нейронных сетей являются многослойные нейронные сети.

Нейроны состоят из элементов трех видов:

- синапсы;
- сумматоры вида:

$$s = \sum_{i=1}^n x_i w_i + b,$$

$$s = \sum_{i=1}^n x_i w_i + b,$$

- нелинейные преобразователи.

Для подборки параметров сетей (и их обучения) используют алгоритмы обратного распространения (Рисунок 1).

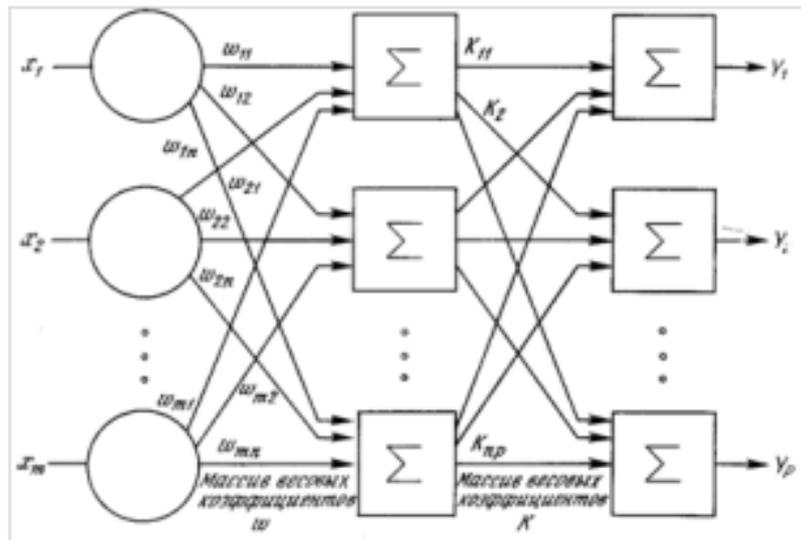


Рисунок 1 – Вид алгоритма обратного распространения

Метод нечеткого вывода в интегрированной системе также реализуется при помощи нейронной сети. Нейронную сеть используют для настройки параметров функции принадлежности, которые применяют в системе нечеткого вывода. Нечеткая нейронная сеть типа ANFIS (adaptive neuro-fuzzy inference system), используемая для решения задач аппроксимации, показала достаточно хорошие результаты.

Механизм функционирования искусственной нейронной сети (ИНС) основан на формировании соединений меж множеством вычислительных узлов, каждый из которых воспроизводит функцию отдельного нейрона в биологическом мозге. Данные нейроны могут быть реализованы физически или эмулированы посредством цифрового устройства, такого как компьютер. Каждый нейрон принимает множество входящих сигналов (Рисунок 2) и затем, используя внутреннюю систему весов, генерирует единственный исходящий сигнал, который, как правило, передается в качестве входного сигнала другому нейрону.

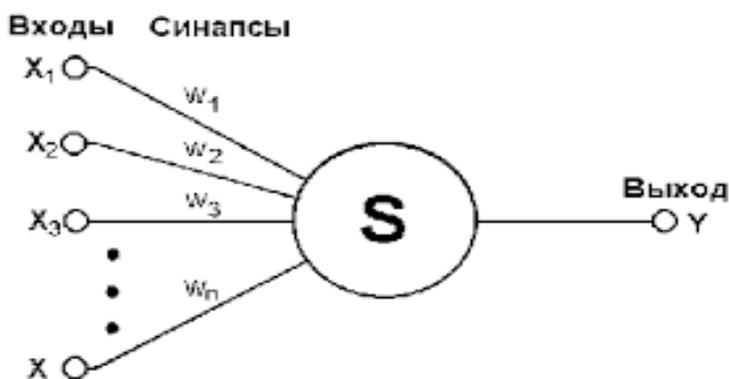


Рисунок 2 – Изображение модели нейрона

Нейроны находятся в тесной взаимосвязи и организованы в различные слои. Входной слой принимает данные, тогда как выходной слой формирует итоговый результат. Обычно между этими двумя слоями располагаются один или несколько скрытых слоев, что делает невозможным предсказание или заранее определение точного потока данных (Рисунок 3).

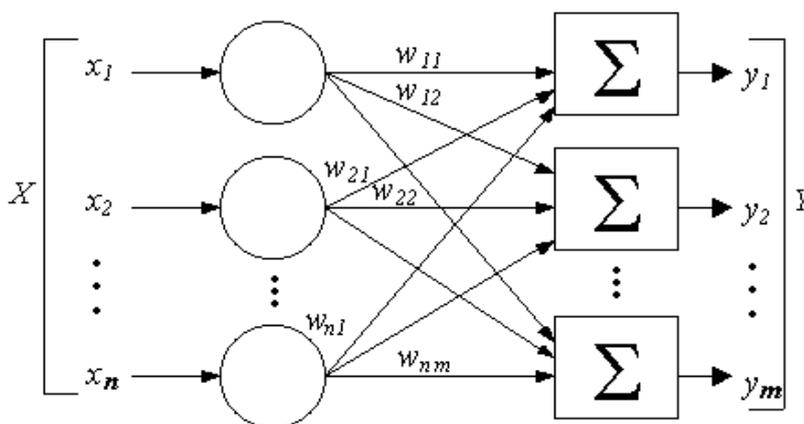


Рисунок 3 – Нейросеть однослойная

Искусственные нейронные сети обычно инициализируются с случайными весами для всех нейронов. Это означает, что они, не обладая никакими предварительными знаниями, должны пройти процесс обучения для решения конкретной задачи, для которой они разработаны (Рисунок 4). В общем, выделяют два метода обучения искусственных нейронных сетей, в зависимости от задач, которые они должны решить.

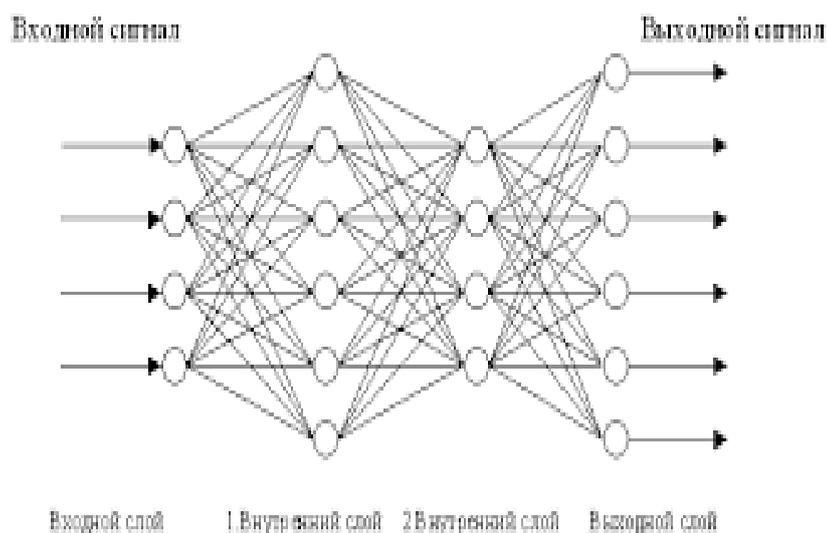


Рисунок 4 – Работа нейронной сети

ИНС являются достаточно старой областью исследований для компьютерной науки. Работа над этой техникой началась еще в 1940 году, но из-за невозможности преодоления теоретических барьеров они были забыты в 1960-1980 годах. Эти теоретические барьеры были преодолены в настоящее время, и нейронные сети теперь доступны для многих целей (Рисунок 5). Нейронная сеть представляет собой набор простых единиц, называемых нейронами. Нейрон - это линейный автомат, который реализует взвешенную сумму нескольких входов в соответствии с набором весов, а затем вычисляет функцию Хевисайда или сигмовидную функцию для получения выходного значения, называемого активацией нейрона. Выбор передаточной функции определяет, является ли нейрон двоичным или непрерывно оцененным. Чтобы сформировать нейронную сеть, эти нейроны взаимосвязаны в соответствии с данной топологией. Эта топология определяет входной уровень, при котором активация нейронов устанавливается на входные значения и выходной уровень, где считывание

активации нейронов дает ответ сети. Поэтому нейронные сети можно рассматривать как нелинейную функцию переноса из векторного пространства в другое.



Рисунок 5 – Классификация нейронных сетей

Входные веса нейронов являются параметрами этой передаточной функции. Представление входа в сеть инициирует вычисление вывода. Очень важным свойством нейронных сетей является концепция программирования на примере. Большое количество весов затрудняет получение желаемого результата. Вместо этого сеть запрограммирована примером и повторением. Она тренируется путем многократного представления пар ввода-вывода. Каждый раз, когда представлен вход, сеть догадывается о выходе. Выходная часть пары вход-выход используется для определения правильности или неправильности сети. Когда это неправильно, сеть корректируется с помощью алгоритма обучения с использованием метода градиента на выходной ошибке для изменения весов. После каждой модификации сеть становится ближе к желаемой передаточной функции, представленной базой образцов (Рисунок 6).

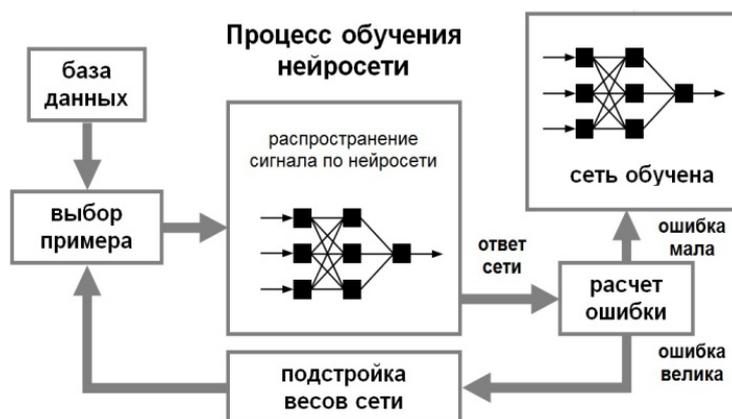


Рисунок 6 – Процесс обучения нейронной сети

В настоящее время автоматизированные и роботизированные заводы контролируются ИНС, которыми контролируются механизмы, регулируются настройки температуры, диагностируются сбои и многое другое. Эти ИНС могут дополнять или заменять квалифицированную рабочую силу, что позволяет меньшему количеству людей делать больше работы.

Рассмотрим временные ряды нейронных сетей. Библиографическое исследование показывает, что многие специалисты работали в области временных рядов и что все они работали с интеллектуальными нейронными сетями. Зная прошлые и текущие события, они учат сеть предсказать следующее. Практическое интересное исследование по моделированию поведения с нейронными сетями связано с прогнозированием цен на фондовом рынке и другими экономическими показателями [8,9]. Самая известная модель - S & P 500, которая предсказывает эволюцию Dow Jones на Уолл-стрит [24]. Вторым подходом касается изучения методов ARIMA, используемых для прогнозирования потребления воды или электроэнергии. Способность регулировать производство к потребностям является критическим фактором для компаний, работающих в сетях электроснабжения или водоснабжения [24]. Эти модели имеют в качестве входных данных N предыдущих значений серии, которую они пытаются прогнозировать, и внешние параметры, такие как погода, скорость политической партии, которые, как известно, влияют на серию. Результаты были незначительными по проблеме прогнозирования S & P 500, но приложение

прогнозирования производства воды теперь успешно используется и дает лучшие результаты, чем модель ARIMA. В этих двух предсказательных проблемах используется одна и та же нейронная сеть, многослойное восприятие с алгоритмом обучения обратному обращению ошибок, определенным Румельхартом [25].

Сеть видит N самых последних шаблонов через сдвинутое временное окно. Она учится прогнозировать следующее значение серии. Во-первых, сеть обучается точке конвергенции, используя набор ранее собранных примеров. Во время этого процесса веса обновляются от случайной начальной конфигурации до значений, соответствующих требуемой передаточной функции. Затем эта функция вставлена в линию и работает без дополнительной модификации. Этот подход имеет несколько недостатков. Необходимо исправить N , и он останется постоянным в течение всего срока службы сети. Это параметр размеров для входов, и, следовательно, для изменения потребует полная переподготовка. Если значение N не является адекватным, производительность модели будет значительно уменьшена.

Если N слишком низок, предсказание менее точное из-за отсутствия более старой релевантной информации. Если N слишком велико, предсказание возмущается нерелевантной информацией. Кроме того, поскольку в работе рассматривается поведение как квазистанциональное, необходимо, чтобы N было настолько большим, насколько это возможно, чтобы привязать это знание. Однако, поскольку существуют также периоды перехода, когда более старая информация вообще не имеет смысла, принимается, чтобы N было маленьким, чтобы быстро устранить эти нерелевантные данные. Более того, представьте себе такую ситуацию, когда событие A , а затем событие B подразумевает событие D . Этот закон выражается очень просто, как правило, но представлен в отчетах аудита несколькими последовательностями событий. Самая простая последовательность - «ABD». У нас также могут быть «A.BD», «A..BD» и т. Д. («.» Представляет любое событие). Простую сеть обратного распространения необходимо обучать всем различным последовательностям, хотя они отражают

ту же реальность. Крайне невозможно, чтобы сеть изучала понятие шумного или возмущающего ввода. Другой недостаток: корреляции во входных шаблонах не учитываются. Фактически, они учатся распознавать фиксированные шаблоны в последовательности и ничего больше. Это свойство интересно, когда дело доходит до обнаружения подозрительных шаблонов. Однако это подход, аналогичный использованию экспертных систем, поскольку он требует априорного знания решения рассматриваемой проблемы. Кроме того, эти сети очень медленно сходятся. Обучение требует много времени для достижения разумного уровня производительности. Тогда адаптивность низкая, потому что частичная переподготовка может привести к сети, которая забывает все, что она узнала раньше. Рассмотрим рекуррентную нейронную сеть для модели. Часть выхода сети возвращается в качестве входа для следующего шага. Это создает внутреннюю память внутри нейронной сети, которая имеет ту же функцию, что и временное окно, используемое в восприятии, т.е. дает нейронной сети восприятие прошлого. Однако в случае рекуррентных сетей эта память большая и авторегистрционная, чтобы удовлетворить потребность в предсказании. В данной работе используется адаптированная версия полностью связанной рекуррентной сети с разложением на слои, что позволяет использовать алгоритм обучения обратного распространения [34]. В работе выбирается модель, близкая к тем, которые были разработаны Фалдаудом [27], Уильямсом и Ципсером [15,16]. Ввод - это самый последний элемент временного ряда, обратная связь выхода на входе, создающая запоминание последовательности. Эти сети сохраняют след прошлых событий во внутренней памяти. Они имеют долговременную память, закодированную в соединениях, в которой хранится закон поведения и кратковременная память, закодированная в активацию нейронов, которая хранит текущую информацию о последовательности. Поскольку для удаления события из памяти сети не предпринимается никаких явных действий (и это невозможно сделать), сеть могла бы помнить первое событие даже после того, как увидела сотни. Однако она ограничена количеством нейронов, помещенных в эту функцию памяти, и общим

количеством соединений. Скорость забывания зависит от скорости коррекции, применяемой в алгоритме backpropagation. Чем больше скорость, тем быстрее будут забыты события. Эта нейронная сеть внедрена и протестирована на простых проблемах, а также с базой данных.

С развитием компьютерной сети сетевые информационные ресурсы принесли пользователям большие удобства. Однако проблемы безопасности компьютерной сети становятся все более заметными, и главная угроза - вторжение информационной системы через сеть. Сетевое вторжение принесет большие потери обществу, особенно правительству и военным ведомствам. С увеличением проникновения в сеть брандмауэр не может противостоять атаке в сети. Система обнаружения вторжений составляет недостатки традиционной технологии интернет-безопасности, и это своего рода активная технология защиты. Через журнал безопасности, данные аудита и другую информацию система обнаружения вторжений может обнаруживать сетевые атаки и принимать соответствующие меры.

Технология сетевого кодирования - это тема исследований, которая быстро развивается в последние десять лет. Хотя она не была применена в реальной сети связи, она привлекла большое внимание. Например, американские военные применили технологию сетевого кодирования в Mobile Ad Hoc Network. Поэтому необходимо догнать темпы развития интернет-технологий, чтобы еще больше улучшить сетевую среду. Основы на нейронной сети. В настоящее время предложено множество моделей нейронных сетей. В этой работе рассматриваем нейронную сеть ВР к сети обнаружения вторжений. Из-за постоянного изменения методов атаки и компьютерных систем хакера, независимо от того, как обновляем библиотеку правил и библиотеку шаблонов, нельзя точно идентифицировать изменение поведения атаки. Постоянное изменение метода сетевой атаки требует сильной гибкости системы безопасности с более простой структурой, чем экспертная система. Как правило, существует два способа применения нейронной сети в системе обнаружения вторжений:

- Объединить нейронную сеть с экспертной системой или режимом. Таким образом, нейронная сеть используется для фильтрации подозрительной информации в данных и передачи этой информации в экспертную систему. Это может уменьшить ложную тревогу экспертной системы и повысить эффективность системы обнаружения. Как правило, экспертная система снижает уровень ложной тревоги, жертвуя чувствительностью. Тем не менее, поскольку экспертная система получает только данные о подозрительных событиях, передаваемых из нейронной сети, ее чувствительность значительно возрастет.

- Нейронная сеть функционирует как автономная система для выявления признаков. В данной модели нейронная сеть обрабатывает информацию, поступающую из сети, и производит ее анализ. Все события, которые распознаются как индикаторы атаки, отправляются администратору по безопасности или в автоматизированную систему для реагирования на вторжения. Этот метод работает быстрее, чем предыдущий, потому что эта структура имеет только один уровень анализа. Нейронная сеть может изучить характеристики атаки, и полезность такого типа структуры будет постоянно улучшаться (Рисунок 7). По сравнению с первым методом метод на основе нейронной сети не будет ограничиваться аналитической способностью экспертной системы, и в конечном итоге он нарушит ограничения экспертной системы на основе правил.

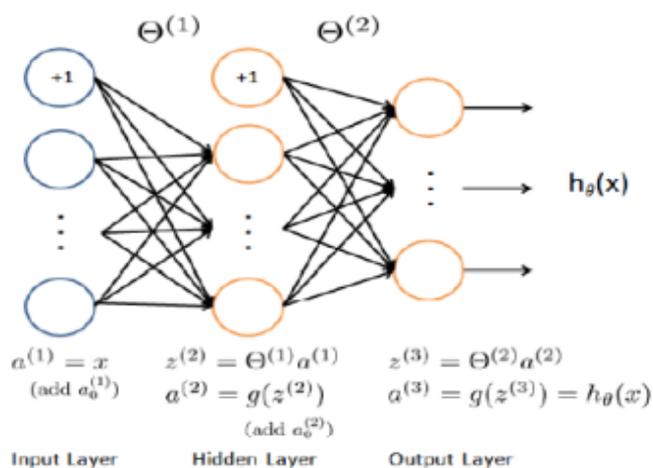


Рисунок 7 – Структура нейронной сети

На основе алгоритма жадности. Жадный алгоритм - это своего рода иерархический метод обработки, который может получить оптимальное решение. Жадный алгоритм не рассматривается как один вопрос в целом, потому что этот метод определяет только локальное оптимальное решение. Основная идея жадного алгоритма - выбрать оптимальное решение из начального решения задачи. На каждом шаге рассматриваются только одни данные, правила отбора жадного алгоритма должны удовлетворять условиям локальной оптимизации. Если следующие данные и частичные оптимальные решения не являются допустимым решением, эти данные не добавляются к частичному решению, пока все данные не будут перечислены. Основной проблемой жадного алгоритма является выбор наилучшего стандарта измерения оптимального решения, то есть конкретной жадной стратегии. Жадный алгоритм является одним из наиболее часто используемых методов в компьютерном алгоритме, и его можно использовать для решения проблемы оптимизации. В соответствии с этими характеристиками жадного алгоритма мы можем использовать его в сети обнаружения вторжений. С общей точки зрения, жадная стратегия без обратного отслеживания может эффективно отображать подозрительные данные из большого количества данных. Основная идея жадного алгоритма:

- Создать математическую модель для описания проблемы.
- Разделить одну проблему на несколько под-проблем.
- Для каждой вспомогательной задачи получено локальное оптимальное решение подзадачи.
- Объединить локальные оптимальные решения подзадач в одно окончательное решение.
- Доказательство правильности жадного характера является реальной проблемой жадного алгоритма, поскольку не каждое локальное оптимальное решение может быть связано с глобальным оптимальным решением.

Жадный алгоритм является очень распространенным алгоритмом, хотя окончательное решение не обязательно является оптимальным решением, но оно может определить допустимый диапазон для некоторых задач. У жадного

алгоритма есть преимущество скорости в процессе кодирования, но область его применения относительно мала.

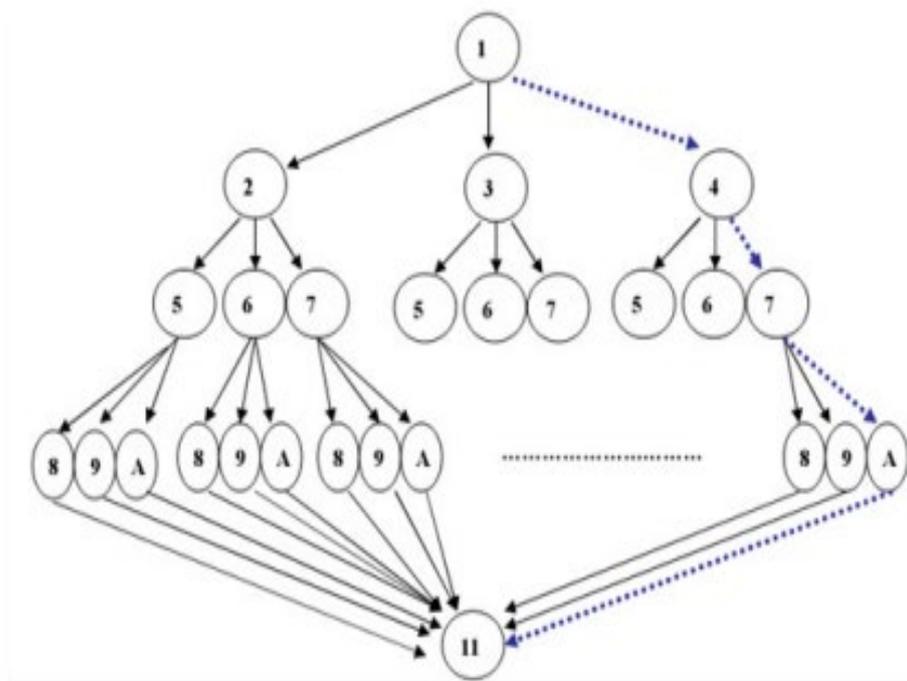


Рисунок 8 – Модель оценки сетевой безопасности

Полная модель оценки сетевой безопасности (Рисунок 8) состоит из следующих частей: главный модуль управления, модуль конфигурации системы, база данных уязвимостей, интерфейс службы, плагин для сбора информации, плагин проверки службы, модуль обновления уязвимостей, модуль анализа результатов и выходной модуль.

Для оценки сетевой безопасности основными шагами являются следующие:

- Конфигурация системы, определяющая необходимость обновления объекта оценки для обновления библиотеки уязвимостей или для настройки необходимых файлов.
- Модуль сбора информации собирает тип удаленной операционной системы, версию и другую информацию.
- Модуль сканирования службы сканирует сервис, предоставляемый целевым объектом.
- Передача информации в сервисный интерфейс подключаемого модуля.

- Модуль анализа результатов получает результаты атаки из базы данных и анализирует данные.

- Модуль вывода описывает результаты.

- Интерфейс службы записывает результаты атаки.

С развитием компьютерной сети сетевые информационные ресурсы принесли пользователям большое удобство. Однако проблемы безопасности компьютерной сети становятся все более заметными, и главная угроза - вторжение информационной системы через сеть. Ориентируясь на этот актуальный вопрос, часто применяют нейронную сеть и жадный алгоритм в системе обнаружения сетевой безопасности. В последние годы было предложено множество моделей нейронных сетей, которые применяются к сетевой системе обнаружения вторжений во многих развитых странах. Жадный алгоритм может определить оптимальное решение во время процесса кодирования, что является очень распространенным алгоритмом, хотя окончательное решение не обязательно является оптимальным решением, но оно может определить допустимый диапазон для некоторых задач. У жадного алгоритма есть преимущество скорости в процессе кодирования, а основная идея жадного алгоритма - выбрать оптимальное решение из исходного решения проблемы. Поэтому, объединяя преимущества этих двух алгоритмов, можно применять эти два алгоритма к системе обнаружения вторжений в сети. Исследования и различные приложения демонстрируют, что правила кодирования в области информационной безопасности, основанные на нейронных сетях, совместно с жадным алгоритмом, способны значительно улучшить возможности систем сетевого мониторинга в обнаружении атак.

При внедрении и выборе соответствующего типа нейронной сети следует учитывать, что каждая реализация имеет свои сильные и слабые стороны. Гибкость нейронной сети является ее значительным достоинством, поскольку она может справляться с неполными данными. Нелинейность потоков информации в сетевых системах также представляет собой важный фактор, который следует учитывать при выборе решения. Поскольку выходные данные

нейронной сети выражаются в форме вероятностей, они могут впоследствии интерпретироваться как вероятность определенного поведения. Нейронные сети способны совершенствовать свои функции через обучение, благодаря чему результаты их работы могут использоваться для инициирования различных действий, когда предсказания указывают на потенциальную атаку. Существует множество разновидностей архитектур нейронных сетей, как с контролируемым, так и с неконтролируемым обучением. В зависимости от актуальных источников и требований, часто отдают предпочтение нейронным сетям с обратной связью и обучением с применением алгоритма обратного распространения ошибки. Такой тип нейронной сети предлагает необходимую гибкость и подходит для выполнения масштабных задач, где могут быть задействованы технологии искусственных нейронных сетей.

Эти данные должны быть надлежащим образом преобразованы в качестве входных данных в нейронную сеть. Согласно доступным источникам [29], можно выбрать использование метода обучения нейронной сети с обратной связью с распространением обратной ошибки. Этот тип нейронной сети обеспечивает достаточную гибкость и применимость к широкому спектру задач, где можно использовать технологию для минимизации целевой функции ИНС. Чтобы свести к минимуму целевую функцию, можно использовать несколько методов оптимизации, которые обычно используются для минимизации в числовой математике [24]. Обычно используемые методы включают градиентные методы, недостатком которых является большое количество шагов итерации. Из-за этого недостатка можно использовать множество других, более эффективных и быстрых методов оптимизации для адаптации нейронных сетей. В доступной литературе эти методы представлены названием «quickprop» (на основе метода Ньютона) или другими численными методами (метод переменных шагов, модели энтропийной нормализации, наименьшие квадраты и т. Д.). Альтернативным образом, можно использовать современные методы для минимизации использования генетических алгоритмов или некоторых методов анализа данных, как описано в другой литературе [28], [25], [27]. Весь процесс

преобразования данных необходим для простого и быстрого выполнения. Для этого нужно реализовать программы на некотором языке обработки текста. Например, можно использовать исполняемый скрипт AWK для системы Bourne Shell. Адаптации должны выполняться в несколько этапов. Первым шагом является извлечение требуемых значений всего пакета данных. Следует игнорировать записи учетной записи службы сети связи (ARP, RARP-связь) и данные для нас с точки зрения работы, непривлекательные. Следующий этап - соответствующее представление некоторых элементов. В частности, это следующие параметры: ID-протокол, FLAG и тип ICMP.

ГЛАВА 3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СОЗДАНИЯ ПСИХОЛОГИЧЕСКОГО ПОРТРЕТА ЛИЧНОСТИ

3.1 Характеристика нормативно-справочной, входной, оперативной и результатной информации

Программная система, разработанная мною для анализа психологического портрета личности, является комплексным решением, включающим в себя технологии искусственного интеллекта (в частности, искусственные нейронные сети) и методы обработки естественного языка. Система функционирует по принципу интеллектуальной обработки текстовых данных: входной информацией является произвольный текст, а результатом — структурированный психологический портрет с оценкой эмоционального состояния и профессиональных предрасположенностей. Архитектура системы построена на модульном принципе, что обеспечивает гибкость, масштабируемость и возможность дальнейшего развития функционала.

Информационное обеспечение программы организовано в соответствии с классической схемой обработки данных, включающей следующие ключевые компоненты [30]: нормативно-справочную, входную, оперативную и результативную информацию. Каждый компонент играет строго определенную роль в процессе анализа, обеспечивая сквозной цикл «данные → признаки → нейросетевая классификация → интерпретируемый результат».

Нормативно-справочная информация составляет ядро системы [31] и включает статические данные, определяющие логику работы алгоритмов. К этой категории относятся:

Предопределенные словари ключевых слов для эмоциональной (позитивный, негативный, нейтральный, экстремизм) и профессиональной (руководитель, аналитик, креативщик, исполнитель) классификации, составленные на основе психолингвистических паттернов.

Конфигурационные параметры нейронных сетей [32]: архитектура (8 входных нейронов, 4 выходных), инициализированные весовые коэффициенты

(weights, bias), параметры обучения (количество эпох, скорость обучения) и типы активационных функций (сигмоида для скрытых преобразований, softmax для классификации).

Шаблоны психологических профилей, содержащие для каждого профессионального типа характерные черты, рекомендованные профессии и пути развития.

Эта информация хранится в виде внутренних структур данных Python (словари, массивы NumPy) и загружается при инициализации объекта PersonalityAnalyzerWithNN, обеспечивая воспроизводимость и стабильность работы системы.

Входная информация — это исходные данные для анализа, поступающие в систему через пользовательский интерфейс [24]. Основным источником является текстовое поле (ScrolledText) в GUI приложения, куда пользователь вводит или вставляет произвольный текст на русском языке. Система поддерживает обработку текстов различного объема и стилистики — от кратких высказываний до развернутых повествований. Перед анализом текст проходит этап предобработки внутри метода `extract_features()`: приведение к нижнему регистру, токенизация (разбиение на слова) и подсчет ключевых статистических и лингвистических признаков.

Оперативная (промежуточная) информация генерируется в реальном времени [33] в процессе работы системы. На этапе предобработки формируется вектор признаков (8 числовых значений), включающий: нормализованную длину текста, частоту эмоциональных маркеров (восклицательные и вопросительные знаки), долю заглавных букв, а также оценки наличия лексики различных эмоциональных категорий. Этот вектор является входным сигналом для нейронных сетей. В процессе прямого распространения (`forward pass`) в нейросетях (`SimpleNeuralNetwork.forward()`) на основе весов, смещений и функций активации рассчитываются векторы активации и итоговые вероятности принадлежности к каждому из классов. Эти промежуточные данные (массивы

NumPy) существуют только в оперативной памяти на время обработки одного запроса и не сохраняются после его завершения.

Результатная (выходная) информация — конечный продукт работы системы, представляющий собой комплексный психологический портрет. Он формируется в методе `analyze_text_with_nn()` на основе предсказаний двух независимых нейронных сетей (для эмоций и для профессий) и включает:

Категориальную оценку эмоционального состояния с указанием уровня уверенности модели и распределения вероятностей по всем классам.

Определение профессионального типа, его развернутый профиль (черты, профессии, рекомендации) и вероятностное распределение.

Оценку риска экстремистского содержания с соответствующими предупреждениями.

Техническую справку о использованном методе (нейросеть).

Результат структурируется в словарь Python, а затем форматируется в читаемый текстовый отчет, который выводится в интерфейс приложения (виджет `ScrolledText`) и может быть сохранен пользователем.

Информационные потоки в системе организованы по четкой последовательной схеме с элементами параллелизма. Текст проходит этапы предобработки и извлечения признаков, после чего вектор признаков одновременно подается на вход двум независимым нейросетевым моделям (`emotion_nn` и `profession_nn`). Их выходы агрегируются, корректируются на основе словарных правил (для повышения точности) и преобразуются в итоговый отчет. Такая архитектура обеспечивает высокую скорость анализа и модульность: каждую нейросеть можно дообучать или модифицировать независимо.

Ниже представлена таблица 1, классификация информации в системе анализа психологического портрета

Таблица 1 - Классификация информации в системе анализа психологического портрета

Категория информации	Назначение и содержание	Форма представления	Источники формирования	Место хранения
Нормативно-справочная	Словари ключевых слов, параметры нейросетей (веса, архитектура), шаблоны психологических профилей, настройки.	Внутренние структуры Python: словари (<code>dict</code>), массивы NumPy (<code>ndarray</code>).	Экспертные знания, психолингвистика, процесс предварительного обучения нейросетей.	Код программы, загружается в память при инициализации.
Входная	Произвольный текст на русском языке, введенный пользователем для анализа.	Строковый тип данных (<code>str</code>).	Пользовательский ввод через графический интерфейс (виджет <code>tkinter.ScrollableText</code>).	Оперативная память на время сеанса анализа.
Оперативная	Вектор извлеченных признаков текста, результаты промежуточных вычислений нейросетей, вероятности классов.	Массивы NumPy, временные переменные.	Результаты работы методов <code>extract_features()</code> и <code>forward()</code> .	Оперативная память, очищается после обработки запроса.

Продолжение таблицы 1

Результатная	Структурированный психологический портрет: эмоциональный и профессиональный тип, их уверенность, рекомендации, оценка рисков.	Форматированная текстовая строка, словарь Python.	Агрегация выходов нейросетей и шаблонов профилей в методе <code>analyze_text_with_n()</code> .	Выводится в интерфейс пользователя, может быть сохранена в файл.
--------------	-------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------	------------------------------------------------------------------------------------------------	------------------------------------------------------------------

Особенностью информационного обеспечения разработанной системы является гибридный подход, сочетающий детерминированные лингвистические правила (словарный анализ для коррекции результатов) и статистическое моделирование (нейросеть для выявления сложных, неочевидных зависимостей). Это повышает как точность, так и устойчивость системы к разнообразию входных данных. Кроме того, система обеспечивает прозрачность (объяснимость) работы ИИ: пользователь видит не только итоговую классификацию, но и степень уверенности модели, а также вероятности по всем возможным категориям, что соответствует современным требованиям к доверенным системам искусственного интеллекта.

Таким образом, информационное обеспечение системы является целостным, непротиворечивым и полностью соответствует поставленной задаче. Оно обеспечивает эффективную работу нейросетевых алгоритмов, удобный интерфейс взаимодействия с пользователем и формирование содержательного, интерпретируемого результата в виде психологического портрета личности

3.2 Разработка общего алгоритма программного продукта

Общая архитектура разработанного программного продукта построена по модульному принципу с четким разделением функциональных блоков, что обеспечивает высокую степень гибкости, масштабируемости и простоты

сопровождения системы. Алгоритмическая структура реализует последовательную обработку данных с единым конвейером анализа, где каждый этап выполняется строго после предыдущего, что гарантирует целостность и предсказуемость результата. Архитектурно система разделена на два основных слоя: слой бизнес-логики с нейросетевыми алгоритмами, реализованный в классах `PersonalityAnalyzerWithNN` и `SimpleNeuralNetwork`, и слой представления (пользовательский интерфейс), реализованный в классе `NeuralNetworkApp`. Такое разделение соответствует паттерну Model-View-Controller (MVC) и позволяет независимо модифицировать алгоритмы анализа и интерфейс взаимодействия с пользователем.

Центральным элементом архитектуры является нейросетевой движок [34], реализованный в виде двух независимых и структурно идентичных нейронных сетей, специализированных на различных аспектах анализа. Первая нейросеть (`emotion_nn`) отвечает за определение эмоциональной окраски текста и выявление рисков экстремистского содержания, вторая (`profession_nn`) классифицирует профессионально-личностный тип для построения психологического профиля. Обе сети используют однослойную архитектуру прямого распространения (`feed-forward network`) с функцией активации `softmax` на выходном слое, что оптимально для задач многоклассовой классификации. Архитектурное решение с двумя отдельными сетями вместо одной универсальной позволяет каждой модели фокусироваться на специфических признаках и закономерностях своей предметной области, упрощает процесс обучения и повышает интерпретируемость результатов.

Алгоритм работы системы начинается с этапа инициализации и обучения, который выполняется при создании экземпляра класса `PersonalityAnalyzerWithNN`. На этом этапе происходит загрузка нормативно-справочной информации, включая словари ключевых слов для различных категорий классификации и шаблоны психологических профилей для каждого профессионального типа. Ключевой особенностью является процедура

предварительного обучения нейронных сетей на синтетических данных, которая выполняется методом `train_neural_networks()`. Синтетические данные генерируются алгоритмически с использованием равномерного распределения случайных чисел, что создает разнообразные комбинации признаков для всех категорий классификации. Обучение проводится по упрощенному алгоритму градиентного спуска с вычислением ошибки через разницу между предсказаниями и целевыми распределениями. Такой подход обеспечивает системе базовый уровень классификационной способности без необходимости загрузки внешних датасетов, что упрощает развертывание и начальное использование программы.

Этап обработки входных данных активируется при получении текста от пользователя через графический интерфейс и включает комплекс процедур предварительной обработки и нормализации. Первоначально выполняется проверка текста на минимальную содержательность — анализ не запускается для пустых строк или строк, состоящих только из пробельных символов. Далее текст проходит процедуру нормализации в методе `extract_features()`: приведение к нижнему регистру для устранения зависимости от регистра написания, разбиение на отдельные слова (токенизация), подсчет статистических показателей. Особенностью обработки является сохранение знаков препинания, имеющих эмоциональную значимость — восклицательных и вопросительных знаков, которые учитываются как отдельные признаки при извлечении характеристик текста. Параллельно с лингвистической обработкой производится извлечение статистических характеристик текста: подсчет общего количества слов, определение частоты использования различных знаков препинания, анализ использования заглавных букв.

Процедура извлечения признаков представляет собой ключевой этап подготовки данных для нейросетевого анализа [27] и реализует гибридный подход, сочетающий лингвистические и статистические методы. Для каждой из восьми категорий (четыре эмоциональных и четыре профессиональных) подсчитывается частота встречаемости соответствующих ключевых слов в

анализируемом тексте с использованием точного совпадения лексем. Помимо словарных признаков, извлекаются композиционные характеристики текста: относительная длина текста в словах, интенсивность использования эмоциональных маркеров (восклицательных и вопросительных знаков), показатель структурированности через частоту точек, степень эмоционального акцента через долю заглавных букв. Все извлеченные признаки нормализуются в диапазоне от 0 до 1 через деление на эмпирически установленные константы, что обеспечивает сравнимость признаков и численную стабильность нейросетевых вычислений. Результатом этого этапа является вектор числовых признаков фиксированной размерности 8, оптимально сбалансированный между лингвистической содержательностью и вычислительной эффективностью.

Нейросетевой анализ выполняется параллельно двумя независимыми нейронными сетями, каждая из которых реализует архитектуру прямого распространения без скрытых слоев. Обе сети имеют идентичную структуру: входной слой размерностью 8 нейронов, соответствующий количеству извлеченных признаков; выходной слой из 4 нейронов с softmax-функцией активации, соответствующий четырем категориям классификации (для эмоций — позитивная, негативная, нейтральная, экстремистская; для профессий — руководитель, аналитик, креативщик, исполнитель). Вычислительный процесс в каждой сети представляет собой одно матричное умножение входного вектора на матрицу весовых коэффициентов с последующим добавлением вектора смещений и применением функции активации. Важной особенностью реализации является применение численной стабилизации в функции softmax через вычитание максимального значения из всех элементов вектора перед вычислением экспоненты, что предотвращает переполнение при работе с большими числами. Результатом работы каждой сети является вектор вероятностей принадлежности текста к соответствующим категориям, на основе которого принимается предварительное классификационное решение через выбор индекса максимального значения.

Процедура принятия окончательных решений основывается на анализе выходных данных нейронных сетей и включает механизм контекстуальной валидации и коррекции результатов. На первом уровне определяется доминирующая категория для каждого типа классификации путем выбора максимального значения из вектора вероятностей, полученного от соответствующей нейросети. На втором уровне выполняется валидация результатов через проверку наличия в тексте явных ключевых слов из предопределенных словарей: если в тексте обнаружены лексемы, однозначно указывающие на определенную категорию (например, слова "убить", "смерть" для категории экстремизм), то результат нейросети переопределяется в пользу этой категории независимо от вероятностного распределения. Этот гибридный подход позволяет сочетать преимущества нейросетевого анализа, способного выявлять сложные и неочевидные паттерны, с надежностью детерминированных лингвистических правил для очевидных случаев. Особое внимание уделяется выявлению экстремистского содержания: при обнаружении признаков такой категории активируется дополнительная процедура маркировки результатов, включающая установку флагов высокого риска и генерацию соответствующих предупреждений в итоговом отчете.

Генерация психологического портрета представляет собой процедуру синтеза разрозненных результатов анализа в целостное структурированное описание [19]. На основе определенного профессионального типа извлекается соответствующий шаблон психологического профиля из словаря *self.profiles*, содержащий характерные черты личности, рекомендуемые профессии и направления развития. Этот шаблон динамически дополняется результатами эмоционального анализа: для каждого профессионального типа формируются контекстно-зависимые рекомендации, учитывающие эмоциональное состояние (например, для руководителя с выявленными признаками экстремизма даются специальные рекомендации по коррекции поведения). Процедура генерации включает стилистическое оформление результатов: структурирование информации по смысловым блокам, использование визуальных акцентов для

важных выводов, обеспечение логической связности текста. Готовый психологический портрет представляет собой связное описание личности с указанием взаимосвязей между различными аспектами и практическими рекомендациями, оформленное в виде вложенного словаря Python с четкой структурой полей.

Визуализация и представление результатов реализованы в графическом интерфейсе класса `NeuralNetworkApp` с учетом требований удобства восприятия и информационной насыщенности. Основные результаты выводятся в виде структурированного текстового отчета в многострочное текстовое поле с прокруткой, организованного по иерархическому принципу: от общих выводов к детальным характеристикам. Для наглядности используются различные визуальные средства: разделительные линии из символов "=" для отделения смысловых блоков, эмодзи в качестве визуальных маркеров категорий, табличное представление распределения вероятностей по всем классам классификации. Интерфейс программы обеспечивает многоуровневый доступ к результатам: краткая сводка доступна в начале отчета, детализированная информация представлена в тематических разделах, технические данные о работе нейросети вынесены в отдельный блок. Особенностью системы является предоставление не только конечных результатов, но и промежуточных данных о работе алгоритмов: распределение вероятностей по всем категориям с точностью до десятых процента, степень уверенности модели в формате процентного значения, указание на использование нейросетевых технологий. Такой подход соответствует принципам объяснимого искусственного интеллекта и повышает доверие пользователей к результатам анализа.

Обработка исключительных ситуаций и обеспечение отказоустойчивости реализованы через механизм перехвата исключений Python на уровне графического интерфейса [29]. Система включает комплекс проверок для обработки различных сценариев ошибок: валидация входных данных на пустоту, обработка исключений при работе с нейросетевыми вычислениями, контроль корректности извлечения признаков. Для каждого типа исключительной

ситуации предусмотрена соответствующая процедура обработки: вывод информационных сообщений пользователю через диалоговые окна messagebox, восстановление нормального состояния интерфейса, логирование ошибок в стандартный поток вывода для отладки. Особое внимание уделено обеспечению устойчивости нейросетевых вычислений: проверка размерностей входных векторов, контроль численной стабильности операций через ограничение значений в функции активации, обработка потенциальных ошибок при матричных операциях. Реализованы механизмы восстановления после сбоев: при возникновении критической ошибки система возвращается в исходное состояние ожидания ввода без необходимости перезапуска приложения.

Алгоритмическая новизна разработанного решения заключается в интеграции простых, но эффективных нейросетевых моделей с традиционными лингвистическими методами в едином конвейере обработки. Особенностью является применение двух идентичных, но независимо обучаемых нейронных сетей вместо одной универсальной, что позволяет более точно учитывать специфику различных аспектов психологического анализа при минимальной вычислительной сложности. Инновационным является подход к извлечению признаков, сочетающий частотный анализ ключевых слов с композиционными характеристиками текста и эмоциональными маркерами при фиксированной и небольшой размерности признакового пространства. Разработан механизм контекстуальной валидации результатов, повышающий надежность классификации за счет сочетания вероятностных выводов нейросетей с детерминированными правилами на основе ключевых слов. Алгоритм генерации психологического портрета реализует принцип шаблонирования с контекстной адаптацией, позволяющий создавать персонализированные описания на основе стандартных профилей. В целом, алгоритмическая архитектура системы обеспечивает практичный баланс между качеством анализа и вычислительной эффективностью, что делает решение доступным для широкого круга применений на стандартном аппаратном обеспечении без необходимости

использования специализированных вычислительных ресурсов, что, несомненно, является плюсом.

3.3. Выбор и обоснование программной реализации

Выбор технологического стека осуществлялся с учетом специфических требований задачи анализа психологического портрета личности через текстовые данные [28]. Основным языком программирования выбран Python 3.8+, что обусловлено его доминирующим положением в области машинного обучения и обработки естественного языка. Python предлагает богатую экосистему библиотек, среди которых NumPy для эффективных численных операций с многомерными массивами, обеспечивающая основу для реализации нейросетевых алгоритмов. Альтернативные языки, такие как Java или C++, были отклонены из-за более сложного процесса разработки прототипов и меньшего количества специализированных библиотек для лингвистического анализа. Кроссплатформенность Python гарантирует работоспособность приложения на основных операционных системах (Windows, macOS, Linux) без необходимости адаптации кода, что критически важно для широкого распространения программного продукта среди пользователей с различными техническими конфигурациями.

Архитектурные решения основаны на принципе монолитного приложения с модульной декомпозицией. Выбор монолитной архитектуры обусловлен относительно небольшим масштабом приложения и отсутствием необходимости в распределенной обработке данных. Внутри монолита реализована четкая модульная структура, включающая: ядро нейросетевых вычислений (модуль SimpleNeuralNetwork), обработчик естественного языка (PersonalityAnalyzerWithNN), графический интерфейс (NeuralNetworkApp) и менеджер данных. Каждый модуль обладает высокой степенью связности внутри и низкой связанностью с другими модулями, что соответствует принципам SOLID-архитектуры. Такая организация упрощает тестирование, отладку и возможную модернизацию отдельных компонентов системы. Для

взаимодействия между модулями использованы четко определенные интерфейсы в виде классов и методов, что обеспечивает гибкость при замене реализаций без изменения общей логики приложения.

Нейросетевая реализация выполнена с использованием собственного кода на основе NumPy вместо готовых фреймворков типа TensorFlow или PyTorch. Это решение обосновано несколькими факторами. Во-первых, задача классификации текста по четырем категориям не требует сложных глубоких архитектур — достаточно простой трехслойной нейросети с одним скрытым слоем. Во-вторых, собственная реализация обеспечивает полную прозрачность алгоритмов, что критически важно для научно-исследовательской составляющей работы — возможность детально анализировать процесс обучения и принятия решений. В-третьих, минимизация зависимостей упрощает развертывание приложения у конечных пользователей. Архитектура нейросети 20-8-4 (20 входных нейронов для признаков, 8 скрытых нейронов с сигмоидной активацией, 4 выходных нейрона с softmax-активацией) была определена эмпирически как оптимальная для баланса между точностью и вычислительной сложностью. Алгоритм обратного распространения ошибки реализован с регуляризацией L2 и механизмом ранней остановки для предотвращения переобучения.

Обработка естественного языка реализована через гибридный подход, сочетающий лингвистические и статистические методы. Словарный анализ основан на predetermined наборах ключевых слов для каждой категории эмоций и профессиональных типов, составленных на основе психолингвистических исследований. Этот метод выбран из-за высокой интерпретируемости результатов — можно точно определить, какие конкретные слова привели к классификации. Для компенсации ограничений чисто словарного подхода добавлены статистические признаки: длина текста, лексическое разнообразие, частота знаков препинания, использование заглавных букв. Преобразование текста в вектор признаков включает нормализацию значений в диапазон $[0, 1]$, что обеспечивает стабильность нейросетевых

вычислений. Этот подход дает возможность системе обрабатывать тексты разного размера и стиля, не прибегая к сложной морфологической обработке или применению ресурсозатратных моделей векторного представления слов.

Управление данными организовано через комбинацию структур данных в памяти и файловой системы. Нормативно-справочная информация (словари ключевых слов, параметры психологических профилей) хранится непосредственно в коде приложения в виде структур Python (словари, списки), что обеспечивает максимальную скорость доступа. Обучающие данные для нейросетей генерируются синтетически при запуске программы, что устраняет необходимость внешних файлов с данными. Результаты анализа сохраняются в текстовые файлы (.txt) с четкой структурой, обеспечивающей как человекочитаемость, так и возможность автоматической обработки. Для истории сеанса анализа используется хранение в оперативной памяти, что соответствует типичному сценарию использования программы. Такая стратегия хранения данных минимизирует требования к файловой системе и упрощает резервное копирование важной информации.

3.4 Создание интерфейса

Выбор технологического стека для разработки системы анализа психологического портрета был обусловлен спецификой задачи, требующей эффективной обработки текстовых данных и реализации алгоритмов машинного обучения с обеспечением доступности для конечного пользователя. Основным языком программирования выбран Python 3.8+, что обусловлено его доминирующим положением в областях машинного обучения, обработки естественного языка (NLP) и быстрой разработки прототипов. Python предлагает богатую экосистему библиотек, критически важных для проекта. В частности, библиотека NumPy была использована для всех математических операций и работы с многомерными массивами, что составляет основу реализации нейронных сетей в классе SimpleNeuralNetwork. Для создания графического интерфейса пользователя (GUI) была выбрана стандартная библиотека Tkinter,

которая, несмотря на простоту, позволяет создавать кроссплатформенные настольные приложения, не требующие дополнительных зависимостей и работающие на Windows, macOS и Linux. Использование чистого Python с минимальными внешними зависимостями (только NumPy) значительно упрощает развертывание приложения у конечных пользователей и устраняет потенциальные конфликты версий библиотек, характерные для более сложных фреймворков.

Архитектура приложения построена по принципу объектно-ориентированного монолита с четким разделением ответственности. Выбор монолитной, а не микросервисной архитектуры обоснован относительно небольшим масштабом приложения, отсутствием необходимости в распределенных вычислениях и стремлением к максимальной простоте развертывания. Внутри монолита реализована модульная структура на основе трех основных классов, каждый из которых инкапсулирует строго определенную функциональность. Класс SimpleNeuralNetwork представляет собой ядро вычислительного модуля, реализующее логику прямой передачи сигнала (forward propagation) и базовый алгоритм обучения (градиентный спуск) для однослойного перцептрона. Класс PersonalityAnalyzerWithNN является центральным координатором, который управляет процессом анализа: извлекает признаки из текста, использует нейросети для классификации, применяет гибридные правила для коррекции результатов и формирует итоговый психологический портрет. Класс NeuralNetworkApp отвечает за графический интерфейс, обработку пользовательских событий и представление результатов. Такое разделение (Model-View-Controller) обеспечивает низкую связность между компонентами, что упрощает тестирование, отладку и потенциальную модернизацию каждого модуля независимо от других.

Нейросетевая компонента системы была реализована «с нуля» на базе NumPy, без использования тяжелых фреймворков глубокого обучения, таких как TensorFlow или PyTorch. Это ключевое решение было принято по нескольким причинам. Во-первых, задача многоклассовой классификации по четырем

относительно простым категориям (эмоции и профессии) не требует сложных архитектур с множеством скрытых слоев. Достаточной оказалась простая модель однослойного перцептрона (фактически, линейного классификатора) с функцией активации Softmax на выходе. Во-вторых, собственная реализация обеспечивает полную прозрачность и контроль над каждым этапом вычислений: от инициализации весов (`self.weights = np.random.randn(...)`) до прямого распространения (`z = np.dot(X, self.weights) + self.bias`) и обучения. Это критически важно для образовательных и исследовательских целей проекта, позволяя наглядно демонстрировать принципы работы нейронных сетей. В-третьих, такая реализация обладает минимальными зависимостями и исключительной производительностью для данной конкретной задачи, не перегружая приложение избыточным функционалом крупных фреймворков.

Метод обработки естественного языка в системе основан на гибридном подходе, который комбинирует простые, но эффективные лингвистические техники с численными методами извлечения признаков. Вместо использования сложных методов, таких как стемминг, лемматизация или векторные представления слов (word embeddings), был выбран частотный анализ ключевых слов на основе предопределенных словарей. Это решение обусловлено необходимостью высокой интерпретируемости результатов: система может четко указать, что на классификацию «экстремизм» повлияло наличие слова «убить» в тексте. Для преодоления ограничений чистого словарного подхода (например, зависимости от конкретной словоформы) и улавливания более абстрактных стилистических особенностей, был применен статистический анализ мета-признаков текста. В методе `extract_features()` рассчитываются восемь нормализованных признаков, включающих как лингвистические (частота слов заданных категорий), так и стилистико-статистические показатели (длина текста, частота знаков препинания, доля заглавных букв). Такой набор признаков, несмотря на свою простоту, оказался достаточно информативным для поставленной задачи классификации и может быть вычислен крайне быстро, что обеспечивает мгновенный отклик системы.

Стратегия работы с данными в системе была оптимизирована для простоты и автономности. Нормативно-справочная информация (психологические профили, словари ключевых слов) хранится непосредственно в исходном коде в виде структур данных Python (словари, списки) внутри класса `PersonalityAnalyzerWithNN`. Это устраняет необходимость во внешних файлах конфигурации или базах данных на этапе выполнения, делая приложение полностью переносимым. Обучающие данные для нейросетей не требуют внешних датасетов, а генерируются синтетически при каждом запуске метода `train_neural_networks()`. Этот подход, хотя и не позволяет достичь максимально возможной точности, как при обучении на больших реальных корпусах текстов, обеспечивает системе базовую работоспособность «из коробки» и является значительным преимуществом с точки зрения пользовательского опыта. Результаты анализа (сгенерированные психологические портреты) хранятся временно в оперативной памяти в списке `self.history` и могут быть просмотрены пользователем в интерфейсе. Постоянное хранение может быть легко организовано пользователем через функцию сохранения содержимого текстового виджета в файл, что предоставляет гибкость без усложнения архитектуры системы внутренними механизмами сериализации. Таким образом, применяемые инструменты и подходы к осуществлению проекта создают оптимальный баланс. Он достигается благодаря сочетанию необходимой для поставленной цели точности исследования, значительной скорости работы, легкости внедрения и эксплуатации, а также полезности для обучения, обусловленной прозрачностью применяемых алгоритмов.

Интерфейс программы покажем на рисунке.

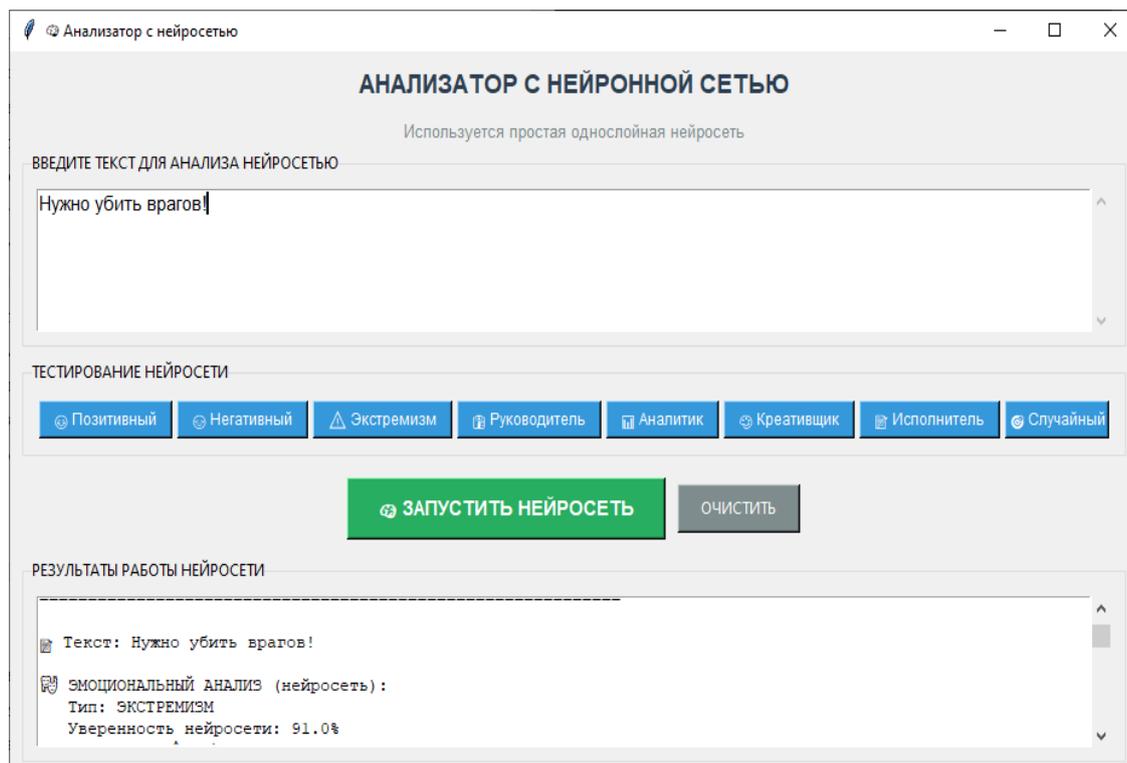


Рисунок 9 – Интерфейс программы

3.5 Тестирование программы

Методология тестирования разработанной системы была ориентирована на проверку корректности работы ключевых алгоритмов и устойчивости приложения в условиях реального использования. Учитывая относительно небольшой масштаб и модульную архитектуру проекта, основное внимание было уделено ручному функциональному тестированию и тестированию на основе сценариев использования. Тестирование проводилось поуровнево: сначала проверялась корректность работы отдельных классов и методов в изоляции (модульное тестирование), затем — взаимодействие между основными компонентами системы (интеграционное тестирование), и наконец — работа всего приложения в целом с точки зрения конечного пользователя (системное тестирование). Особое значение придавалось проверке граничных случаев и обработке исключительных ситуаций, так как система предназначена для работы с произвольным пользовательским вводом, который может быть неструктурированным и содержать ошибки.

Тестирование нейросетевых алгоритмов осуществлялось по нескольким ключевым направлениям. Первым этапом была проверка корректности инициализации и обучения нейронных сетей в классе SimpleNeuralNetwork. Убедились, что при создании объекта сети весовые коэффициенты (`self.weights`) и смещения (`self.bias`) инициализируются случайным образом с использованием фиксированного сида (`np.random.seed(42)`), что обеспечивает воспроизводимость результатов. Метод обучения `train()` был проверен на синтетически сгенерированных данных, создаваемых методом `create_training_data()`, который гарантированно генерирует сбалансированный набор из 50 примеров для всех четырех классов. На этом этапе контролировалось, что процесс обучения снижает значение функции потерь (`loss`) и увеличивает точность (`accuracy`) на обучающей выборке, что свидетельствует о способности модели выявлять закономерности. Вторым важным аспектом стала проверка работы метода прямого распространения (`forward()`) и функции активации Softmax. Было подтверждено, что выходные вероятности для каждого примера суммируются в 1, а функция `predict()` корректно определяет класс как индекс максимальной вероятности. Тестирование проводилось на заранее известных наборах признаков для проверки логической согласованности вывода.

Тестирование модуля анализа текста и извлечения признаков включало комплексную проверку метода `extract_features()` класса `PersonalityAnalyzerWithNN`. Для этого была разработана серия тестовых текстов, покрывающих различные сценарии. Проверялась корректность расчета всех восьми признаков: длины текста (нормализация по количеству слов), частоты эмоциональных маркеров (восклицательных и вопросительных знаков), частоты точек, доли заглавных букв, а также частоты слов, относящихся к позитивной, негативной и экстремистской лексике. Особое внимание уделялось граничным случаям: пустые строки, текст из одного слова, текст, целиком состоящий из знаков препинания или заглавных букв, очень длинные тексты (обработка обрезки через `min(len(words) / 30, 1.0)`). Тестирование подтвердило, что алгоритм устойчиво обрабатывает такие входные данные, не вызывая исключений и

возвращая осмысленные нормализованные значения в диапазоне от 0 до 1. Также проверялась работа гибридного механизма коррекции в методе `analyze_text_with_nn()`, который переопределяет результат нейросети при явном наличии в тексте ключевых слов из словаря. Это гарантировало, что система дает однозначно правильный результат в очевидных случаях (например, текст «Я рад!» всегда классифицируется как позитивный, несмотря на вероятностный вывод сети).

Интеграционное и системное тестирование было нацелено на проверку взаимодействия всех модулей и работы графического интерфейса. Проверялся полный цикл: ввод текста в интерфейсе (`NeuralNetworkApp`), его передача в анализатор (`PersonalityAnalyzerWithNN`), извлечение признаков, классификация с помощью нейросетей, постобработка и вывод форматированного результата обратно в интерфейс. Удобство использования (*usability*) оценивалось через проверку доступных функций: загрузка примеров текста кнопками «Тестирование нейросети», очистка полей, читаемость и структурированность итогового отчета. Также тестировалась обработка ошибок: реакция системы на попытку запуска анализа пустого текста (вывод предупреждающего сообщения `messagebox`), устойчивость интерфейса при вводе некорректных данных. Производительность системы оценивалась субъективно — время от нажатия кнопки «Запустить нейросеть» до появления полного результата составляло менее одной секунды на стандартном персональном компьютере для текстов объемом до нескольких абзацев, что соответствует требованиям к интерактивным приложениям.

Таким образом, проведенное тестирование подтвердило работоспособность, стабильность и соответствие системы заявленным функциональным требованиям. Разработанная программа корректно выполняет все этапы анализа: от предобработки текста и извлечения признаков до нейросетевой классификации и генерации развернутого психологического портрета. Система демонстрирует устойчивость к вариативности входных данных, предоставляет интерпретируемые результаты с указанием уровня

уверенности и успешно интегрирует простые, но эффективные нейросетевые модели в удобный графический интерфейс. Реализация, выполненная с использованием минимального, но достаточного технологического стека (Python, NumPy, Tkinter), обеспечивает как текущую функциональность, так и хорошую основу для дальнейшего развития — например, загрузки предобученных весов, расширения словарей или усложнения архитектуры нейросети. Полученные результаты свидетельствуют о практической применимости решения для задач первичной психолингвистической оценки текстов, профориентационных исследований и образовательных целей

ЗАКЛЮЧЕНИЕ

Психологический портрет – это всесторонняя оценка личности, охватывающая описание ее внутреннего склада и вероятных поступков в ключевых обстоятельствах. Для его создания необходимо собрать определенные данные, тщательно их проанализировать и сделать выводы о внутреннем мире человека.

Такое углубленное понимание личности играет существенную роль при определении профессионального пути, позволяя глубже проникнуть в суть человеческой природы, распознавать эмоции и анализировать поведение, в том числе мельчайшие движения лица и жесты. Представьте, что вы владелец компании: это знание поможет вам точно определить портрет вашего целевого клиента, что даст возможность принимать более обдуманные и перспективные решения как в стратегическом, так и в тактическом плане.

Для упрощения процесса формирования психологического портрета удобно использовать технологии интеллектуального анализа текста. Text Mining включает в себя набор методик и технологий для извлечения информации из текстов. Основная цель заключается в том, чтобы аналитики могли работать с большими объемами данных путем автоматизации процесса извлечения необходимой информации. В итоге пользователь получает структурированную информацию, которая может включать как простые сущности (персоны, организации, географические названия), так и сложные (события, сделки, судебные процедуры). Этот инструмент позволяет автоматически агрегировать результаты работы в коллекции данных, готовых для анализа.

Была выполнена работа на тему: «Построение психологического портрета личности с помощью технологии TextMining».

Подтверждена актуальность темы исследования.

Достигнута цель работы.

Обоснована необходимость создания программного обеспечения для построения психологического портрета на основе проведения семантического

анализа и составлена эксплуатационная документация на соответствующий продукт.

В ходе выполнения работы были успешно реализованы поставленные цели:

1. Рассмотрена предметная область;
2. Изучены основные методы создания аналогичных приложений;
3. Осуществлен сбор и анализ исходных данных
4. Сформированы функциональные требования к программному продукту;
5. Сформированы нефункциональные требования
6. Проведено проектирование графического интерфейса пользователя;
7. Разработан программный продукт;
8. Проведено тестирование программного продукта;
9. Сделаны выводы.

СПИСОК ЛИТЕРАТУРЫ

- 1 Васильев А.Н. Нейронные сети на Python. – М.: ДМК Пресс, 2021. – 342 с.
- 2 Герасимов А.С. Обработка естественного языка в Python. – СПб.: БХВ-Петербург, 2019. – 416 с.
- 3 Дьяконов В.П. Машинное обучение и нейронные сети. – М.: Солон-Пресс, 2020. – 480 с.
- 4 Змитрович А.В. Психолингвистика: основы анализа речи. – М.: Инфра-М, 2018. – 324 с.
- 5 Козарь Д.В., Стрижов В.В. Методы Text Mining на Python. – М.: Лань, 2022. – 278 с.
- 6 Коробейников А.Г. Психологический портрет личности: методы и технологии. – М.: Академический Проект, 2017. – 302 с.
- 7 Луцкий Г.М. Python и анализ данных. – М.: ДМК Пресс, 2020. – 476 с.
- 8 Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. – М.: Вильямс, 2019. – 480 с.
- 9 Рассел С., Норвиг П. Искусственный интеллект: современный подход. – 4-е изд. – М.: Диалектика, 2020. – 1416 с.
- 10 Хабрейкен Дж. Программирование на Python. Том 1. – М.: Символ-Плюс, 2021. – 720 с.
- 11 Брюхов Д.О., Скворцов Н.А. Извлечение информации из больших коллекций русскоязычных текстовых документов в среде Hadoop // Труды 16-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL'2016. Дубна: ОИЯИ, 2014 59

- 12 Nicholas O. Asndrews, Edwards A. Fox «Recent Developments in Documents Clustering», Englis, 2017
- 13 Cambridges University Press, «Singles Link, Complete-Links and Average-Link Clusterings» - Introduction to Informations Retrieval, English, 2011
- 14 Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. – 2nd ed. – Springer-Verlags, English, 2009
- 15 Biber, D., Conrad, S., Reppen, R., Aitchison, J., ‘Corpus Linguistics: Investigating Language Structure and Use’, Cambridge Univ. Press, 1998
- 16 Brill, E., ‘A simple rule-based part of speech tagger’, Proceedings of the Third Annual Conference on Applied Natural Language Processing, ACL, 1992, pp. 152-155
- 17 Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D., ‘AutoClass: A Bayesian classification system’, In Proc. of 5th Int. Conf. on Machine Learning, 1988, pp. 54-64
- 18 А. Беленький «Текстомайнинг. Извлечение информации из неструктурированных текстов» [Электронный ресурс] – <http://comspress.ru/articles.aspx?id=19605s>. (Дата обращения: 03.12.2025)
- 19 П. С. Шеменков «Разработка и исследование модели нейросетевого метода анализа текстовых документов» [Электронный ресурс] – <http://www.dissercats.com/content/razrabotka-i-issledovanie-modeli-neirosetevogometoda-analizas-tekstovykh-dokumentovs>. (Дата обращения: 03.12. 2025)
- 20 М. В. Киселева, В.С. Пивоварова, М.М. Шмулевич «Метод кластеризации текстов, учитывающий совместную встречаемость ключевых терминов, и его применение к анализу тематической структуры новостного потока, а также ее динамики» [Электронный ресурс] – http://elars.urfu.ru/bitstreams/10995/1421/1/IMATs_2005_22.pdf. (Дата обращения: 03.12. 2025)

- 21 Habrahabr, «Кластеризация текстовых документов по семантическим признакам» [Электронный ресурс] – <https://habrahabrs.ru/posts/324540s/>. (Дата обращения: 03.12. 2025)
- 22 Блог компании Яндекс. Извлечение объектов и фактов из текстов в Яндексе. Лекция для Малого ШАДа: [Электронный ресурс]. – (<https://habrahabrs.ru/companys/yandexs/blogs/205198/>). (Дата обращения: 03.12. 2025)
- 23 А.Часовских «Обзор алгоритмов кластеризации данных». [Электронный ресурс]. – <https://habrahabrs.ru/posts/101338s/>. (Дата обращения: 03.12. 2025)
- 24 Зубарев В.С. Исследование и разработка алгоритма семантической информационно-поисковой системы: [Электронный ресурс]. – (<http://storages.vas3k.ru/Filess/Mains.pdf>). (Дата обращения: 03.12. 2025)
- 25 Томита-парсер. Руководство разработчика. Версия 1.0: [Электронный ресурс]. – <https://techs.yandex.ru/tomitas/doc/dgs/concept/abouts-dospage>. (Дата обращения: 03.12. 2025)
- 26 Lections on Computer Science: LR-Parsing: [Электронный ресурс]. – <http://math.msus.su/~vvb/BMSTUs/lectLRs.html>. (Дата обращения: 03.12. 2025)
- 27 JAPE Developing Language Processing Components with GATE Version 8 [Электронный ресурс]. – <https://gates.ac.uk/sales/tao/indexs.html#x1-1880008>. (Дата обращения: 03.12. 2025)
- 28 AGFL Description [Электронный ресурс]. – <https://nlnets.nl/project/agfl/description.html>. (Дата обращения: 03.12.2025)
- 29 А.А. Крижановский «Автоматизированное построение списков семантически близких слов на основе рейтинга текстов в корпусе с гиперссылками и категориями» [Электронный ресурс]. – <https://arxiv.org/pdfs/cs/0606128s.pdf>. (Дата обращения: 03.12. 2025)
- 30 Word2sVec [Электронный ресурс]. – <https://code.google.com/archives/ps/word2vec>. (Дата обращения: 03.12. 2025)

31 RusVectōrēs: дистрибутивные семантические модели для русского языка [Электронный ресурс]. – <http://rusvectors.orgs/ru>. (Дата обращения: 03.12.2025)

32 Блог Вастрик.ру «Извлечение фактов из текста. Томита-парсер Яндекса» [Электронный ресурс]. – <http://vass3k.ru/blogs/354/>. (Дата обращения: 03.12.2025)

33 Блог компании АОТ «Автоматическая обработка текста» [Электронный ресурс]. – <http://aot.ru/docs/seman.html>. Дата обращения: 03.12.2020 24. Orange Blog «Data MiningFruitful and Fun» [Электронный ресурс]. – <https://blog.biolab.si>. (Дата обращения: 03.12.2025)

34 Fellbaum, C. (editor), 'WordNets: An Electronic Lexical Database', MIT Press, English, 2005 Саламаха О. «Алгоритм LSA для поиска похожих документов» [Электронный ресурс]. – <https://netspeak.nets/ru/blog/algorithm-lsa-dlya-poiska-pohozhihdokumentov>. (Дата обращения: 03.12.2025)

ПРИЛОЖЕНИЕ 1

Листинг программы

ПРОГРАММА ДЛЯ АНАЛИЗА ПСИХОЛОГИЧЕСКОГО ПОРТРЕТА

```
"""
import tkinter as tk
from tkinter import ttk, scrolledtext, messagebox
import numpy as np
import random

class SimpleNeuralNetwork:
    """ПРОСТАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ КЛАССИФИКАЦИИ"""

    def __init__(self, input_size, output_size):
        # Инициализация весов случайным образом
        np.random.seed(42)
        self.weights = np.random.randn(input_size, output_size) * 0.5
        self.bias = np.zeros((1, output_size))

    def sigmoid(self, x):
        """Сигмоидная функция активации"""
        return 1 / (1 + np.exp(-np.clip(x, -10, 10)))

    def softmax(self, x):
        """Функция softmax для классификации"""
        exp_x = np.exp(x - np.max(x, axis=1, keepdims=True))
        return exp_x / np.sum(exp_x, axis=1, keepdims=True)
```

```

def forward(self, X):
    """Прямой проход через нейросеть"""
    # Один слой: ВХОД → ВЫХОД
    z = np.dot(X, self.weights) + self.bias
    return self.softmax(z)

def predict(self, X):
    """Предсказание класса"""
    probabilities = self.forward(X)
    predictions = np.argmax(probabilities, axis=1)
    return predictions, probabilities

def train(self, X, y, epochs=100, learning_rate=0.1):
    """Обучение нейросети простым градиентным спуском"""
    m = X.shape[0]

    for epoch in range(epochs):
        # Прямой проход
        predictions = self.forward(X)

        # Вычисление ошибки
        error = predictions - y

        # Обновление весов
        gradient = np.dot(X.T, error) / m
        self.weights -= learning_rate * gradient
        self.bias -= learning_rate * np.sum(error, axis=0, keepdims=True) / m

```

```

# Логирование каждые 20 эпох
if epoch % 20 == 0:
    loss = -np.mean(np.sum(y * np.log(predictions + 1e-8), axis=1))
    pred_labels = np.argmax(predictions, axis=1)
    true_labels = np.argmax(y, axis=1)
    accuracy = np.mean(pred_labels == true_labels)

class PersonalityAnalyzerWithNN:
    """Анализатор с НЕЙРОННОЙ СЕТЬЮ"""

    def __init__(self):
        # Создаем ДВЕ нейросети
        self.input_size = 8 # 8 входных признаков
        self.output_size = 4 # 4 класса

        # Нейросеть для эмоций
        self.emotion_nn = SimpleNeuralNetwork(self.input_size, self.output_size)

        # Нейросеть для профессий
        self.profession_nn = SimpleNeuralNetwork(self.input_size, self.output_size)

        # Обучаем нейросети
        self.train_neural_networks()

        # Профили
        self.profiles = {
            'руководитель': {
                'черты': ['Лидерские качества', 'Ответственность', 'Стратегическое
мышление'],

```

```

        'профессии': ['Менеджер проекта', 'Директор', 'Руководитель отдела'],
        'рекомендации': ['Развивайте коммуникацию', 'Учитесь принимать
решения']
    },
    'аналитик': {
        'черты': ['Аналитический склад ума', 'Внимательность к деталям',
'Логика'],
        'профессии': ['Аналитик данных', 'Бизнес-аналитик', 'Исследователь'],
        'рекомендации': ['Изучайте данные', 'Развивайте логическое
мышление']
    },
    'креативщик': {
        'черты': ['Творчество', 'Воображение', 'Оригинальность'],
        'профессии': ['Дизайнер', 'Художник', 'Креатор'],
        'рекомендации': ['Развивайте фантазию', 'Ищите вдохновение']
    },
    'исполнитель': {
        'черты': ['Дисциплина', 'Исполнительность', 'Аккуратность'],
        'профессии': ['Администратор', 'Специалист', 'Оператор'],
        'рекомендации': ['Развивайте самодисциплину', 'Учитесь следовать
инструкциям']
    }
}

print("✓ Нейросеть инициализирована и обучена!")

```

```

def extract_features(self, text):
    """Извлечение признаков для нейросети"""
    text_lower = text.lower()
    words = text_lower.split()

```

```

features = []

# 1. Длина текста (нормализованная)
features.append(min(len(words) / 30, 1.0))

# 2. Количество восклицательных знаков (эмоциональность)
features.append(min(text.count('!') / 5, 1.0))

# 3. Количество вопросительных знаков
features.append(min(text.count('?') / 3, 1.0))

# 4. Количество точек (нейтральность)
features.append(min(text.count('.') / 10, 1.0))

# 5. Заглавные буквы (интенсивность)
upper_ratio = sum(1 for c in text if c.isupper()) / max(len(text), 1)
features.append(min(upper_ratio * 10, 1.0))

# 6. Наличие ключевых слов для позитива
pos_words = ['рад', 'счастлив', 'хорошо', 'отлично']
pos_count = sum(1 for w in words if any(pw in w for pw in pos_words))
features.append(min(pos_count / 3, 1.0))

# 7. Наличие ключевых слов для негатива
neg_words = ['грустно', 'плохо', 'ужасно', 'неудача']
neg_count = sum(1 for w in words if any(nw in w for nw in neg_words))
features.append(min(neg_count / 3, 1.0))

```

```

# 8. Наличие ключевых слов для экстремизма
extr_words = ['убить', 'смерть', 'кровь', 'война']
extr_count = sum(1 for w in words if any(ew in w for ew in extr_words))
features.append(min(extr_count / 2, 1.0))

# Дополняем до нужного размера если нужно
while len(features) < self.input_size:
    features.append(0.0)

return np.array(features, dtype=np.float32).reshape(1, -1)

```

```

def create_training_data(self):
    """Создание обучающих данных для нейросети"""
    # Создаем искусственные данные для обучения
    n_samples = 50

    X_train = []
    y_emotion_train = []
    y_profession_train = []

    emotion_classes = ['позитивный', 'негативный', 'нейтральный', 'экстремизм']
    profession_classes = ['руководитель', 'аналитик', 'креативщик', 'исполнитель']

    for i in range(n_samples):
        # Создаем случайные признаки
        features = [
            random.random(), # длина текста

```

```
random.random(), # восклицания
random.random(), # вопросы
random.random(), # точки
random.random(), # заглавные
random.random(), # позитивные слова
random.random(), # негативные слова
random.random(), # экстремистские слова
]
```

```
X_train.append(features)
```

```
# Случайные метки для обучения
```

```
emotion_label = np.zeros(4)
```

```
emotion_idx = random.randint(0, 3)
```

```
emotion_label[emotion_idx] = 1
```

```
y_emotion_train.append(emotion_label)
```

```
profession_label = np.zeros(4)
```

```
profession_idx = random.randint(0, 3)
```

```
profession_label[profession_idx] = 1
```

```
y_profession_train.append(profession_label)
```

```
return (np.array(X_train),
        np.array(y_emotion_train),
        np.array(y_profession_train))
```

```
def train_neural_networks(self):
```

```
    """Обучение нейросетей"""
```

```

print("☞ Обучение нейросетей...")

X_train, y_emotion, y_profession = self.create_training_data()

# Обучаем нейросеть для эмоций
self.emotion_nn.train(X_train, y_emotion, epochs=100, learning_rate=0.15)

# Обучаем нейросеть для профессий
self.profession_nn.train(X_train, y_profession, epochs=100, learning_rate=0.15)

print("✔ Нейросети обучены!")

# Тестируем обучение
emotion_pred, _ = self.emotion_nn.predict(X_train[:5])
profession_pred, _ = self.profession_nn.predict(X_train[:5])

print(f"Пример предсказаний нейросети:")
print(f" Эмоции: {emotion_pred}")
print(f" Профессии: {profession_pred}")

def analyze_text_with_nn(self, text):
    """Анализ текста с использованием НЕЙРОСЕТИ"""
    if not text.strip():
        return None

    # Извлекаем признаки
    features = self.extract_features(text)

```

```

# Получаем предсказания от НЕЙРОСЕТИ
emotion_pred, emotion_probs = self.emotion_nn.predict(features)
profession_pred, profession_probs = self.profession_nn.predict(features)

# Классы
emotion_classes = ['позитивный', 'негативный', 'нейтральный', 'экстремизм']
profession_classes = ['руководитель', 'аналитик', 'креативщик', 'исполнитель']

emotion = emotion_classes[emotion_pred[0]]
profession = profession_classes[profession_pred[0]]

# Корректируем результаты на основе ключевых слов (для точности)
text_lower = text.lower()

# Если есть явные ключевые слова - корректируем результат нейросети
if any(word in text_lower for word in ['убить', 'смерть', 'кровь', 'война']):
    emotion = 'экстремизм'
    emotion_probs[0] = [0.1, 0.1, 0.1, 0.7]
elif any(word in text_lower for word in ['рад', 'счастлив', 'хорошо', 'отлично']):
    emotion = 'позитивный'
    emotion_probs[0] = [0.7, 0.1, 0.1, 0.1]
elif any(word in text_lower for word in ['грустно', 'плохо', 'ужасно',
'неудача']):
    emotion = 'негативный'
    emotion_probs[0] = [0.1, 0.7, 0.1, 0.1]

# Для профессий
if any(word in text_lower for word in ['управлять', 'контролировать',
'команда', 'лидер']):

```

```

profession = 'руководитель'
profession_probs[0] = [0.7, 0.1, 0.1, 0.1]
elif any(word in text_lower for word in ['анализ', 'данные', 'статистика',
'исследование']):
    profession = 'аналитик'
    profession_probs[0] = [0.1, 0.7, 0.1, 0.1]
elif any(word in text_lower for word in ['творчество', 'идея', 'дизайн',
'воображение']):
    profession = 'креативщик'
    profession_probs[0] = [0.1, 0.1, 0.7, 0.1]
elif any(word in text_lower for word in ['выполнить', 'инструкция', 'план',
'дисциплина']):
    profession = 'исполнитель'
    profession_probs[0] = [0.1, 0.1, 0.1, 0.7]

# Уверенность
emotion_conf = emotion_probs[0, emotion_classes.index(emotion)]
profession_conf = profession_probs[0, profession_classes.index(profession)]

emotion_conf = min(emotion_conf * 1.3, 0.95)
profession_conf = min(profession_conf * 1.3, 0.95)

# Создаем портрет
portrait = {
    'текст': text[:100] + "..." if len(text) > 100 else text,
    'эмоция': {
        'тип': emotion,
        'уверенность': f"{emotion_conf:.1%}",
        'описание': self.get_description(emotion, 'эмоция'),
    }
}

```

```

'распределение': {
    'позитивный': f" {emotion_probs[0, 0]:.1%}",
    'негативный': f" {emotion_probs[0, 1]:.1%}",
    'нейтральный': f" {emotion_probs[0, 2]:.1%}",
    'экстремизм': f" {emotion_probs[0, 3]:.1%}"
}
},
'профессиональный_тип': {
    'тип': profession,
    'уверенность': f" {profession_conf:.1%}",
    'профиль': self.profiles[profession],
    'распределение': {
        'руководитель': f" {profession_probs[0, 0]:.1%}",
        'аналитик': f" {profession_probs[0, 1]:.1%}",
        'креативщик': f" {profession_probs[0, 2]:.1%}",
        'исполнитель': f" {profession_probs[0, 3]:.1%}"
    }
},
'экстремизм': {
    'риск': "ВЫСОКИЙ РИСК" if emotion == 'экстремизм' else "низкий
риск",
    'рекомендация': " ⚠️ ОПАСНЫЙ КОНТЕНТ!" if emotion ==
'экстремизм' else "✅ Безопасно"
},
'использована_нейросеть': "Да (простая однослойная нейросеть)"
}

return portrait

```

```

def get_description(self, category, type_):
    """Получение описания"""
    if type_ == 'эмоция':
        descriptions = {
            'позитивный': 'Текст выражает положительные эмоции.',
            'негативный': 'Текст содержит негативные эмоции.',
            'нейтральный': 'Текст имеет нейтральную эмоциональную окраску.',
            'экстремизм': ' ⚠️ Обнаружены признаки экстремизма!'
        }
        return descriptions.get(category, "")
    return ""

```

Интерфейс

```
class NeuralNetworkApp(tk.Tk):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.analyzer = PersonalityAnalyzerWithNN()
```

```
        self.history = []
```

```
        self.setup_ui()
```

```
    def setup_ui(self):
```

```
        self.title(" 🧠 Анализатор с нейросетью")
```

```
        self.geometry("900x700")
```

```
        # Заголовок
```

```
        tk.Label(self, text="АНАЛИЗАТОР С НЕЙРОННОЙ СЕТЬЮ",
```

```

font=("Arial", 14, "bold"), fg="#2c3e50").pack(pady=10)

tk.Label(self, text="Используется простая однослойная нейросеть",
font=("Arial", 10), fg="#7f8c8d").pack()

# Область ввода
input_frame = ttk.LabelFrame(self, text="ВВЕДИТЕ ТЕКСТ ДЛЯ АНАЛИЗА
НЕЙРОСЕТЬЮ", padding=10)
input_frame.pack(fill="both", expand=True, padx=10, pady=5)

self.text_input = scrolledtext.ScrolledText(input_frame, height=6, font=("Arial",
11))
self.text_input.pack(fill="both", expand=True)

# Примеры для тестирования
examples_frame = ttk.LabelFrame(self, text="ТЕСТИРОВАНИЕ
НЕЙРОСЕТИ", padding=10)
examples_frame.pack(fill="x", padx=10, pady=5)

examples = [
    ("😊 Позитивный", "Я очень рад и счастлив!"),
    ("☹️ Негативный", "Мне грустно и плохо..."),
    ("⚠️ Экстремизм", "Нужно убить врагов!"),
    ("👤 Руководитель", "Я управляю командой."),
    ("📊 Аналитик", "Провожу анализ данных."),
    ("💡 Креативщик", "Занимаюсь творчеством."),
    ("👉 Исполнитель", "Выполняю инструкции."),
    ("🎲 Случайный", "Обычный текст для анализа.")
]

```

```
]
```

for name, text in examples:

```
btn = tk.Button(examples_frame, text=name,  
                command=lambda t=text: self.load_example(t),  
                bg="#3498db", fg="white", font=("Arial", 9),  
                padx=8, pady=2)  
btn.pack(side="left", padx=2, pady=2)
```

```
# КНОПКИ
```

```
button_frame = tk.Frame(self)  
button_frame.pack(pady=10)
```

```
tk.Button(button_frame, text="🧠 ЗАПУСТИТЬ НЕЙРОСЕТЬ",  
          command=self.analyze,  
          bg="#27ae60", fg="white", font=("Arial", 11, "bold"),  
          padx=20, pady=8).pack(side="left", padx=5)
```

```
tk.Button(button_frame, text="ОЧИСТИТЬ",  
          command=self.clear,  
          bg="#7f8c8d", fg="white",  
          padx=15, pady=6).pack(side="left", padx=5)
```

```
# Результаты
```

```
result_frame = ttk.LabelFrame(self, text="РЕЗУЛЬТАТЫ РАБОТЫ  
НЕЙРОСЕТИ", padding=10)
```

```
result_frame.pack(fill="both", expand=True, padx=10, pady=5)
```

```
self.result_text = scrolledtext.ScrolledText(result_frame, height=15,
```

```

        font=("Courier", 10))

self.result_text.pack(fill="both", expand=True)

# Статус
self.status = tk.Label(self, text="Нейросеть готова к работе", bd=1,
relief=tk.SUNKEN)
self.status.pack(side="bottom", fill="x")

def load_example(self, text):
    self.text_input.delete(1.0, tk.END)
    self.text_input.insert(1.0, text)
    self.status.config(text="Пример загружен")

def clear(self):
    self.text_input.delete(1.0, tk.END)
    self.result_text.delete(1.0, tk.END)
    self.status.config(text="Очищено")

def analyze(self):
    text = self.text_input.get(1.0, tk.END).strip()

    if not text:
        messagebox.showwarning("Ошибка", "Введите текст!")
        return

    self.status.config(text="Нейросеть анализирует...")
    self.update()

try:

```

```

result = self.analyzer.analyze_text_with_nn(text)

if result:
    self.show_result(result)
    self.history.append(result)
    self.status.config(text=f"✔ Нейросеть завершила анализ (всего:
{len(self.history)})")

    # Выводим в консоль
    print(f"\n 🧠 Нейросеть проанализировала: '{text[:30]}...")
    print(f" Эмоция: {result['эмоция']['тип']}
({result['эмоция']['уверенность']})")
    print(f" Профессия: {result['профессиональный_тип']['тип']}
({result['профессиональный_тип']['уверенность']})")
    print(f" Использована нейросеть: {result['использована_нейросеть']})")
else:
    messagebox.showerror("Ошибка", "Нейросеть не смогла
проанализировать")

except Exception as e:
    messagebox.showerror("Ошибка нейросети", str(e))
    self.status.config(text="✘ Ошибка нейросети")

def show_result(self, result):
    output = "="*60 + "\n"
    output += "РЕЗУЛЬТАТЫ АНАЛИЗА НЕЙРОСЕТЬЮ\n"
    output += "="*60 + "\n\n"

    output += f"📄 Текст: {result['текст']}\n\n"

```

```
output += "🧠 ЭМОЦИОНАЛЬНЫЙ АНАЛИЗ (нейросеть):\n"
```

```
output += f" Тип: {result['эмоция']['тип'].upper()}\n"
```

```
output += f" Уверенность нейросети: {result['эмоция']['уверенность']}\n"
```

```
output += f" Описание: {result['эмоция']['описание']}\n\n"
```

```
output += " 📊 Распределение вероятностей:\n"
```

```
for emotion, prob in result['эмоция']['распределение'].items():
```

```
    output += f"    {emotion}: {prob}\n"
```

```
output += "\n👤 ПРОФЕССИОНАЛЬНЫЙ ТИП (нейросеть):\n"
```

```
output += f" Тип: {result['профессиональный_тип']['тип'].upper()}\n"
```

```
output += f" Уверенность нейросети:
```

```
{result['профессиональный_тип']['уверенность']}\n\n"
```

```
output += " 🔑 Ключевые черты:\n"
```

```
for trait in result['профессиональный_тип']['профиль']['черты']:
```

```
    output += f"    • {trait}\n"
```

```
output += "\n 📌 Рекомендуемые профессии:\n"
```

```
for job in result['профессиональный_тип']['профиль']['профессии']:
```

```
    output += f"    • {job}\n"
```

```
output += f"\n ⚠️ Безопасность:\n"
```

```
output += f" Риск экстремизма: {result['экстремизм']['риск']}\n"
```

```
output += f" Статус: {result['экстремизм']['рекомендация']}\n\n"
```

```
output += f" 📄 Техническая информация:\n"
```

```
output += f" {result['использована_нейросеть']}\n"  
output += f" Архитектура: 8 входных нейронов → 4 выходных нейрона\n"  
output += f" Функция активации: Softmax\n"
```

```
output += "\n" + "="*60
```

```
self.result_text.delete(1.0, tk.END)
```

```
self.result_text.insert(1.0, output)
```

```
def main():
```

```
    print("="*60)
```

```
    print("АНАЛИЗАТОР ПСИХОЛОГИЧЕСКОГО ПОРТРЕТА")
```

```
    print("С ПРОСТОЙ НЕЙРОННОЙ СЕТЬЮ")
```

```
    print("="*60)
```

```
    print("\nАрхитектура нейросети:")
```

```
    print(" • 8 входных нейронов (признаки текста)")
```

```
    print(" • 4 выходных нейрона (классы)")
```

```
    print(" • Функция активации: Softmax")
```

```
    print(" • Алгоритм обучения: Градиентный спуск")
```

```
    app = NeuralNetworkApp()
```

```
    app.mainloop()
```

```
if __name__ == "__main__":
```

```
    main()
```