Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт
Систем искусственного интеллекта
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой

_____ _____
подпись инициалы, фамилия
«____» _____ 20 __ г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – Информационные системы и технологии
код – наименование направления

Проектирование ИТ формирования групп общения для абонентов
тема
мобильных сетей

Руководитель _____ доцент, к.т.н. К.В. Раевич
подпись, дата должность, ученая степень инициалы, фамилия

Выпускник _____ О.А. Мельситова
подпись, дата инициалы, фамилия

Красноярск 2019

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт
Систем искусственного интеллекта
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ _____
подпись инициалы, фамилия
« ___ » _____ 20 ___ г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме _____
бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту _____
фамилия, имя, отчество

Группа _____ Направление (специальность) _____
номер код

_____
наименование

Тема выпускной квалификационной работы_____
_____
_____

Утверждена приказом по университету № _____ от _____

Руководитель ВКР _____
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР _____
_____
_____

Перечень разделов ВКР_____
_____
_____
_____

Перечень графического материала _____
_____
_____
_____

Руководитель ВКР _____ _____
подпись инициалы и фамилия

Задание принял к исполнению _____
подпись, инициалы и фамилия студента

« ___ » _____ 20__ г.

РЕФЕРАТ

Работа посвящена проектированию информационной технологии формирования групп общения абонентов мобильной сети по данным, собираемым оператором, с помощью кластерного анализа. Рассмотрены разные методы кластеризации, выбран наиболее подходящий для данной задачи - метод иерархической кластеризации.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Одна из наиболее конкурентных сфер в российском бизнесе - мобильные телекоммуникации. Всего 4 федеральных оператора делят рынок, оценивающийся в 150 миллионов абонентов. Таким образом операторы "большой четверки" конкурируют за внимание и лояльность абонентов, создавая наиболее комфортные и выгодные условия для клиентов.

Телекоммуникационные организации известны сбором огромных объемов данных. Все, что делают клиенты, регистрируется: сколько времени и когда люди звонят, пики прямых сообщений, использование интернета и так далее. Это лишь некоторые из показателей, которые собираются и при грамотном анализе могут быт использованы в стратегических целях компании.

Для решения проблемы оттока абонентов используется «племенная» модель поведения клиента. Эта модель основана на том факте, что есть люди, которые имеют большое влияние на других и которые хорошо связаны с различными группами. Если один из этих клиентов переключит провайдера связи, это может вызвать эффект домино и заставить других в его сети сделать то же самое.

Большие данные, как совокупность технологий, обрабатывают большие объемы данных и должны уметь работать со структурированными и плохо структурированными данными параллельно в разных аспектах.

Появление различных алгоритмов больших данных позволило быстро и эффективно работать с этими огромными массивами: прогнозировать поведение потребителей, оптимизировать ресурсы, анализировать эффективность работы персонала, снизить риски, улучшить процессы обработки данных. Одним из таких алгоритмов является «Кластерный анализ», служащий для разбиения множества объектов определенной структуры на подмножества по некоторым комбинированным признакам. Главной его особенностью является отсутствие фиксированного набора параметров для разбиения. Разбиение происходит по совокупности признаков, таким образом, что объекты одного множества имеют примерно одинаковые характеристики.

Предлагается выполнить кластеризацию абонентов мобильной сети на основе информации о них, которая хранится у оператора. Целью такой кластеризации является получение новых знаний для поиска индивидуального подхода к каждому клиенту.

Цель: проектирование ИТ формирования групп общения для абонентов мобильных сетей.

Для достижения поставленной цели необходимо решить следующие задачи:

- Обзор и анализ методов классификации;
- Проектирование ИТ кластеризации абонентов мобильных сетей и формирование групп общения.

1 Теоретическая часть

1.1 Информация об абонентах мобильных сетей

При звонке, в логи, то есть в файлы, содержащие системную информацию работы сервера или компьютера, в которые заносятся определенные действия пользователя или программы, заносится местоположение обоих сторон. Но для вызывающей стороны местоположение адресата известно с точностью только до сети, а для самого адресата входящий вызов также логгируется вместе с его местоположением в этот момент, после чего эти логи можно сопоставить друг с другом.

В дополнение к метаданным, которые собираются со всех коммуникаций наземных абонентов, для мобильных абонентов логгируются также:

- Внутренний ID абонента у оператора (привязан к договору со всеми персональными данными);
- Номер телефона;
- IMSI сим карты;
- IMEI аппарата.
- CI - идентификатор соты, в которой находится абонент.

Местоположение регистрируются в моменты входа/выхода из сети, перемещение из одной соты в другую и по ручному запросу в любой момент. Операторам могут отслеживать местоположение непрерывно с записью всего трека, но это остается на усмотрение оператора, практика может варьироваться в зависимости от региона.

В настоящий момент не существует системы тотальной записи всех разговоров в федеральном масштабе. Прослушивание разговоров осуществляется адресно. В прошлом это было невозможно технически, в последнее время технических препятствий становится все меньше, но еще не вся аппаратура готова к такому, особенно в регионах.

Существует требование, в соответствии с которым, SMS (как и метаданные о разговорах) должны храниться минимум три года. Но, поскольку хранение SMS технически не вызывает проблем, а юридически ограничен только минимальный срок хранения, то в реальности SMS хранятся практически вечно.

1.2 BigData

Сотовые операторы получают преимущества в конкурентной гонке больше за счет ориентации на узкую часть аудитории потребителей, чем за счет снижения тарифных планов, так как операторы связи – обладатели огромного архива статистического материала о своих абонентах. Его умелая обработка может дать массу полезной информации, узнать которую иначе не представляется возможным. Так, анализ геопространственных данных дает очень точную и, главное, крайне оперативную картину жизни города в течение дня, что невозможно при использовании теоретических выкладок или данных переписи населения.

Обработать массив геопространственных данных и сделать выводы операторам позволяет технология больших данных.

Большие данные (англ. big data) — серия подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объёмов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста, распределения по многочисленным узлам вычислительной сети, сформировавшихся в конце 2000-х годов, альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

Таким образом под Big Data понимается не какой-то конкретный объём данных и даже не сами данные, а методы их обработки, которые позволяют распределено обрабатывать информацию. Эти методы можно применить как к огромным массивам данных (таким как содержание всех страниц в интернете), так и к маленьким (таким как содержимое статьи).

Последовательность работы с Big Data состоит из сбора данных, структурирования полученной информации с помощью отчетов и дашбордов, создания инсайтов и контекстов, а также формулирования рекомендаций к действию. Так как работа с Big Data подразумевает большие затраты на сбор данных, результат обработки которых заранее неизвестен, основной задачей является четкое понимание, для чего нужны данные, а не то, как много их есть в наличии. В этом случае сбор данных превращается в процесс получения исключительно нужной для решения конкретных задач информации.

Собранные операторами мобильной связи данные могут использоваться как для улучшения качества собственных сервисов и продаж, так и для решения задач сторонних бизнесов – с использованием инсайтов операторов рекламодатели могут запускать кампании с чрезвычайно точным таргетингом.

Основным преимуществом использования big data является очень подробный таргетинг баз абонентов, который позволяет использовать множество параметров как отдельно, так и вместе, например:

- Пол и возраст;
- ARPU (средний счет абонента в месяц);
- Операционная система и модель телефона;
- Наличие трафика мобильного интернета;
- Контент посещаемых сайтов;
- Пользователи социальных сетей;
- Наличие любимых номеров и получатели вызовов;
- Звонки по направлениям, городам, странам;
- Частота и направления роуминга;
- Пользование услугами Мобильного банка;
- Музыкальные предпочтения;
- Траектории предпочтения (где абонент живет/работает);
- Геотаргетинг (где абонент находится в данный момент) и многое другое…

Big data - это многообещающий комплекс подходов к обработке больших массивов данных. В мобильной связи активно используются самые разные методы

анализа данных для достижения поставленных целей. Технология больших данных представляет собой современные методы обработки огромных объемов информации для увеличения эффективности бизнеса, создания новых продуктов и повышения конкурентоспособности в целом. Работа с большими данными позволяет бизнесу достичь выгоды, недоступной для более традиционных подходов к обработке информации, благодаря точной оценке перспектив развития продуктов, эффективному распределению инвестиционных затрат и созданию принципиально другого клиентского опыта в традиционных индустриях за счет умных услуг и сервисов на основе так называемого машинного обучения.

1.3 Классификация

Классификация — одна из важнейших задач, встречающихся при анализе данных. В зависимости от постановки можно различать следующие задачи: кластеризацию (классификацию в отсутствии обучающей выборки) и классификацию (при наличии обучающей выборки), когда данные необходимо соотнести с уже известными классами.

Первая задача возникает, как правило, когда исследуются новые объекты или явления; в этом случае кластеризация позволяет выделить однородные группы объектов, и далее, планировать последующие изыскания. Классификация при наличии обучающей выборки предполагает, основываясь на уже имеющихся данных и их соответствии известным классам, определение класса для вновь поступающих данных. Распознавание текста, речи, аутентификация\авторизация по отпечатку пальца — представляют задачи классификации этого типа.

Кластеризация, как метод поиска закономерностей, предназначен для разбиения совокупности объектов на однородные группы (кластеры) или поиска существующих структур в данных. Это задача многомерной классификации данных. Внутри каждой группы должны оказаться «похожие» объекты, а объекты разных группы должны быть как можно более отличны. Главное отличие кластеризации от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма. При этом число кластеров может постулироваться заранее, или определяться в процессе кластерного анализа. В зависимости от того, каким образом определяются расстояния между кластеризуемыми объектами, результаты классификации будут различны. Если данные представляют собой наборы количественных признаков, вполне естественно рассматривать их как точки многомерного факторного пространства, а в качестве расстояния, например, использовать евклидову метрику. В случае трехмерного факторного пространства такой подход будет иметь ясную геометрическую интерпретацию и связь с реальным трехмерным миром (каждая точка данных - некоторая точка в трехмерном пространстве; расстояние между данными - это расстояние между такими точками).

Следует помнить, что количественные признаки могут иметь различную физическую интерпретацию: один признак может отождествляться с длиной, другой и массой объекта, поэтому понимание "расстояния" между элементами в этом

случае весьма условно. Однако, если выбранные признаки полно характеризуют исследуемые объекты, то даже такие "бессмысленные" расстояния (несмотря на различие единиц измерения) имеют важное свойство: элементы, имеющие близкие значения количественных признаков, как следствие, характеризуются малыми расстояниями между друг другом, вполне ожидаемо похожи друг на друга. Таким образом, расстояния, построенные даже на базе разнородных признаков, выполняют свою главную роль — характеризуют сходство объектов.

Существует около 100 разных алгоритмов кластеризации, однако, наиболее часто используемые — иерархический кластерный анализ и кластеризация методом k-средних.

Существует множество практических применений кластеризации как в информатике, так и в других областях. Вот несколько примеров применения кластеризации:

1) Анализ данных:
   - Упрощение работы с информацией;
   - Визуализация данных;
2) Извлечение и поиск информации:
   - Построение удобных классификаторов;
3) Группировка и распознавание объектов:
   - Распознавание образов;
   - Группировка объектов;

Кластерный анализ можно представить в виде следующей последовательности действий [3]:

1) Выбор множества объектов;
2) Определение множества переменных, для оценки объектов и составление векторов характеристик;
3) Нормирование векторов характеристик одним из доступных методов;
4) Определение сходства между объектами по заданной метрике;
5) Применение выбранного метода кластерного анализа для разбиения множества объектов на кластеры по из степени схожести;
6) Представление результатов анализа.

Проанализировав результаты кластеризации можно скорректировать выбранные параметры, метрику или метод кластеризации, для улучшения результатов. Проводя данные улучшения можно прийти к наилучшему результату.

1.4 Формальные определения

Для дальнейших рассуждений, введем понятия, которыми будем оперировать.

Объектом будем называть элементарный набор данных, с которым работает алгоритм кластеризации.

Для каждого объекта определяются параметры, описывающие его, которые объединяются в вектор характеристик:

$x = (x_1, x_2, ..., x_m)$, где m – размерность пространства характеристик, а x_i-отдельная характеристика объекта (качественная или количественная).

Меру сходства двух объектов d (u,v) вычисленную по заданной метрике будем называть расстоянием между объектами, где u,v – элементы множества.

1.5 Выделение вектора характеристик

Первым шагом необходимо выделить характеристики объектов, которые будут использованы в процессе кластеризации. Это могут быть как количественные (рост, вес, координаты, счетчики, …) так и качественные характеристики (цвет, статус, настроение, …).

Чаще всего работают с количественными характеристиками, так как для них применимо большое число метрик.

На большом пространстве характеристик, процесс кластеризации происходит довольно медленно, и его результаты не всегда приемлемы. Поэтому, при большой размерности пространства характеристик, нужно постараться его уменьшить, оставив наиболее важные свойства объектов.

Получившийся набор характеристик каждого объекта необходимо нормализовать, для лучших результатов. Нормализовать вектор, значит привести его к фиксированному размеру. Характеристики каждого нормализованного вектора будут лежать в фиксированном отрезке, например, [0;1] или [-1;1], в зависимости от поставленной задачи.

1.6 Меры расстояний

После выявления вектора характеристик необходимо выбрать функцию для определения степени сходства двух объекта, называемую мерой расстояний. Выбранная функция должна удовлетворять всем условиям метрики [2].

Существуют различные метрики для вычисления близости объектов. Обозначив u,v – объекты между которыми вычисляется расстояние, а d(u,v) и u_i, v_i – их координаты, опишем некоторые из существующих метрик:

1) Евклидово расстояние. Классическая метрика Евклида, являющаяся геометрическим расстоянием в многомерном пространстве.

$$d(u,v) = \sqrt{\sum_{i=1}^{m}(u_i - v_i)^2}.$$

2) Квадрат евклидова расстояния, равный евклидову расстоянию, возведенному в квадрат. Используется для увеличения веса более отдаленных друг от друга объектов.

3) Расстояние городских кварталов или манхэттенское расстояние. Вычисляется как средняя разность по координатам и чаще всего приводит к результатам аналогичным обычному расстоянию Евклида.

$$d(u,v) = \sum_{i=1}^{m}|u_i - v_i|$$

4) Степенное расстояние, применяемое при необходимости изменить вес, в большую или меньшую сторону, относящийся к размерности, для которой соответствующие объекты сильно отличаются. Степенное расстояние вычисляется по формуле, схожей с формулой расстояния Евклида:

$$d(u,v) = \sqrt[r]{\sum_{i=1}^{m}(u_i - v_i)^p}.$$

где r и p – параметры, определяемые пользователем. Параметр p отвечает за постепенное взвешивание разностей по отдельным координатам, а параметр r ответственен за прогрессивное взвешивание больших расстояний между объектами. Как было сказано раннее, при значениях обоих равны двум, данная метрика совпадает с расстоянием Евклида.

5) Расстояние Чебышева. Мера, применяемая ввиду необходимости определить два объекта как различные, при какой-то одной отличной координате. Расстояние Чебышева вычисляется по формуле

$$d(u,v) = \max|u_i - v_i|$$

От выбора метрики во многом зависят результаты кластеризации, и для различных метрик они могут существенно отличаться.

1.7 Классификация алгоритмов

1.7.1 Иерархические и плоские

Плоские алгоритмы разбивают заданное множество объектов на кластеры, строя единственное разбиение, и для получения другого разбиения, необходимо повторять процесс кластеризации с другими параметрами.

Иерархические алгоритмы, в отличии от плоских, создают не единственное разбиение, а систему вложенных разбиений на непересекающиеся кластеры. В результате выполнения этого алгоритма получается дерево разбиений, корнем которого является кластер, содержащий все множество объектов, а листьями — более мелкие кластера.

1.1.1 Четкие и нечеткие

Данная классификация определяет, может ли один объект выборки принадлежать одновременно нескольким кластерам, или он всегда принадлежит единственному кластеру.

В четких (или непересекающихся) алгоритмах каждый объект выборки принадлежит только одному кластеру, т.е. каждому объекту сопоставляется единственный номер кластера, которому он принадлежит. В нечетких (или пересекающиеся) алгоритмах каждому объекту сопоставляется набор вещественных значений, отображающих вероятность отношения данного объекта к каждому из

кластеров. Другими словами, в нечетких алгоритмах, каждый объект принадлежит всем кластерам с разной степенью

1.8 Объединение кластеров

Как было сказано раннее, некоторые алгоритмы кластеризации выполняют разбиение ступенчато, а именно на каждом шаге два наиболее близко расположенных объекта объединяются и рассматриваются я как один кластер. В связи с этим возникает необходимость введения меры расстояний между кластерами, для определения их близости.

1) Одиночная связь (Ближайший сосед).

В данном способе, расстоянием между кластерами считается расстояние между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах. Т.е. вычисляется расстояния между всеми возможными парами объектов из различных кластеров и после этого вычисляется минимальное из этих расстояний. Этот «минимум» и считается расстоянием между кластерами.

2) Полная связь (Наиболее удаленный сосед).

Расстояния между кластерами вычисляется аналогично Одиночной связи, но вместо минимального расстояния вычисляется максимальное, т.е. расстояние между наиболее удаленными соседями. Данный способ, как правило, работает довольно хорошо, если кластеры имеют форму близкую к сферической. Если же кластеры являются «цепочечными» или имеют удлиненную форму, то этот метод непригоден.

3) Невзвешенное попарное среднее.

Расстояние между двумя отличными друг от друга кластерами можно определить, как среднее расстояние между всеми парами объектов из различных кластеров. Данный метод довольно эффективен, при условии, что объекты формируют различные группы, однако он работает также хорошо и в случаях, протяженных или «цепочечного» типа кластеров.

4) Взвешенное попарное среднее.

Этот метод схож с методом невзвешенного попарного среднего, однако в нем, при вычислении расстояний учитывается размер соответствующих кластеров, (т.е. число объектов, содержащихся в них), и используется в качестве весового коэффициента. В связи с этим данный метод необходимо использовать, если ожидаются кластеры, значительно отличающиеся по размерам.

5) Невзвешенный дендроидный метод.

Для вычисления расстояния этим методом, необходимо вычислить центры тяжести каждого из кластеров, а после этого считать за расстояние между кластерами расстояние между их центрами тяжести. При вычислении центра тяжести вес каждого объекта считается равным единице.

6) Взвешенный дендроидный метод (медиана).

Этот метод идентичен предыдущему, за исключением того, что при вычислениях используются веса для учета разницы между размерами кластеров. Поэтому, если имеются или подозреваются значительные отличия в размерах кластеров, этот метод оказывается предпочтительнее предыдущего.

1.9 Алгоритмы кластеризации

1.9.1 Алгоритмы иерархической кластеризации

Алгоритмы иерархической кластеризации принято разделять на два типа: нисходящие и восходящие алгоритмы. Нисходящие, действуют по принципу «от большего к меньшему»: в начале процесса все объекты помещаются в единственный кластер, вершину дерева, после чего, на каждом шаге, один из существующих кластеров разбивается на два более мелких, пока каждый объект не будет принадлежать собственному кластеру.

Второй тип алгоритмов, восходящие, более распространен, и работает в обратную сторону, относительно первого. Сначала каждый из объектов помещается в собственный кластер, и на каждом шаге алгоритма, два ближайших кластера объединяются в один, до тех пор, пока не останется единственный кластер, содержащий всю выборку объектов.

Результатом кластеризации данным алгоритмом является дерево разбиений, называемое дендрограммой. Наиболее популярный пример использования иерархической кластеризации - классификация животных и растений.

1.9.2 Алгоритмы квадратичной ошибки

Задачу кластеризации можно интерпретировать иначе: необходимо построить оптимальное разбиение объектов на группы. При этом условие оптимальности может быть задано требованием минимизации среднеквадратической ошибки разбиения:

$$e^2(X)u_i - v_i = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - c_j \right\|^2,$$

где X-множество объектов, $x^j, x_i^{(j)}$- их координаты, c_j- центр масс кластера j (считая массу каждой точки равной единице).

Алгоритмы данной категории относятся к классу плоских алгоритмов. Метод k-средних считается наиболее популярным в этой категории, ввиду того, что алгоритм разбивает заданное множество объектов на указанное число кластеров, расположенных на как можно большем расстоянии друг от друга. Работа этого метода разбивается на несколько этапов:

1) Случайно выбрать *k* начальных «центров масс» кластеров.
2) Отнести каждый объект к кластеру с ближайшим «центром масс».
3) Пересчитать «центры масс» кластеров согласно их текущему составу.
4) Проверить критерий остановки, и в случае его не выполнения, вернуться к пункту 2.

В качестве критерия остановки работы алгоритма как правило используют

минимальное изменение среднеквадратической ошибки. Также работа алгоритма завершается, если на шаге 2 не было объектов, сменивших свой кластер.

Результат использования данного алгоритма на плоскости, близок к диаграмме Вороного. К недостаткам этого алгоритма можно отнести необходимость задавать число кластеров для разбиения.

1.9.3 Нечеткие алгоритмы

Как было сказано выше, алгоритмы нечеткой кластеризации относят каждый объект к каждому кластеру с некоторой вероятностью, в отличии от четких методов. Алгоритмов данной категории не слишком много, ввиду этого рассмотрим наиболее популярный – метод c-средних (c-means). Этапы работы алгоритма схожи с этапами метода k-средних:

1) Задать начальное нечеткое разбиение n объектов на k кластеров путем выбора матрицы принадлежности U размера $n*k$.

2) Найти значение критерия нечеткой ошибки, используя матрицу U

$$e^2(X,U) = \sum_{i=1}^{N}\sum_{j=1}^{K}\left\|x_i^{(j)} - c_j\right\|^2, \quad c_k = \sum_{i=1}^{n_j}\left\|x_i^{(j)} - c_j\right\|^2,$$

где X-множество объектов, $x^j, x_i^{(j)}$- их координаты, c_j- центр масс кластера j (считая массу каждой точки равной единице), U_{ij}- матрица принадлежности.

3) Перегруппировать объекты с целью уменьшения этого значения критерия нечеткой ошибки.

4) Возвращаться в п. 2 до тех пор, пока изменения матрицы U не станут незначительными.

Этот алгоритм стоит применять только если заранее известно число кластеров и необходимо вычислить отношение каждого объекта ко всем кластерам.

1.9.4 Алгоритмы, основанные на теории графов

Особенность графовых алгоритмов в том, что вся выборка объектов представляется в виде графа G = (V, E) , где V – множество вершин и E – множество ребер, в роли вершин которого выступают объекты выборки, а вес ребер равен расстоянию между объектами, которые они соединяют.

Преимуществами алгоритмов данной категории являются их наглядность и относительная простота реализации с возможностью внесения модернизаций, основанных на геометрических суждениях. В этой категории наиболее популярными являются алгоритм выделения связных компонент, алгоритм построения минимального покрывающего (остовного) дерева и алгоритм послойной кластеризации.

1.9.4.1 Алгоритм выделения связных компонент

Для работы данного алгоритма необходим параметр R, задающий пороговое значение для весов ребер. В процессе работы этого алгоритма постепенно удаляются все ребра, вес которых превышает пороговое значение. В результате работы получается граф, в котором остаются только ребра, соединяющие наиболее близкие объекты.

Чтобы получить кластеры, остается только подобрать значение R так, чтобы граф разделился на несколько связных компонент, которые и будут являться кластерами.

Чаще всего, для подбора параметра R пользуются построением гистограммы распределений попарных расстояний. Если кластерная структура выражена относительно хорошо, то гистограмма будет иметь два пика, один из которых соответствует внутрикластерным расстояниям, а второй – межкластерным. Параметр *R* подбирается из зоны минимума между этими пиками.

Минус этого алгоритма в довольно сложном управлении результирующим количеством кластеров.

1.9.4.2 Алгоритм минимального покрывающего дерева

Суть этого алгоритма в построении минимального покрывающего дерево, и последовательном удалении ребер с наибольшим весом. На рисунке 1 изображено минимальное покрывающее дерево, полученное для десяти объектов, являющихся точками на плоскости.

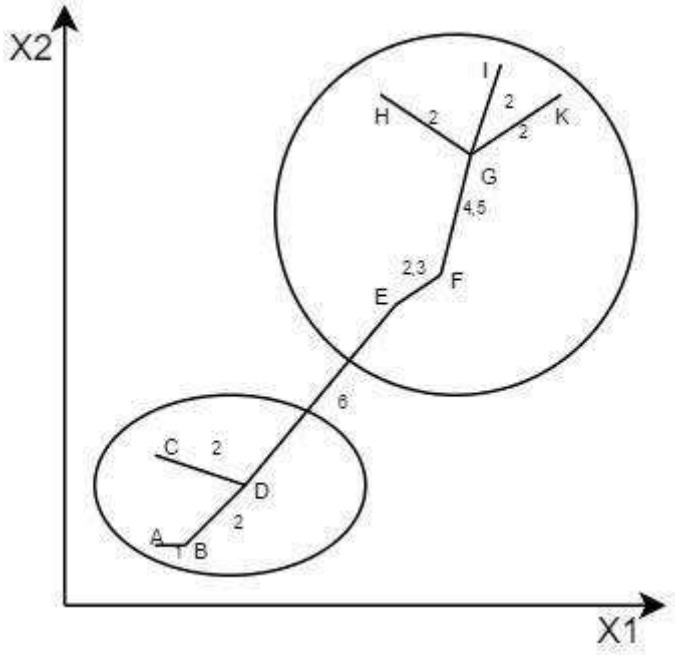

Рисунок 1 - Минимальное покрывающее дерево

Путём удаления связи с максимальным весом, помеченной DE, получаем две компоненты: $\{A, B, C, D\}$ и $\{E, F, G, H, I, K\}$. Второй кластер в дальнейшем может быть разделён ещё на два кластера путём удаления максимального ребра FG, которое имеет вес, равный 4,5 единицам.

2 Формулирование требований

2.1 Определение качественных показателей

2.1.1 Определение типа ПС для оценки

Оценка качества показателей системы выполнена с использованием ГОСТ 28195-89 «Оценка качества программных средств».

Настоящий стандарт, устанавливает общие положения по оценке качества программных средств вычислительной техники (далее - ПС), поставляемых через фонды алгоритмов и программ (ФАП), номенклатуру и применяемость показателей качества ПС.

Согласно ГОСТ 28195-89 программные средства (далее ПС) в разных группах оцениваются разными показателями, группы ПС представлены на рисунке 1.

Номер показателя по табл.1	Применяемость показателя по подклассам (группам) ПС											
	5011	5012	5013	5014	5015	5016	5017	503	504	505	506	509
1.1	+	+	+	+	+	+	+	-	±	+	±	
1.2	+	+	+	+	+	+	+	+	+	+	+	
2.1	±	±	±	±	±	±	±	±	±	±	±	
2.2	±	±	±	±	±	±	±	-	±	±	±	
2.3	±	±	±	±	±	±	±	-	±	±	±	
2.4	±	±	±	±	±	±	±	±	±	±	±	
3.1	±	±	±	+	+	+	+	±	+	±	±	
3.2	+	+	+	+	+	+	+	+	+	+	+	
3.3	+	+	±	+	+	+	+	-	+	+	±	
4.1	±	±	±	±	±	±	±	-	±	±	±	
4.2	±	±	±	±	±	±	±	±	±	±	±	
4.3	+	+	+	±	±	+	±	-	±	±	±	
5.1	-	±	-	±	±	-	-	-	+	±	±	
5.2	±	±	±	±	±	±	±	±	±	±	±	
5.3	+	+	±	±	±	±	±	-	±	±	±	
6.1	+	+	+	+	+	+	+	+	+	+	+	
6.2	+	+	+	+	+	+	+	+	+	+	+	
6.3	+	+	+	+	+	+	+	+	+	+	+	
6.4	+	+	+	+	+	+	+	+	+	+	+	

Рисунок 1 – показатели по группам ПС

2.1.2 Номенклатура критериев оценки

Номенклатура критериев оценки представлена в таблице 1.

Таблица 1 – Номенклатура критериев оценки

1. Показатели надежности ПС Характеризуют способность ПС в конкретных областях применения выполнять заданные функции в соответствии с программными документами в условиях возникновения отклонений в среде функционирования, вызванных сбоями технических средств, ошибками во входных данных, ошибками обслуживания и другими дестабилизирующими воздействиями		
Наименование групп и комплексных показателей качества	Обозначение	Характеризуемое свойство
1.1. Устойчивость функционирования	Н1	Способность обеспечивать продолжение работы программы после возникновения отклонений, вызванных сбоями технических средств, ошибками во входных данных и

		ошибками обслуживания
1.2. Работоспособность	H2	Способность программы функционировать в заданных режимах и объемах обрабатываемой информации в соответствии с программными документами при отсутствии сбоев технических средств
2. Показатели сопровождения Характеризуют технологические аспекты, обеспечивающие простоту устранения ошибок в программе и программных документах и поддержания ПС в актуальном состоянии		
2.1. Структурность	C1	Организация всех взаимосвязанных частей программы в единое целое с использованием логических структур "последовательность", "выбор", "повторение"
2.2. Простота конструкции	C2	Построение модульной структуры программы наиболее рациональным с точки зрения восприятия и понимания образом
2.3. Наглядность	C3	Наличие и представление в наиболее легко воспринимаемом виде исходных модулей ПС, полное их описание в соответствующих программных документах
2.4. Повторяемость	C4	Степень использования типовых проектных решений или компонентов, входящих в ПС

Продолжение таблицы 1

3. Показатели удобства применения Характеризуют свойства ПС, способствующие быстрому освоению, применению и эксплуатации ПС с минимальными трудозатратами с учетом характера решаемых задач и требований к квалификации обслуживающего персонала		
3.1. Легкость освоения	У1	Представление программных документов и программы в виде, способствующем пониманию логики функционирования программы в целом и ее частей
3.2. Доступность эксплуатационных программных документов	У2	Понятность, наглядность и полнота описания взаимодействия пользователя с программой в эксплуатационных программных документах
3.3. Удобство эксплуатации и обслуживания	У3	Соответствие процесса обработки данных и форм представления результатов характеру решаемых задач
4. Показатели эффективности Характеризуют степень удовлетворения потребности пользователя в обработке данных с учетом экономических, вычислительных и людских ресурсов		
4.1. Уровень автоматизации	Э1	Уровень автоматизации функций процесса обработки данных с учетом рациональности функциональной структуры программы с точки зрения взаимодействия с ней пользователя и использования вычислительных ресурсов
4.2. Временная эффективность	Э2	Способность программы выполнять заданные действия в интервал времени, отвечающий заданным требованиям
4.3. Ресурсоемкость	Э3	Минимально необходимые вычислительные ресурсы и число обслуживающего персонала для эксплуатации ПС
5. Показатели универсальности Характеризуют адаптируемость ПС к новым функциональным требованиям, возникающим вследствие изменения области применения или других условий функционирования		
5.1. Гибкость	Г1	Возможность использования ПС в различных областях применения
5.2. Мобильность	Г2	Возможность применения ПС без существенных дополнительных трудозатрат на ЭВМ аналогичного класса
5.3. Модифицируемость	Г3	Обеспечение простоты внесения необходимых изменений и доработок в программу в процессе эксплуатации

Окончание таблицы 1

6. Показатели корректности Характеризуют степень соответствия ПС требованиям, установленным в ТЗ, требованиям к обработке данных и общесистемным требованиям		
6.1. Полнота реализации	К1	Полнота реализации заданных функций ПС и достаточность их описания в программной документации
6.2. Согласованность	К2	Однозначное, непротиворечивое описание и использование тождественных объектов, функций, терминов, определений, идентификаторов и т.д. в различных частях программных документов и текста программы
6.3. Логическая корректность	К3	Функциональное и программное соответствие процесса обработки данных при выполнении задания общесистемным требованиям
6.4. Проверенность	К4	Полнота проверки возможных маршрутов выполнения программы в процессе тестирования

2.1.3Оценка показателей

Согласно Таблице 1 и рисунку 1, кластерный аналииз относится к подгруппе 503, и данной ПС присваиваются следующие критерии (обязательные и применимые для данной ПС):

- работоспособность;
- легкость освоения;
- доступность эксплуатационных программных документов;
- полнота реализации;
- согласованность;
- логическая корректность;
- проверенность.

Для показателей качества на всех уровнях (факторы, критерии, метрики, оценочные элементы) принимается единая шкала оценки от 0 до 1.

Таким образом, значения критериев:

- работоспособность;

Метрики критерия работоспособность представлены на рисунке 2.

H0401	Вероятность безотказной работы	То же	$P = 1 - Q/N$, где Q — число зарегистрированных отказов, N — число экспериментов,
H0501	Оценка по среднему времени восстановления	»	$Q_в = \begin{cases} 1, \text{ если } T_в \le T_в^{доп} \\ \dfrac{T_в^{доп}}{T_в}, \text{ если } T_в > T_в^{доп} \end{cases}$, где $T_в^{доп}$ — допустимое среднее время восстановления; $T_в$ — среднее время восстановления, которое определяется по формуле $T_в = \frac{1}{N} \sum_{i}^{N} T_{в_i}$, где N — число восстановлений; $T_{в_i}$ — время восстановления после i-го отказа
H0502	Оценка по продолжительности преобразования входного набора данных в выходной	Расчетный	$Q_{п_i} = \begin{cases} 1, \text{ если } T_{п_i} \le T_{п_i}^{доп} \\ \dfrac{T_{п_i}^{доп}}{T_{п_i}}, \text{ если } T_{п_i} > T_{п_i}^{доп} \end{cases}$, где $T_{п_i}^{доп}$ — допустимое время преобразования i-го входного набора данных; $T_{п_i}$ — фактическая продолжительность преобразования i-го входного набора данных

Рисунок 2 – Метрики критерия работоспособность

Вероятность безотказной работы, в результате экспериментов:

$$\boldsymbol{P = 1 - \frac{Q}{N} = 1 - \frac{1}{11} = 0.909}$$

Оценка по среднему времени восстановления, в результате экспериментов:

$$\boldsymbol{Q_в = \begin{cases} 1, \text{если } T_в \le T_в^{доп} \\ \dfrac{T_в^{доп}}{T_в}, \text{если } T_в > T_в^{доп} \end{cases}}, \text{то есть } \boldsymbol{Q_в = 1}, \text{так как } \boldsymbol{T_в \le T_в^{доп}}$$

Оценка по продолжительности преобразования входного набора данных в выходной, в результате экспериментов:

$$\boldsymbol{Q_{ni} = \begin{cases} 1, \text{если } T_{ni} \le T_{ni}^{доп} \\ \dfrac{T_{ni}^{доп}}{T_{ni}}, \text{если } T_{ni} > T_{ni}^{доп} \end{cases}}, \text{то есть } \boldsymbol{Q_{ni} = 1}, \text{так как } \boldsymbol{T_{ni} \le T_{ni}^{доп}}$$

легкость освоения;

Метрики критерия легкость освоения представлены на рисунке 3.

У0101	Возможность освоения программных средств по документации	Экспертный	0—1
У0102	Возможность освоения ПС на контрольном примере при помощи ЭВМ	То же	0—1
У0103	Возможность поэтапного освоения ПС	»	0—1
У0201	Полнота и понятность документации для освоения	»	0—1
У0202	Точность документации для освоения	»	0—1
У0203	Техническое исполнение документации	»	0—1
У0301	Наличие краткой аннотации	»	0—1
У0302	Наличие описания решаемых задач	»	0—1
У0303	Наличие описания структуры функций ПС	»	0—1
У0304	Наличие описания основных функций ПС	»	0—1
У0306	Наличие описания частных функций	»	0—1
У0307	**Наличие описания алгоритмов**	»	0—1
У0308	Наличие описания межмодульных интерфейсов	»	0—1
У0309	Наличие описания пользовательских интерфейсов	»	0—1
У0310	Наличие описания входных и выходных данных	»	0—1
У0311	Наличие описания диагностических сообщений	»	0—1
У0312	Наличие описания основных характеристик ПС	»	0—1
У0314	Наличие описания программной среды функционирования ПС	»	0—1
У0315	Достаточность документации для ввода ПС в эксплуатацию	»	0—1
У0316	Наличие информации технологии переноса для мобильных программ	»	0—1

Рисунок 3 – метрики критерия легкость освоения

Таким образом, оценки метрик:

У0101 – 0,73;

У0102 – 0,8;

У0103 – 0,4;

У0201 – 0,1;

У0202 – У0316: 0,1.

доступность эксплуатационных программных документов;

Метрики критерия доступность эксплуатационных программных документов представлены на рисунке 4.

У0401	Соответствие оглавления содержанию документации	»	0—1
У0402	Оценка оформления документации	»	0—1
У0403	Грамматическая правильность изложения документации	»	0—1
У0404	Отсутствие противоречий	»	0—1
У0405	Отсутствие неправильных ссылок	»	0—1
У0406	Ясность формулировок и описаний	»	0—1
У0407	Отсутствие неоднозначных формулировок и описаний	»	0—1
У0408	Правильность использования терминов	»	0—1
У0409	Краткость, отсутствие лишней детализации	»	0—1
У0410	Единство формулировок	»	0—1
У0411	Единство обозначений	»	0—1
У0412	Отсутствие ненужных повторений	»	0—1
У0413	Наличие нужных объяснений	»	0—1
У0501	Оценка стиля изложения	»	0—1
У0502	Дидактическая разделенность	»	0—1
У0503	Формальная разделенность	»	0—1
У0504	Ясность логической структуры	»	0—1
У0505	Соблюдение стандартов и правил изложения в документации	»	0—1
У0506	Оценка по числу ссылок вперед в тексте документов	»	0—1
У0601	Наличие оглавления	Экспертный	0—1
У0602	Наличие предметного указателя	То же	0—1
У0603	Наличие перекрестных ссылок	»	0—1
У0604	Наличие всех требуемых разделов	»	0—1
У0605	Соблюдение непрерывности нумерации страниц документов	»	0—1
У0606	Отсутствие незаконченных разделов абзацев, предложений	»	0—1
У0607	Наличие всех рисунков, чертежей, формул, таблиц	»	0—1
У0608	Наличие всех строк и примечаний	»	0—1
У0609	Логический порядок частей внутри главы	»	0—1
У0701	Наличие полного перечня документации	»	0—1

Рисунок 4 – метрики критерия доступность эксплуатационных программных документов

Таким образом, оценки метрик:
У0401 – У0413: 0,7;
У0501 – У0506: 0,65;
У0601 – У0609: 0,8;
У0701 – 0,7.
полнота реализации;
Метрики критерия полнота реализации представлены на рисунке 5.

К0101	Наличие всех необходимых документов для понимания и использования ПС	Экспертный	0—1
К0102	Наличие описания и схемы иерархии модулей программы	То же	0—1
К0103	Наличие описания основных функций	»	0—1
К0104	Наличие описания частных функций	»	0—1
К0105	Наличие описания данных	»	0—1
К0106	Наличие описания алгоритмов	»	0—1
К0107	Наличие описания интерфейсов между модулями	»	0—1
К0108	Наличие описания интерфейсов с пользователями	»	0—1
К0109	Наличие описания используемых числовых методов	»	0—1
К0110	Указаны ли все численные методы	»	0—1
К0111	Наличие описания всех параметров	»	0—1
К0112	Наличие описания методов настройки системы	»	0—1
К0113	Наличие описания всех диагностических сообщений	»	0—1
К0114	Наличие описания способов проверки работоспособности программы	»	0—1

Рисунок 5 – метрики критерия полнота релизации

Таким образом, оценки метрик:
К0101 – 0,5;
К0102 – К0109: 0,2;
К0110 – К0114: 0,3.
согласованность;
Метрики критерия согласованность представлены на рисунке 6.

К0501	Единообразие способов вызова модулей	»	0—1
К0502	Единообразие процедур возврата управления из модулей	»	0—1
К0503	Единообразие способов сохранения информации для возврата	»	0—1
К0504	Единообразие способов восстановления информации для возврата	»	0—1
К0505	Единообразие организации списков передаваемых параметров	»	0—1
К0601	Единообразие наименования каждой переменной и константы	»	0—1
К0602	Все ли одинаковые константы встречаются во всех программах под одинаковыми именами	»	0—1
К0603	Единообразие определения внешних данных во всех программах	»	0—1
К0604	Используются ли разные идентификаторы для разных переменных	»	0—1
К0605	Все ли общие переменные объявлены как общие переменные	»	0—1
К0606	Наличие определений одинаковых атрибутов	»	0—1

Рисунок 6 – метрики критерия согласованность

Таким образом, оценки метрик:
К0501 – К0505: 0,4;
К0601 – К0606: 0,2.
логическая корректность;
Метрики критерия логическая корректность представлены на рисунке 7.

К0801	Соответствие организации и вычислительного процесса эксплуатационной документации	»	0—1
К0802	Правильность заданий на выполнение программы, правильность написания управляющих и операторов (отсутствие ошибок)	»	0—1
К0803	Отсутствие ошибок в описании действий пользователя	»	0—1
К0804	Отсутствие ошибок в описании запуска	»	0—1
К0805	Отсутствие ошибок в описании генерации	»	0—1
К0806	Отсутствие ошибок в описании настройки	»	0—1

Рисунок 7 – метрики критерия логическая корректность

Таким образом, оценки метрик:

- К0801: 0,1;
- К0803:0,1;
- К0802, К0804 – К0806: 0,2;

- проверенность.

Метрики критерия проверенность представлены на рисунке 8.

К1001	Наличие требований к тестированию программ	»	0—1
К1002	Достаточность требований к тестированию программ	»	0—1
К1003	Отношение числа модулей, отработавших в процессе тестирования и отладки ($Q_т^м$) к общему числу модулей ($Q_о^м$)	Расчетный	$\frac{Q_т^м}{Q_о^м}$
К1004	Отношение числа логических блоков, отработавших в процессе тестирования и отладки ($Q_т^б$), к общему числу логических блоков в программе ($Q_о^б$)	То же	$\frac{Q_т^б}{Q_о^б}$

Рисунок 8 – метрики критерия проверенность

Таким образом, оценки метрик:

- К1001 – К1004: 0,2.

2.2 Техническое задание согласно ГОСТ 34.602-89

2.2.1 Общие сведения

2.2.1.1 Полное наименование системы и ее условное обозначение

Полное наименование: Система кластеризации абонентов мобильной сети.
Условное обозначение: СКАМС.

2.2.1.2 Шифр темы или шифр (номер) договора

Шифр темы и номер договора отсутствуют по причине выполнения данного проекта в рамках учебной деятельности.

2.2.1.3 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты

Исполнитель: Сибирский федеральный университет, Институт космических и информационных технологий, группа КИ15-12Б, Мельситова О.А.
Заказчик: Мобильный оператор.

2.2.1.4 Документы и информационные материалы, на основании которых разрабатывалось ТЗ и которые использованы при создании системы

ГОСТ 6.10.1-80 «Унифицированные системы документации. Основные положения»;
ГОСТ 19.201-78 «Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению (с Изменением N 1)»;
ГОСТ 24.601-86 «Единая система стандартов автоматизированных систем управления. Автоматизированные системы. Стадии создания»;
ГОСТ 34.201-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем».
ГОСТ 19.301-79 «Единая система программной документации. Программа и методика испытаний. Требования к содержанию и оформлению».
ГОСТ 34.601-90 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания».

2.2.1.5 Плановые сроки начала и окончания работ

Дата начала выполнения работ: 15.01.2019.
Дата окончания работы: 15.06.2019.

2.2.1.6 Сведения об источниках и порядке финансирования работ

Работа осуществляется без финансирования в связи с тем, что разработка проводится в рамках учебной программы.

2.2.1.7 Порядок оформления и предъявления Заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических комплексов системы

Выполнятся в соответствии с ГОСТ 34.602-89.

Подробно порядок оформления и предъявления Заказчику результатов работ по созданию АС определен следующими разделами настоящего технического задания (далее – ТЗ):

- состав и содержание работ по созданию АС (раздел 2);
- порядок контроля и приемки АС (раздел 7);
- требования к системе (раздел 5).

2.2.2 Назначение и цели создания (развития) системы

2.2.2.1 Назначение системы

Кластеризация абонентов по группам общения.

2.2.2.2 Цели создания системы

Создание системы для разбиения множества абонентов на группы.

2.2.3Характеристика объекта автоматизации

2.2.3.1 Краткие сведения об объекте автоматизации

Объектами автоматизации является процесс разделения абонентов на группы общения.

2.2.4Требования к системе

2.2.4.1 Требования к системе в целом

Показатель качества «Согласованность» равен 0,86.

2.2.4.2 Требования к структуре и функционированию системы

2.2.5 Перечень подсистем, их назначение и основные характеристики

В состав системы должны входить следующие подсистемы:

- Разработка системы кластеризации абонентов;
- программная реализация системы.

Строк кода не менее 150.

Объем памяти, занимаемый приложением не более 4 Гб.

2.2.6 Требования к способам и средствам связи для информационного обмена между компонентами системы

Не предъявляются.

2.2.7 Требования к численности и квалификации персонала системы

Для эксплуатации СКАМС требуется два человека: главный инженер и сотрудник отдела технической поддержки.

2.2.8 Показатели назначения

Изменение параметров приспосабливаемости системы не предусмотрены.

2.2.9 Требования к надежности

2.2.9.2 Состав и количественные значения показателей надежности для системы в целом или ее подсистем

Средняя наработка на отказ составила 2 ч 30 мин.

2.2.9.3 Перечень аварийных ситуаций, по которым должны быть регламентированы требования к надежности, и значения соответствующих показателей

- сбой в электроснабжении компьютера не более 1 раза в месяц;
- отказ компьютера не более 1 раза в месяц.

Вероятность безотказной работы равна 0,909.

Оценка по среднему времени восстановления равна 1.

2.2.9.4 Требования к надежности технических средств и программного обеспечения

- аппаратные средства не менее 0,90.

2.2.10 Требования к эргономике и технической эстетике

- Показатели удобства применения:

- легкость освоения – 0,73;
- доступность эксплуатационных программных документов – 0,8.

2.2.11Требования к транспортабельности для подвижных АС

Разрабатываемое АС не является подвижным.

2.2.12Требования по сохранности информации при авариях

Резервное копирование на сервер не реже 1 раза в неделю.

2.2.13Требования к патентной чистоте

Разрабатываемое АС предполагает использование только на территории Российской Федерации. Патентная чистота должна быть обеспечена на территории РФ.

2.2.14Требования по стандартизации и унификации

Используются стандартные программные оболочки для кластеризации.

2.2.15Требования к функциям (задачам), выполняемым системой

Ошибка классификатора объектов составляет не более 25%.
Количество объектов для распознавания классификатором не менее 1 изображения.
Показатель сопровождения равен 0,75.

2.2.16Требования к видам обеспечения

2.2.16.1 Требования к математическому обеспечению системы

При разработке алгоритма кластеризации объектов используется математическая статистика, дискретная математика**.**

2.2.16.2 Требования к лингвистическому обеспечению

При реализации системы применяются следующие языки высокого уровня: Python.

2.2.16.3 Требования к техническому обеспечению

Для стабильного функционирования АС персональные компьютеры должны соответствовать следующим характеристикам:

- операционная система: Windows (версии 7 и выше);
- оперативная память: не менее 4 GB RAM DDR3;
- видеокарта: не хуже NVIDIA GeForce GTX 1060;

– процессор: 500 MHz;
– место на диске: не менее 5 GB.

Дополнительно: поддержка клавиатуры и мыши, наличие монитора.

2.2.16.4 Требования к метрологическому обеспечению

Дополнительные требования к метрологическому обеспечению не предъявляются.

2.2.16.5 Требования к методическому обеспечению

Дополнительные требования к методическому обеспечению не предъявляются.

2.2.17 Состав и содержание работ по созданию (развитию) системы

Состав, содержание и порядок выполнения работ на установленных стандартом ГОСТ 24.601-86 стадиях и этапах определяют в нормативно-технической документации по созданию АС соответствующего вида. Этапы проведения работ приведены в Таблице 2.

Таблица 2 – этапы проведения работ

Стадии	Этапы работ
Разработка алгоритма работы кластеризации	- подбор эффективной архитектуры; - настройка параметров сети.
Доработка проекта и тестирование	- тестирование эффективности; - оформление графического интерфейса; - тестирование на возможные ошибки.
Создание пояснительной записки	- разработка пояснительной записки для проекта.
Защита проекта	- предоставление готового проекта Заказчику.

2.2.18Порядок контроля и приемки системы

Все создаваемые в рамках настоящей работы программные изделия (за исключением покупных) передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, представляемых в электронной форме на стандартном машинном носителе (например, на компакт-диске).

2.2.19Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

2.2.19.1 Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ

Предоставление лицензионного ПО в соответствии с:

– ГОСТ Р ИСО/МЭК 17799-2005 «Информационная технология. Практические правила управления информационной безопасностью»;

– Федеральный закон от 27.07.2006 N 149-ФЗ "Об информации, информационных технологиях и о защите информации";

– ГОСТ 19781-90 "Обеспечение систем обработки информации программное. Термины и определения";

– Закон РФ от 23 сентября 1992 г. № 3523I «О правовой охране программ для электронных вычислительных машин и баз данных».

Показатель качества «Полнота реализации» составил 0,71.

2.2.19.2 Сроки и порядок комплектования штатов и обучения персонала

В соответствии с приказом Минтруд от 09.01.14 г. №2н.

3 Проектное решение и архитектура системы

3.1 Входные данные

Имеются лог-файлы сервера за каждый день его функционирования в течение месяца. Данные за различные периоды(недели) разделены на разные каталоги.

В каждой строке лог-файла содержится различная информация о пользователе, такая как :

- ctn - номер абонента, зашифрован;
- imsi – зашифрован;
- разговор- длительность диалога;
- смс- количество смс;
- ммс- количество ммс;
- геотаргетинг – зашифрован;
- num2 - это второй номер, входящий/исходящий в зависимости от события, зашифрован;
- imei телефона – зашифрован;

3.2 Предобработка

Во входных данных содержится много лишней информации, которая не будет использоваться в дальнейшем, поэтому из каждой записи важно выделить только необходимую информацию, и сохранить в более компактном виде.

Первым шагом, из всей информации, представленной в каждой записи, было выделено три основных параметра:

- Идентификатор собеседника

- Количество минут разговора
- Количество смс

3.3 Нормирование векторов

Как уже было сказано раннее, кластерный анализ лучше всего работает на множестве нормированных векторов, поэтому перед началом кластеризации необходимо произвести нормирование таблицы.

Встает выбор между формулами

$$X^* = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \text{и} \quad X^* = \frac{x}{x_{max}},$$

Где X^*-новое значение ячейки, x_{min}- минимальное значение в строке, x_{max}-максимальное значение в строке.

Нормирование векторов по первой формуле, укладывает все значения в отрезок [0;1], причем значения 0 и 1 будут обязательно присутствовать в каждом векторе.

Использование второй формулы не гарантирует присутствие 0 в каждой строке. Характеристика будет принимать нулевое значение только при условии, что ее значение было равным нулю и до нормирования.

Исходя из поставленной задачи, правильнее будет использовать вторую формулу, так как очень важно даже минимальное значение характеристики, которое в первом случае отождествляется с нулем.

Данный процесс был реализован с помощью библиотеки NumPy [13], преимущество которой в том, что она позволяет работать со строками матрицы так же, как и с обычными ячейками. Например, процесс деления всей строки на максимальный элемент можно реализовать без циклов, в одну строку.

3.4 Кластеризация

Полученные нормированные данные можно кластеризовать. Для решения этой задачи хорошо подходит библиотека SciPy. В этой библиотеке реализован модуль cluster.hierarchy, предоставляющим набор средств для иерархической кластеризации.

Для выполнения кластеризации использовалась соответствующая функция linkage(z, method, metric). Данная функция принимает следующие входные параметры [15]:

1) Z - массив векторов или матрица, описывающие объекты.
2) Method – метод вычисления расстояния между кластерами.
3) Metric – метрика для вычисления расстояния между точками. По умолчанию – Евклидова. u[i], v[j]

Функция linkage подерживает множество методов вычисления расстояния между кластерами:

Single. Расстояние между кластерами вычисляется как минимальное расстояние, среди расстояний между всеми парами точек из каждого кластера. Этот метод еще называется «Ближайший сосед»

$$d(U,V) = \min(d(u[i],v[j])),\ \forall i, j\ .$$

Complete. Метод аналогичен предыдущему, но вместо минимального расстояния, берется максимальное.

$$d(U,V) = \max(d(u[i],v[j])),\ \forall i, j.$$

Average. В данном методе расстояние вычисляется по формуле:

$$d(U,V) = \sum_{i,j} \frac{d(u[i],v[j])}{| \quad | \ | \quad |},\ \forall i, j.$$

Ward. Данный метод использует алгоритм минимизации дисперсии Уорда. Расстояние вычисляется по формуле:

$$d(U,V) = \sqrt{\frac{|V|+|S|}{Q} * d(V,S) + \frac{|V|+|T|}{Q} * d(V,T) - \frac{|V|}{T} * d(S,T)^2}\ ,$$

После выполнения, функция linkage возвращает массив связей, являющийся результатом иерархической кластеризации. Этот массив можно графически отобразить с помощью специальных функций или, выполнив функцию fcluster, получить разбиение на заданное число кластеров .

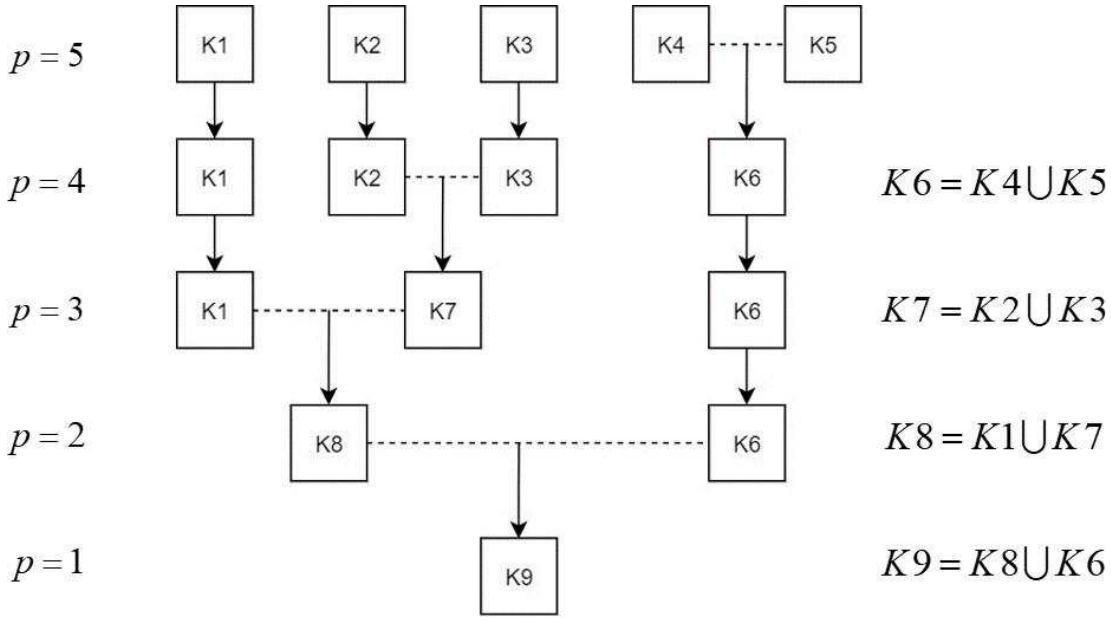

Рисунок 5 - Схема отображающая процесс кластеризации

4 Анализ результатов

4.2 Визуализация

Самым простым методом визуализации результатов иерархической кластеризации является дендрограмма – бинарное дерево разбиения кластеров. Выполнив функцию dendrogram из библиотеки SciPy была получена дендрограмма (рис. 6), отображающая процесс разбиения множества пользователей на кластеры. По оси абсцисс отображаются размеры кластеров, а по оси ординат – расстояние d между кластерами.

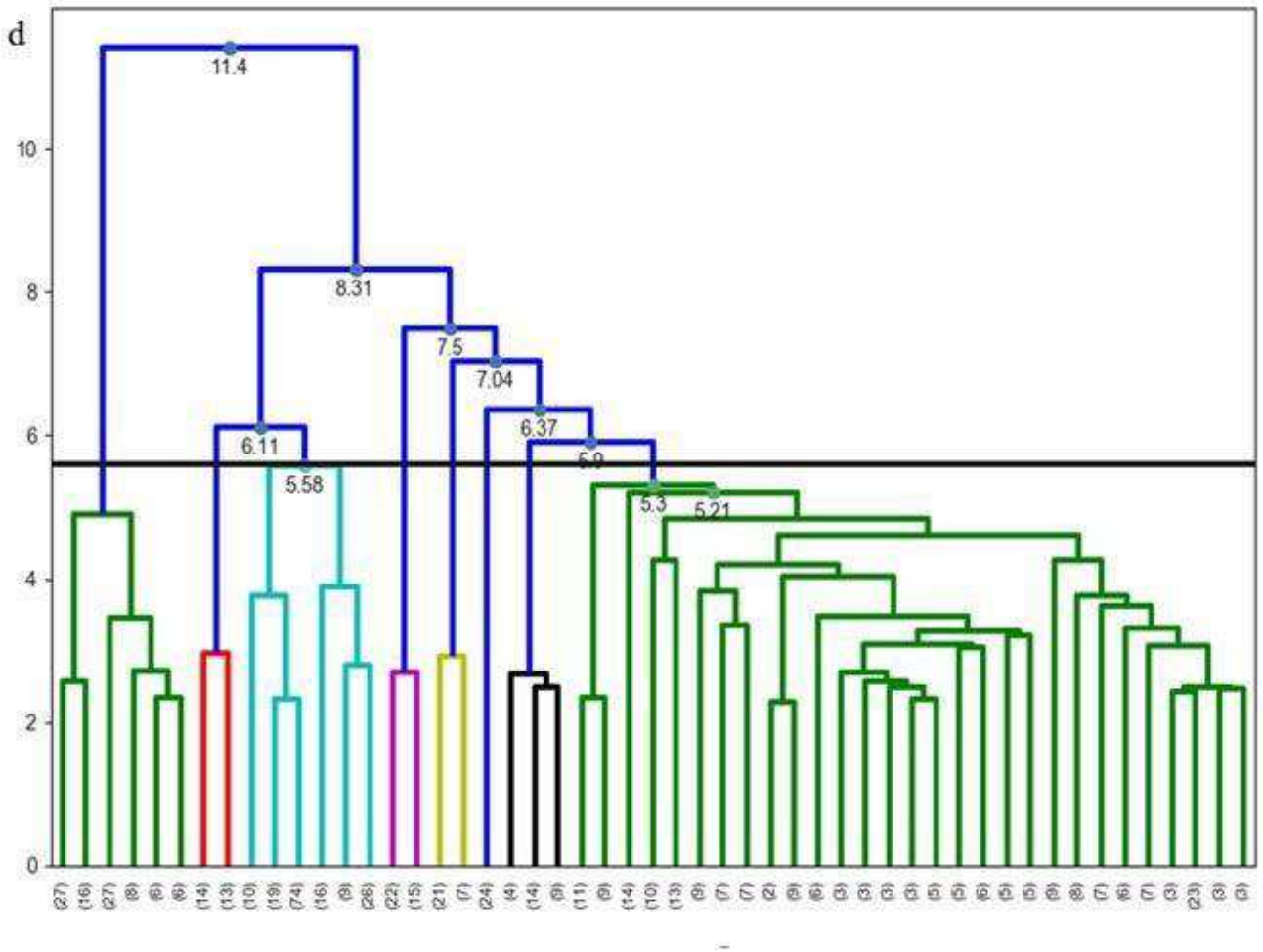

Рисунок 6 - Дендрограмма иерархической кластеризации

Данный метод визуализации показывает дерево разбиений, в данном случае - усеченное, но не дает достаточного представления об расположении объектов в пространстве. Возникает проблема отображения многомерных кластеров на двухмерную плоскость. Поэтому, для проекции на двухмерную плоскость используются методы понижения размерности пространства.

Метод главных компонент (PCA) - один из основных способов уменьшить размерность данных, потеряв наименьшее количество информации. Другим языком, данный метод аппроксимирует n-мерное пространство объектов до n-мерного эллипсоида, полуоси которого, в дальнейшем, будут считаться главными компонентами. При проекции именно на эти оси сохраняется наибольшее количество информации.

На рис. 7 и 8 представлены результаты проекции данного множества объектов на главные компоненты при разном числе кластеров. Красной тенью на изображениях показана относительная плотность множества объектов.

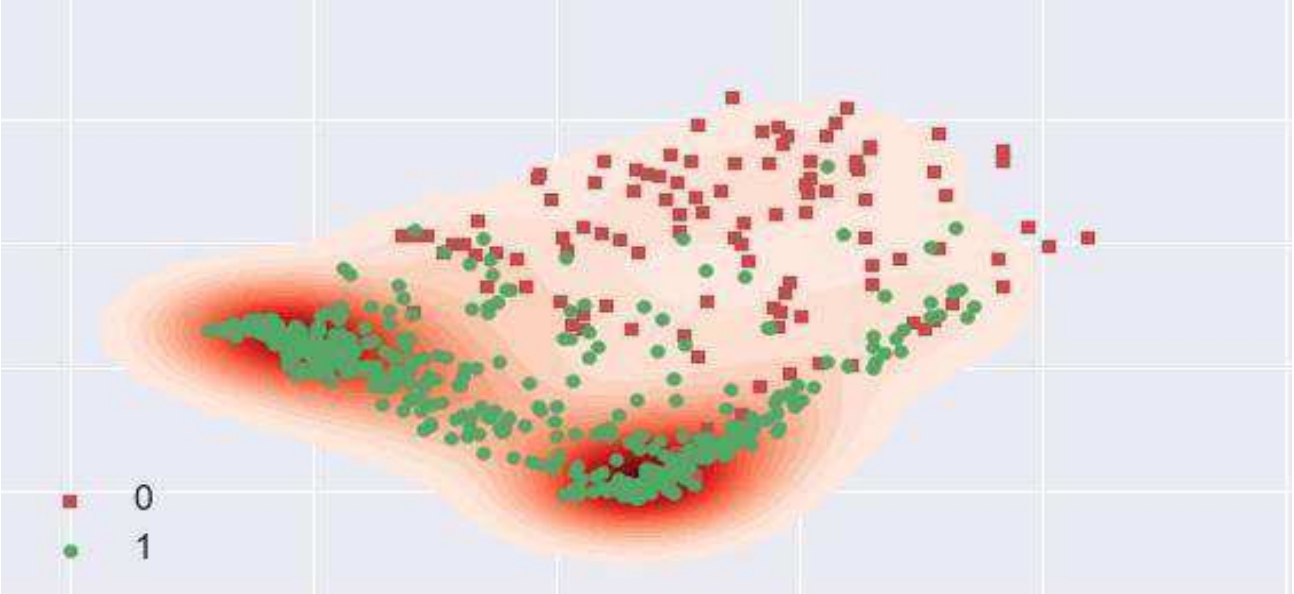

Рисунок 7- Результат применения метода главных компонент для 2 кластеров

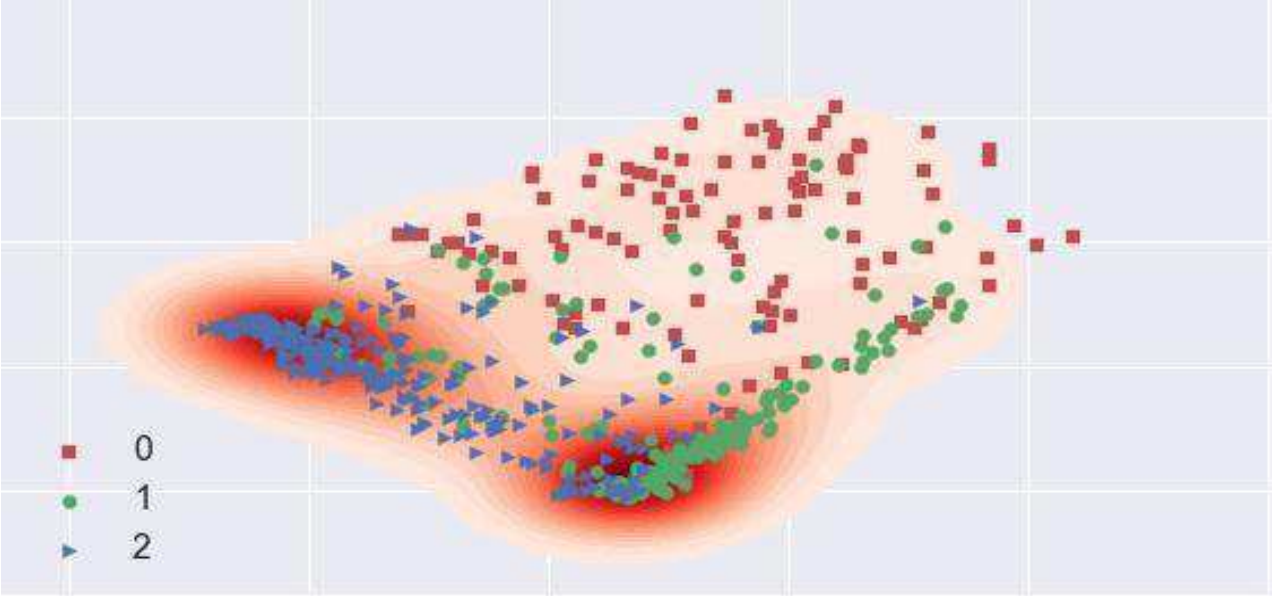

Рисунок 8 - Результат применения метода главных компонент для 3 кластеров

Как видно по рисунку, в результате применения метода главных компонент получились схожие изображения, отличающиеся только маркерами, отвечающими за принадлежность объекта к одному из кластеров. В этом минус использования данного метода для проекции различных разбиений, ведь его применение не показывает изменение расположения кластеров и их формы в зависимости от разбиения.

Для получения различных проекций, зависящих от разбиения, часто используют другой метод понижения размерности пространства - линейный дискриминантный анализ.

Линейный дискриминантный анализ (LDA) - это метод поиска линейной комбинации переменных, наилучшим образом разделяющей два или более класса. Чаще всего результаты LDA используют как часть линейного классификатора, однако другим возможным применением является понижение размерности входных данных.

В отличии от PCA, при использовании LDA, проекции при различном числе кластеров будут существенно отличаться, что поможет выявить оптимальное число кластеров, на которое необходимо разбивать данное множество объектов. На рис. 9 и 10 отображены проекции кластеров пользователей с использованием линейного дискриминантного анализа в качестве метода понижения размерности пространства.

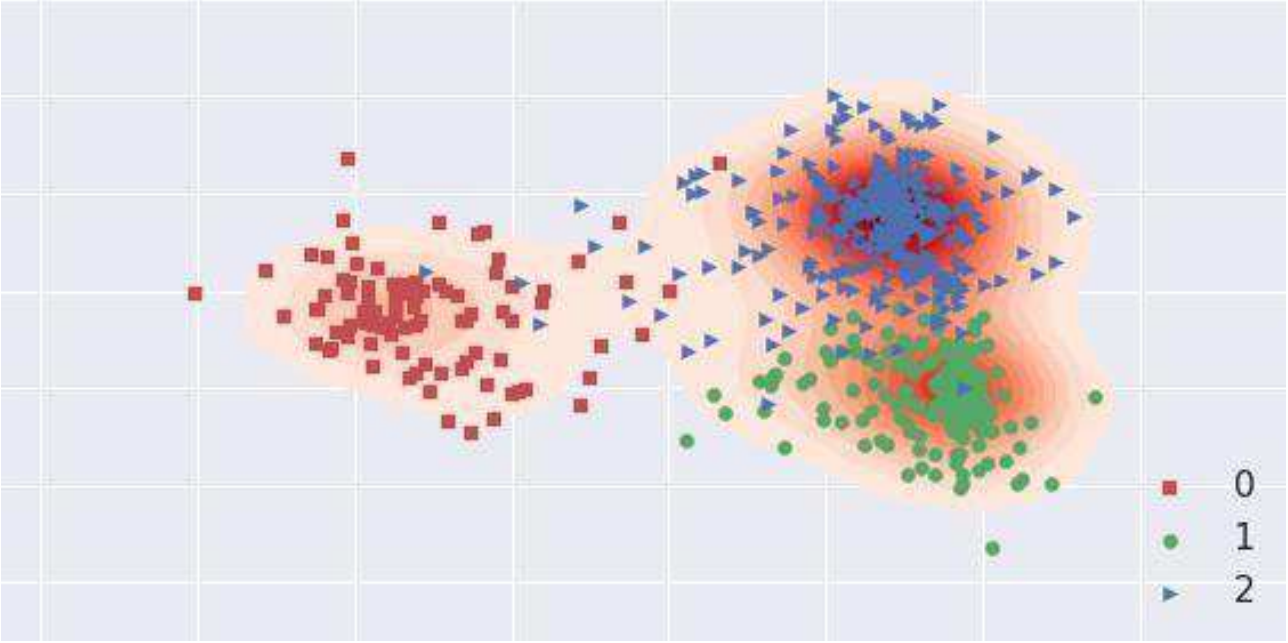

Рисунок 9 - Результат применения линейного дискриминантного анализа для 3 кластеров

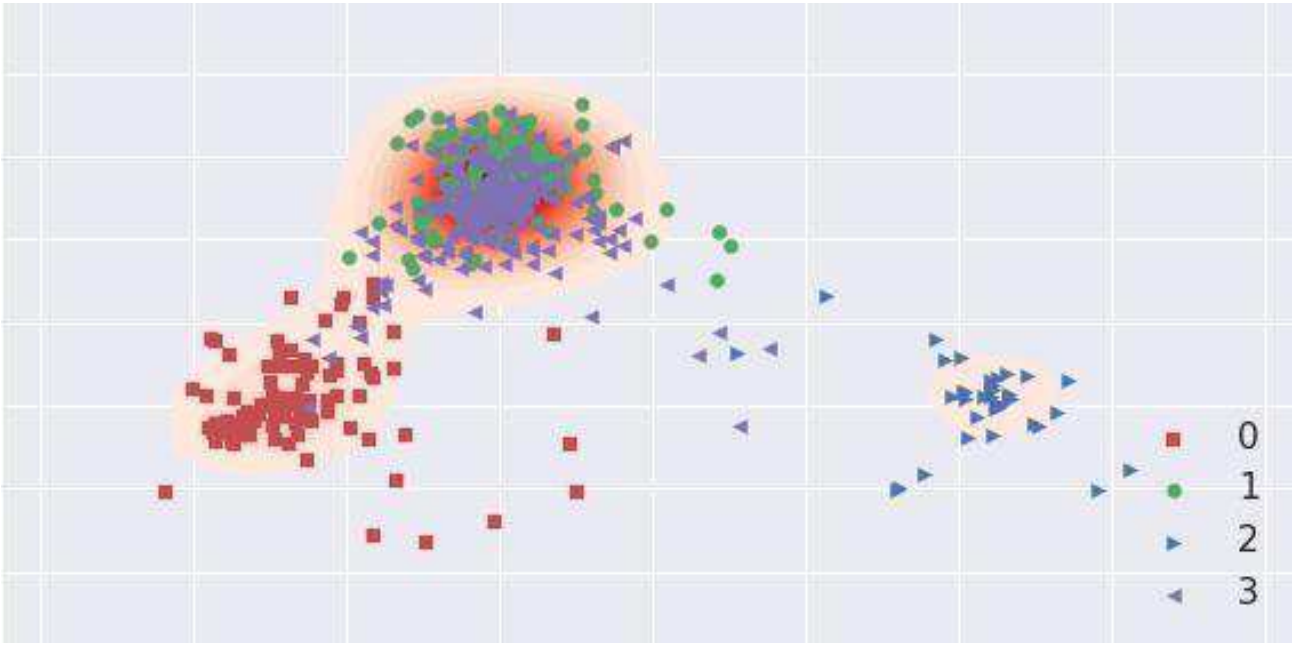

Рисунок 10 - Результат применения линейного дискриминантного анализа для 4 кластеров

Исследуя проекции можно заметить, что при числе кластеров равном 3 видно отчетливое разделение кластеров, в отличии от 4.

4.3 Оптимальное число кластеров

Одним из важных вопросов при решении задачи кластеризации является выбор необходимого числа кластеров. В некоторых случаях это число может быть задано априорно, однако в общем случае это число определяется в процессе разбиения множества на различное число кластеров [10].

Анализируя разбиения на различные числа кластеров, можно определить такое число кластеров, при котором кластеры разделены наилучшим образом. Такое число кластеров будем называть оптимальным для заданной выборки данных. Определить это число можно с помощью визуализации, рассмотренной выше, или другими различными методами.

Одним из таких методов является метод локтя. Этот метод заключается в построении функции изменения вариаций данных внутри кластеров при изменении числа кластеров. Число кластеров, при котором отмечено наибольшее изменение функции, можно считать оптимальным. Если таких чисел несколько, то все они могут являться оптимальными.

Как видно по графику, изображенному на рис. 11, при значениях количества кластеров равных 2 и 3 функция сильно изменяется, поэтому при дальнейшем анализе можно использовать только эти значения числа кластеров.

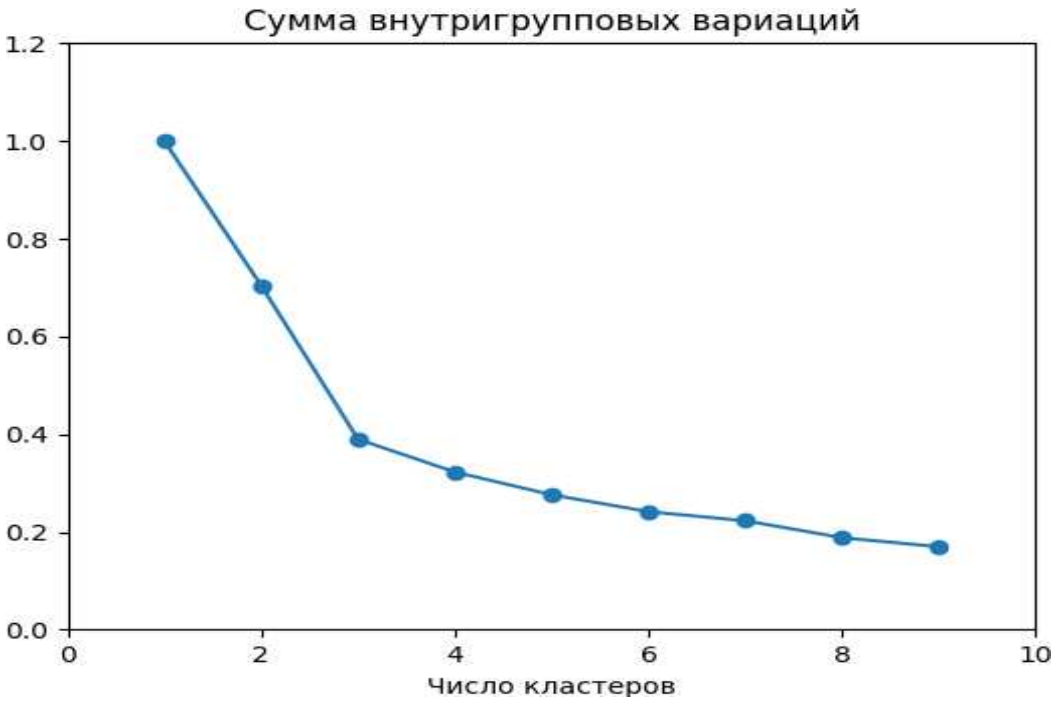

Рисунок 11 - График изменения суммы внутригрупповых вариаций от числа кластеров

4.4 Динамический анализ

Проведя кластерный анализ пользователей за один промежуток времени, например, за неделю, нельзя с уверенностью сказать, что пользователи, находящиеся в одном кластере, имеют действительно схожие предпочтения, а не оказались рядом случайно. Для проверки достоверности кластеризации можно выполнить аналогичные действия с данными взятыми из лог-файлов за другой, к примеру, последующий, промежуток времени, после чего сравнив результаты различных разбиений, можно отличить настоящих «соседей», от пользователей, случайно попавших в кластер или перемещающихся между ними.

На рис. 12-13 представлены результаты разбиений пользователей по данным взятым за последовательные 4 недели на 5 и 8 кластеров соответственно. По оси абсцисс указаны промежутки времени, высота каждого столбика в одной группе равна числу пользователей в соответствующем кластере, а сумма всех пользователей равна 591. Основываясь на этих данных, можно предположить, что устойчивые группы пользователей действительно присутствуют.

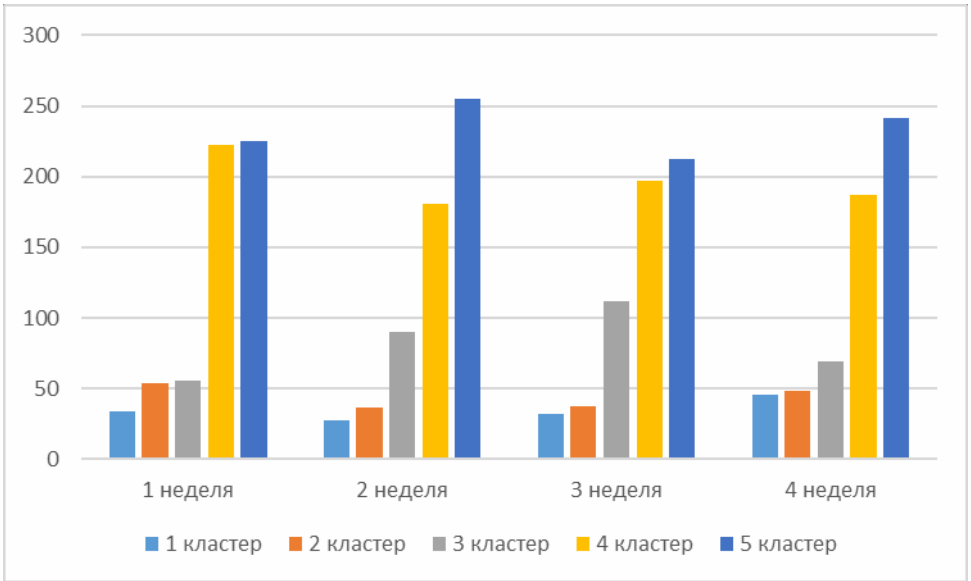

Рисунок 12 – Гистограмма разбиения пользователей по данным за 4 различные недели на 5 кластеров

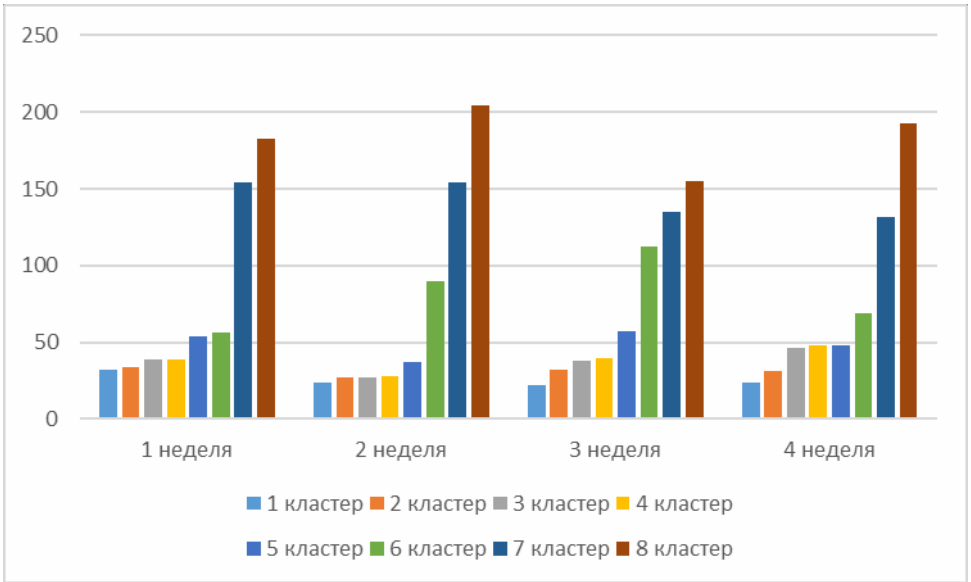

Рисунок 13- Гистограмма разбиения пользователей по данным за 4 различные недели на 8 кластеров

При необходимости выделить группы пользователей, у которых вектора принадлежности идентичны, можно воспользоваться следующим алгоритмом (рис. 14):

1) Выбрать пользователя, не принадлежащего ни одной из групп
2) Для выбранного пользователя выбрать множество, состоящее из кластеров, к которым он принадлежит в каждом разбиении
3) Найти пересечение этих кластеров, которое в результате будет состоять из пользователей, имеющих одинаковые вектора принадлежности, и пометить это множество как отдельную группу

4) Если каждый пользователь имеет свою группу – завершить работу, иначе перейти к шагу 1.

Рисунок 14 - Алгоритм выделения устойчивых групп пользователей

В результате анализа различных разбиений было выявлено, что крупные группы пользователей присутствуют в любом разбиении, и достоверность таких групп напрямую зависит от количества разбиений и кластеров.

Таблица 4, отображает размер групп, которые были выявлены среди 591 пользователя прокси сервера.

Таблица 4 – характеристики группы (не единичных) для различных разбиений

Кол. разбиений	Кол. кластеров	Кол. групп	Средняя группа	Макс. группа
2	3	9	66	208
2	5	18	33	148
2	8	34	17	111
3	3	20	29	153
3	5	38	15	100
3	8	60	9	71

4	3	37	16	116
4	5	55	10	88
4	8	74	7	63
4	25	77	4	25

На рисунке 15 отображены расположение устойчивых групп, полученных из разбиений за 2 последующие недели на 3 кластера. Как видно из рисунка, больше половины пользователей из кластеров 1 и 2 не изменили свой кластер за 2 недели.

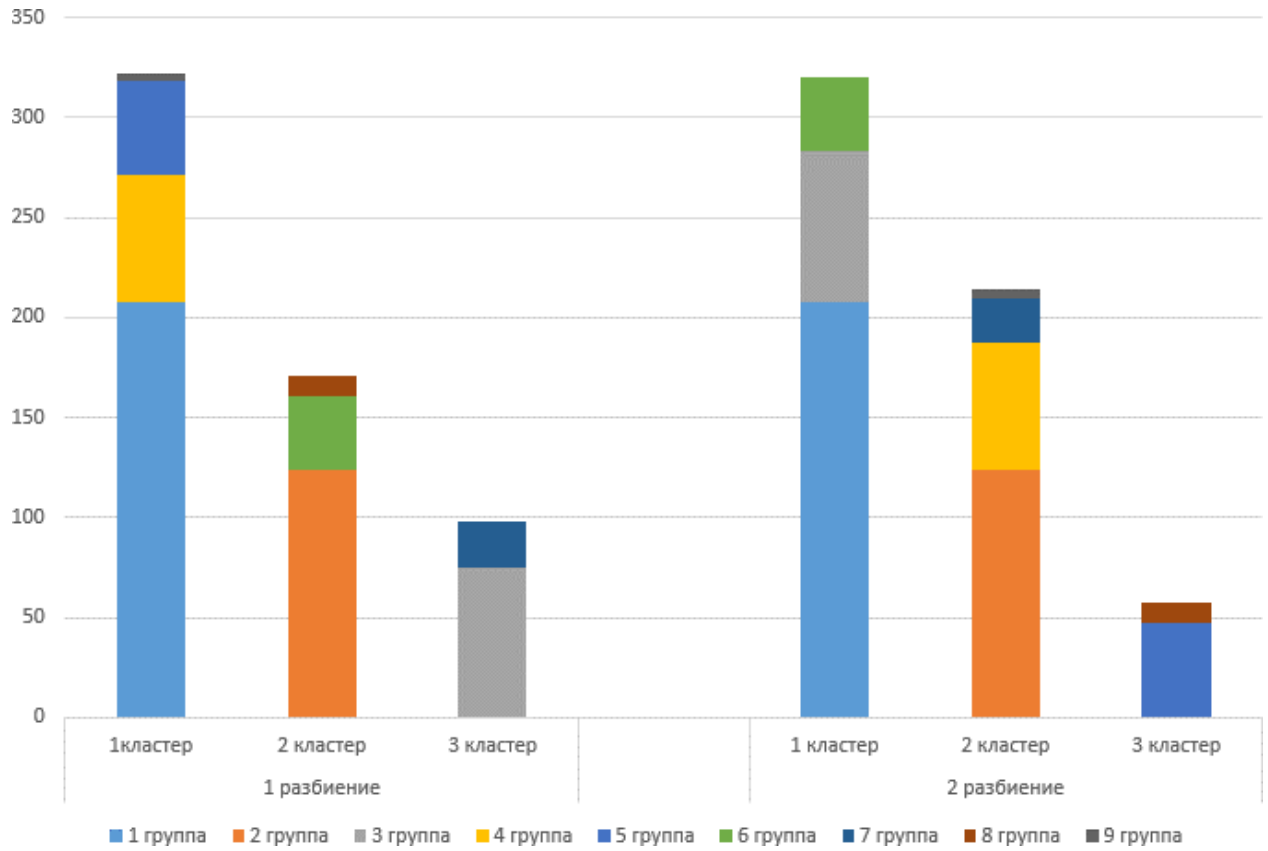

Рисунок 15 – Расположение устойчивых групп в различных разбиениях

Найденные группы можно использовать в различных целях: в маркетинге, организации эффективной работы, предложении новых тарифов и программ лояльности и для решения других интересных задач

ЗАКЛЮЧЕНИЕ

В работе получены следующие результаты:

1. обработаны записи журнала прокси-сервера за период равный одному месяцу;
2. выполнена предобработка данных;
3. произведена иерархическая кластеризация абонентов мобильной сети относительно одного клиента, по частоте общения с ним;
4. выявлены оптимальные значения числа кластеров;
5. выделены устойчивые группы пользователей со схожим поведением.

Полученные результаты могут быть использованы для решения различных задач.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бериков, В. С. Современные тенденции в кластерном анализе / В. С. Бериков, Г. С. Лбов. – Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно - телекоммуникационные системы", 2008. – 26 с.
2. Ганенкова, Е. Г. Функциональный анализ: основные классы пространств / Е. Г. Ганенкова, К. Ф. Амозова. – Петрозаводск: ПетрГУ, 2013. – 26 с.
3. Ершов, К. С. Анализ и классификация алгоритмов кластеризации / К. С. Ершов, Т. Н. Романова. // Новые информационные технологии в автоматизированных системах. – 2016. – №19. – С. 274-279.
4. Котов, А. Кластеризация данных [Электронный ресурс]. Режим доступа: http://logic.pdmi.ras.ru/~yura/internet/02ia-seminar-note.pdf.
5. Мандель, И. Д. Кластерный анализ / И. Д. Мандель. – М.: Финансы и статика, 1988. – 176 с.
6. Протасов, С.С. Как большие данные стали одной из самых интересных задач IT-индустрии [Электронный ресурс]. – Режим доступа: http://andrew--r.github.io/bigdata/.
7. Суслов, С. А. Кластерный анализ: сущность, преимущества и недостатки
/ С. А. Суслов. // Вестник НГИЭИ. – 2011. – №1. – С. 51-56.
8. Blanco-Silva, F. J. Learning SciPy for Numerical and Scientific Computing F. J. Blanco-Silva. – Packt publishing, 2015. – 150 p.
9. Downey, A. B. Think Python: An Introduction to Software Design / A. B. Downey. – O'Reilly Media, 2002. – 300 p.
10. Duran, B. S. Cluster Analysis - A Survey / B. S. Duran, P. L. Odell. – Springer, 1974. – 146 p.
11. Jain, A. K. Data Clustering: A Review / A. K. Jain, M. N. Murty, P. J. Flynn.
– ACM Computing Surveys, 1999. – 323 p.
12. Lutz, M. Programming Python / M. Lutz. – O'Reilly Media, 1996. – 1632 p.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Г. М. Цибульский
подпись
«29» 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – Информационные системы и технологии

Проектирование информационной технологии формирования групп общения для абонентов мобильных сетей

Руководитель _____ 25.06.19 доц.,канд. техн. наук К. В. Раевич
подпись, дата

Выпускник _____ 25.06.19 О. А. Мельситова
подпись, дата

Красноярск 2019