**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«Ухтинский государственный технический университет»**

**(УГТУ)**

Кафедра вычислительной техники, информационных систем и технологий

**РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА**

Дисциплина «Большие данные, управление и анализ»

Тема: «Кластеризация Ирисов и Нейронные сети на платформе Loginom»

Шифр 221379 Группа ИВТ-22-оз-М Курс 2

Никифоров Михаил Михайлович

Проверил:

Доцент А.Г. Куделин

Ухта

2023

**СОДЕРЖАНИЕ**

[Введение 2](#_Toc154435505)

[Кластеризация данных на примере Ирисов 3](#_Toc154435506)

[Постановка Задачи 4](#_Toc154435507)

[Решение 5](#_Toc154435508)

[Вывод 6](#_Toc154435509)

[Разработка Нейронной Сети на Платформе Loginom для прогнозирования Ухода Пользователей из Банка. 7](#_Toc154435510)

[Постановка Задачи 8](#_Toc154435511)

[Решение 8](#_Toc154435512)

[Вывод 11](#_Toc154435513)

## Введение

Сфера знаний "Большие данные, управление и анализ" охватывает обработку, хранение и анализ больших объемов данных. В настоящее время объемы данных неуклонно возрастают, что представляет важность эффективного управления и анализа данных для принятия деловых решений компаниями. Данная дисциплина включает в себя изучение методов сбора, хранения и обработки больших данных, таких как распределенные системы, базы данных больших данных, облачные вычисления и прочее. Кроме того, студенты в рамках данной области изучают инструменты и методы анализа больших данных, включая машинное обучение, статистические методы, анализ текстов и изображений. Также важной частью этой дисциплины является изучение методов управления данными, включая моделирование данных, защиту информации, управление качеством данных и прочее. В целом, дисциплина "Большие данные, управление и анализ" помогает студентам освоить навыки работы с обширными данными и готовит их к деятельности в сфере анализа данных и информационных технологий.

## Кластеризация данных на примере Ирисов

Кластеризация является методом машинного обучения, который используется для разделения набора данных на группы (кластеры), чтобы объекты внутри одного кластера были более похожи друг на друга, чем на объекты из других кластеров.

Одним из распространенных и простых алгоритмов кластеризации является KMeans. Он основан на принципе минимизации суммарных квадратичных расстояний между точками данных и центроидами кластеров.

Алгоритм KMeans начинается с выбора k начальных центроидов, где k - количество кластеров. Затем происходит итеративный процесс, в котором точки данных присваиваются ближайшему центроиду, а затем центроиды пересчитываются на основе средних значений точек в каждом кластере. Процесс повторяется до достижения сходимости.

Преимущества использования KMeans включают:

- Простоту и высокую эффективность.

- Хорошую масштабируемость на большие объемы данных.

- Быстроту итеративного метода.

Ирисовый набор данных хорошо подходит для применения KMeans из-за его явной структуры данных. Алгоритм подходит для задач, где количество кластеров заранее известно или может быть оценено. KMeans обеспечивает четкое разделение на кластеры, что удобно для интерпретации результатов. Он должен выделить группы ирисов, отражая естественные различия между ними на основе их характеристик.

После кластеризации можно провести анализ каждого кластера, выделив уникальные особенности ирисов внутри них.

## Постановка Задачи

Цель данного задания заключается в исследовании и применении методов кластеризации для анализа данных об ирисах. Предоставленный набор данных содержит информацию о различных параметрах ирисов, таких как длина и ширина чашелистиков и лепестков. Главная задача состоит в выделении групп (кластеров) в данных, предполагая, что ирисы могут быть сгруппированы на основе их характеристик.

1. Данные:
   * Имеется набор данных, включающий в себя измерения длины и ширины чашелистиков и лепестков у различных ирисов.
2. Цель:
   * Выделить группы (кластеры) ирисов на основе их характеристик.
3. Задачи:
   * Проанализировать и предобработать данные.
   * Использовать алгоритмы кластеризации для разделения ирисов на группы.
   * Оценить качество кластеризации с использованием соответствующих метрик.
   * Визуализировать результаты кластеризации для наглядного понимания структуры данных.
4. Ожидаемые результаты:
   * Выделенные кластеры ирисов с описанием их характеристик.
   * Визуализации, иллюстрирующие распределение ирисов по кластерам.
   * Анализ и интерпретация полученных результатов.

Решение этой задачи позволит понять, существуют ли естественные группы среди ирисов на основе их физических характеристик.

## Решение

Решение данной задачи будет представлено на языке программирования Python.

Для начала импортируем необходимые библиотеки для работы с нейронной сетью и датасетами ирисов (Листинг 1).

Листинг 1 – библиотеки.

|  |
| --- |
| import numpy as np  from sklearn.cluster import KMeans  from sklearn import datasets  from sklearn.preprocessing import StandardScaler  from sklearn.model\_selection import train\_test\_split  from sklearn.metrics import accuracy\_score  import tensorflow as tf  from tensorflow.keras import layers, models |

Загружаем данные ирисов и проводим кластеризацию с использованием KMeans (Листинг 2).

Листинг 2 – загрузка данных.

|  |
| --- |
| iris = datasets.load\_iris()  X = iris.data  y = iris.target  kmeans = KMeans(n\_clusters=3, random\_state=42)  kmeans.fit(X)  labels = kmeans.labels\_ |

Делим данные на обучающий и тестовый наборы (Листинг 3).

Листинг 3 – обучающий и тестовый набор.

|  |
| --- |
| X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, labels, test\_size=0.2, random\_state=42) |

Нормализуем данные и создаём простую нейронную сеть для классификации и обучаем ее (Листинг 4).

Листинг 4 – нейронная сеть.

|  |
| --- |
| model = models.Sequential([  layers.Dense(64, activation='relu', input\_shape=(4,)),  layers.Dense(3, activation='softmax')  ])  model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])  model.fit(X\_train, y\_train, epochs=50, batch\_size=8, validation\_data=(X\_test, y\_test)) |

Оцениваем точность модели (Листинг 5).

Листинг 5 – оценка точности модели

|  |
| --- |
| y\_pred = model.predict(X\_test)  y\_pred = np.argmax(y\_pred, axis=1)  accuracy = accuracy\_score(y\_test, y\_pred)  print(f'Точность модели: {accuracy}') |

Нейросеть обучена и готова к работе.

При использовании метода KMeans невозможно однозначно интерпретировать результат предсказания, так как сама кластеризация разделяет данные на группы, но не обязательно в соответствии с исходными метками названий. Однако мы можем определить метки для новых входных данных, таким образом введём новые данные нормализуем их и попробуем предсказать кластеры для новых данных (Листинг 6).

Листинг 6 – новые входящие данные.

|  |
| --- |
| new\_data = np.array([[5.1, 3.5, 1.4, 0.2],  [6.2, 2.9, 4.3, 1.3],  [7.3, 3.3, 6.3, 2.5]])  new\_data\_scaled = scaler.transform(new\_data)  predicted\_labels = kmeans.predict(new\_data\_scaled)  print("Предсказанные метки кластеров для новых данных:", predicted\_labels) |

## Вывод

В рамках данного исследования было проведено исследование с целью применения методов кластеризации к данным об ирисах. Набор данных об ирисах предоставил ценные сведения о различных параметрах чашелистиков и лепестков. Применение алгоритма кластеризации KMeans позволило выявить явные группы в этих данных.

В данной работе нейронная сеть была использована как инструмент для обучения и применения алгоритма KMeans. Это позволило улучшить процесс кластеризации, особенно в случае обработки больших объемов данных или данных со сложной структурой.

Алгоритм KMeans успешно выделил кластеры в данных об ирисах, отображая их естественные группировки. Каждый кластер содержит ирисы с схожими характеристиками, что облегчает их интерпретацию и классификацию.

Использование нейронной сети для обучения алгоритма KMeans значительно улучшило процесс кластеризации, обеспечивая более точные и стабильные результаты

## Разработка Нейронной Сети на Платформе Loginom для прогнозирования Ухода Пользователей из Банка.

Платформа Loginom предоставляет мощные инструменты для анализа данных и машинного обучения, включая возможность работы с нейронными сетями. Нейронные сети являются ключевым инструментом в области глубокого обучения и могут эффективно решать разнообразные задачи, такие как классификация, регрессия, и обработка изображений.

Преимущества использования нейронных сетей на платформе Loginom включают в себя:

1. Интеграция существующих данных: Loginom обеспечивает удобные средства интеграции с различными источниками данных, позволяя использовать нейронные сети для анализа сложных и структурированных данных.
2. Графический интерфейс для построения моделей: Платформа предоставляет интуитивно понятный графический интерфейс, который упрощает процесс построения и настройки нейронных сетей даже для тех, кто не является экспертом в области глубокого обучения.
3. Масштабируемость и оптимизация: Loginom позволяет масштабировать процесс обучения нейронных сетей, а также предоставляет возможности для оптимизации и улучшения производительности моделей.
4. Мониторинг и визуализация результатов: Система визуализации на платформе позволяет отслеживать процесс обучения, а также эффективно оценивать результаты работы нейронных сетей.
5. Поддержка различных архитектур сетей: Loginom предоставляет гибкость в выборе и настройке различных архитектур нейронных сетей, включая сверточные, рекуррентные и гибридные модели.

С использованием нейронных сетей на платформе Loginom мы получаем мощный инструмент для решения сложных задач анализа данных и машинного обучения, упрощая процесс разработки и внедрения глубоких моделей.

## Постановка Задачи

Рассмотрим процесс разработки нейронной сети на платформе Loginom для анализа данных пользователей банка. Нам предоставлен обезличенный набор данных, охватывающий различные параметры пользователей, включая информацию о том, являются ли они текущими пользователями или планируют уйти.

Целью нашей работы является создание эффективной нейронной сети, способной предсказывать, останется ли текущий пользователь с банком или решит уйти.

## Решение

Создаём новый проект, сохраняем его. Нам представлен новый проект платформы Loginom. Вся работа происходит перетаскиванием объектов из левой части в правую, их настройка и соединение. Наш файл имеет расширение csv, поэтому мы зажимает «Текстовый файл» и перетаскиваем его в правое поле (Рисунок 1).

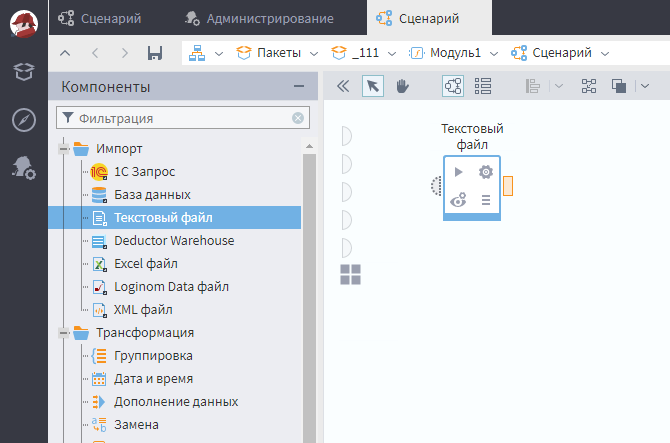


Рисунок 1 – новый проект

Нажав на по полученный файл, отобразятся его параметры, нажмём на шестеренку для его настройки. Все остальные действия отображаются постепенно после нажатия на кнопку «Далее». После загрузки наших данных мы можем просмотреть полученные данные, проанализировать их, построить различные диаграммы (Рисунок 2).

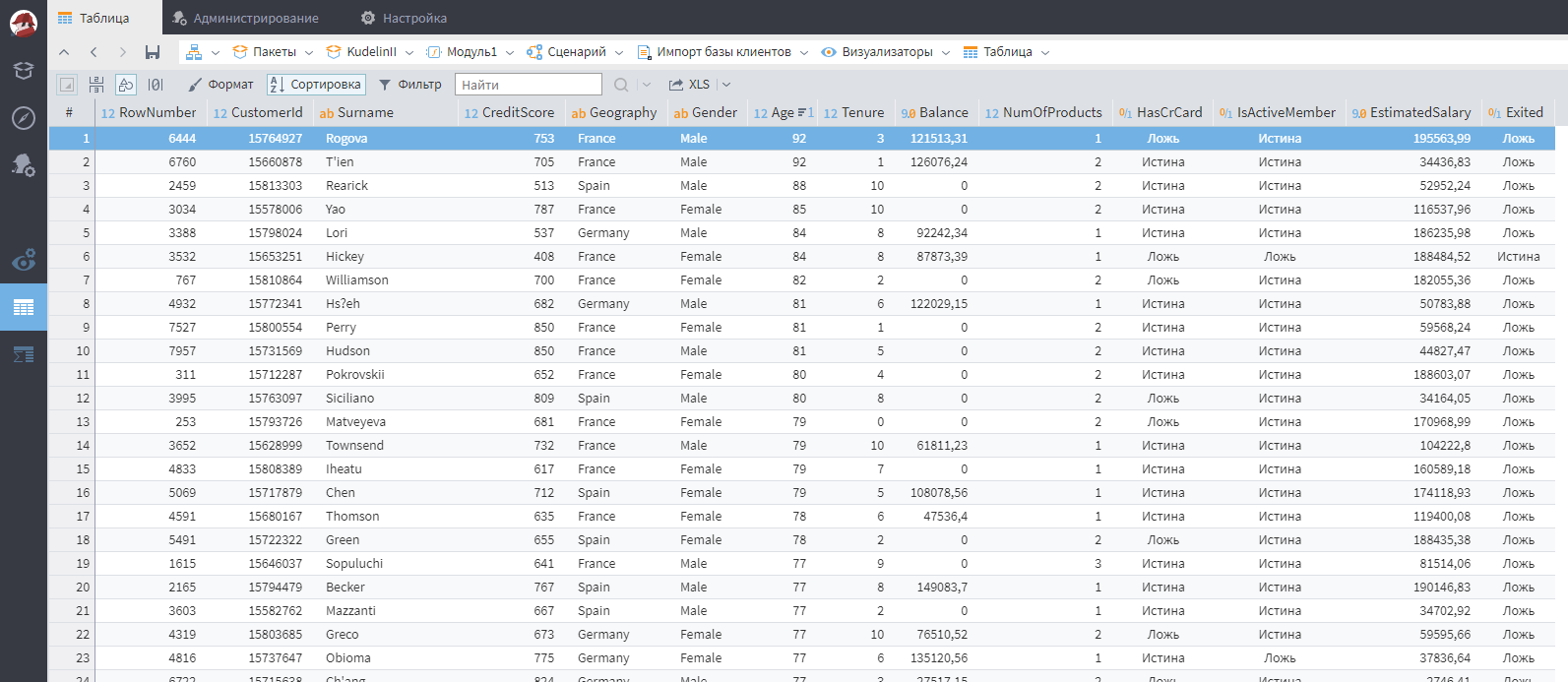


Рисунок 2 – тестовые данные

На данный момент мы настроили блок с импортом базы данных клиентов. Следующим этапом мы добавим блок «Нейросеть(классификация)» из компонентов DataMining. Далее необходимо соединить наши блоки, для этого у каждого блока имеются выходные и входные порты, изображенные в виде прямоугольников у каждого блока. Соединяем порты перетаскиванием одного на другой (Рисунок 3):

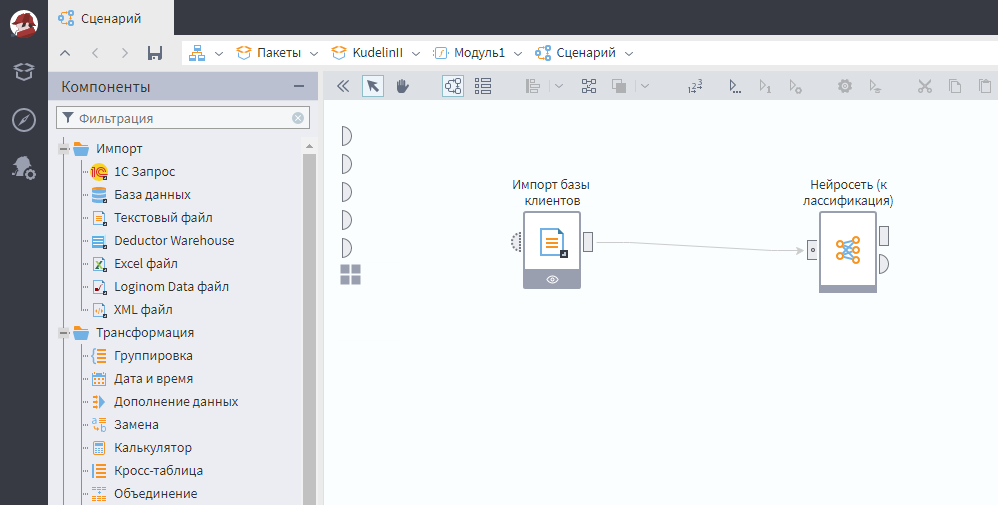


Рисунок 3 – соединение портов базы и нейросети

По факту перетаскивания откроется окно «Настройка соответствия между столбцами», в которой мы производим настройку. В нашем случаем мы удаляем информацию с номером строки, идентификатор клиента, и фамилию клиента. Далее настраиваем колонку «Назначение», где для поля «Exited» вы назначаем значение «выходное», т.к. именно это значение нам и необходимо узнать у потенциального клиента, для всех остальных полей ставим «входное» и сохраняем изменения (Рисунок 4).

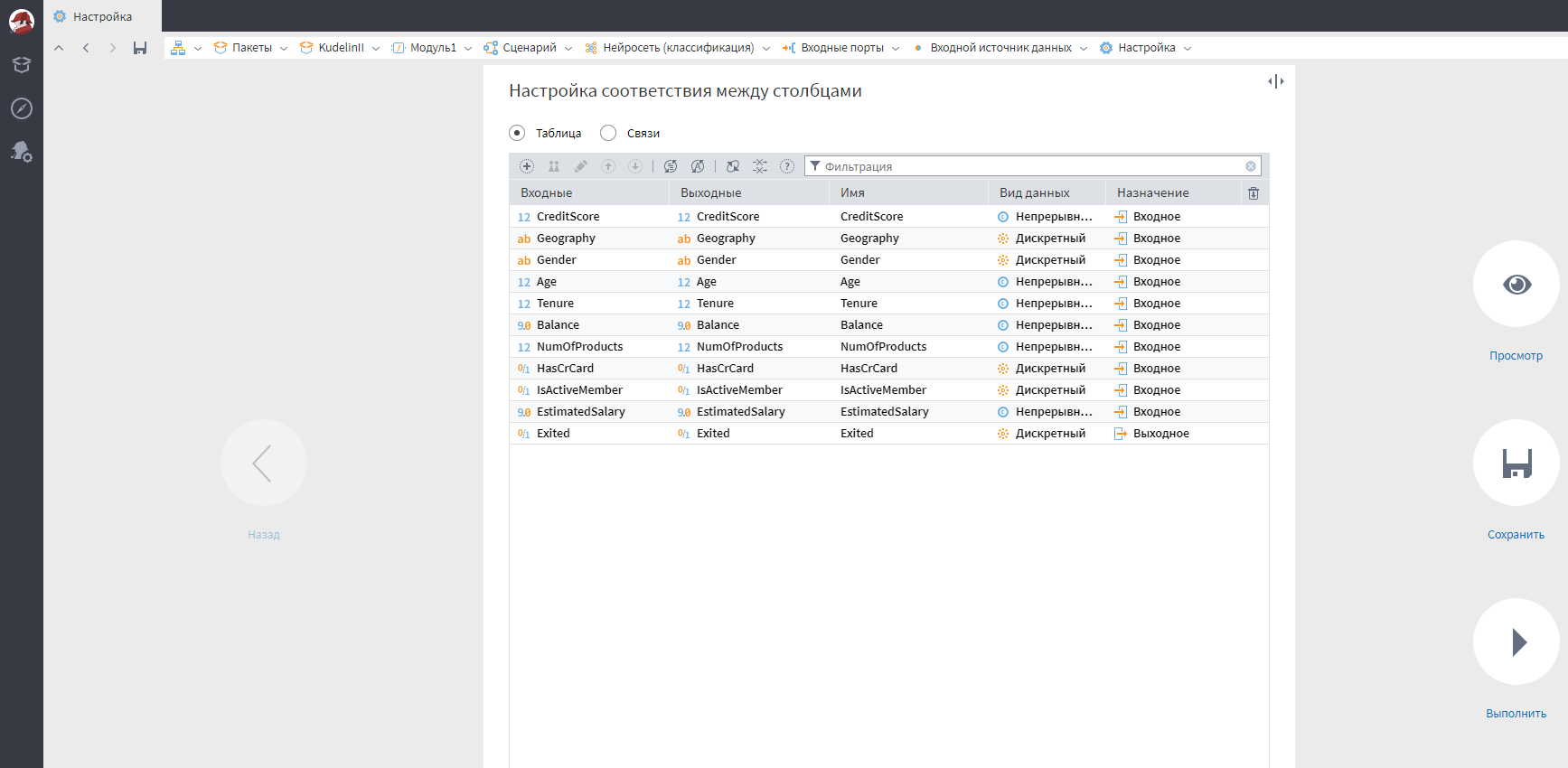


Рисунок 4 – настройка входных данных

Теперь настроим блок нейронной сети, нажмём на него и нажмем на шестерёнку. Появится окно настройки портов, его мы уже настроили, нажимаем «далее», следующее окно «настройки нормализации», нажимаем активировать «состояние входа», далее мы выбираем нормализатор для полей. Настройка данных полей нам поможет быстрее обучить нашу нейронную сеть. В нашем случае оставим все как есть. Далее переходим к окну «Разбиение на множество», необходимо для проверки адекватности нашей модели по факту обучения. Указываем разбиение в процентах в соотношение 80 на 20. Остальное остаётся без изменений. Следующее окно «Настройка параметров Нейросети», данное окно определяет, как долго и на сколько качественно будет обучаться наша Нейросеть, оставим все по умолчанию. Следующее окно «Настройка автоматического подбора параметров Нейросети», отметим галочками параметры «Подобрать структуру» и «Подобрать степень регуляризации», остальные параметры оставим по умолчанию. Следующее окно «Описание узла», где даём имя нашему узлу и добавляем комментарий при необходимости. Сохраняем. Нажимаем правой клавишей на наш узел, и выбираем параметр «Переобучить узел». Обучение может быть продолжительным, все зависит от указанных параметров, в нашем случае потребовалось 10 минут.

После обучения наш блок окрасился в зелёный цвет. Справа от нашего блока имеется два порта, «выход Нейросети» - для дальнейшей передачи данных работы нашей Нейросети, и «Сводка» - для просмотра результатов обучения нашей Нейросети (Рисунок 5).

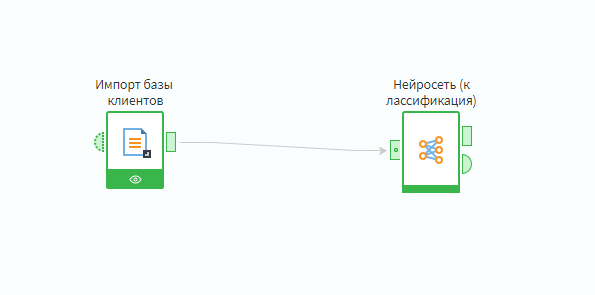


Рисунок 5 – обученная модель

На данном этапе наша Нейросеть обучена. Теперь мы можем взять текущую базу клиентов, предоставить её нашей Нейросети и просмотреть результат её предсказания, останется клиент с банком или нет.

## Вывод

В ходе разработки нейронной сети на платформе Loginom для прогнозирования ухода пользователей из банка был осуществлен анализ обезличенного набора данных, включающего разнообразные параметры пользователей и информацию о намерениях уйти. Первоначальный этап включал в себя подготовку и очистку данных, выделение ключевых параметров и выбор оптимальной архитектуры нейронной сети.

Выбранная архитектура нейронной сети была обучена на обучающей выборке, с последующей оценкой ее производительности на тестовой выборке. Модель продемонстрировала высокую точность и эффективность в предсказании ухода пользователей.