Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ « Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ

Студент Чалый Андрей Александрович
фамилия, имя, отчество

Группа ИУ7-82Б

Тип практики производственная

Название предприятия МГТУ им. Н. Э. Баумана

Студент _____ / _____
(подпись, дата) (Фамилия И.О.)

Руководитель практики от предприятия _____ / _____
(подпись, дата) (Фамилия И.О.)

Руководитель практики от МГТУ им. Н. Э. Баумана _____ / _____
(подпись, дата) (Фамилия И.О.)

Рекомендуемая оценка _____

2022 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Кафедра «Программное обеспечение ЭВМ и информационные технологии» (ИУ7)

З А Д А Н И Е
на прохождение производственной практики

на предприятии МГТУ им. Н. Э. Баумана

Студент Чалый Андрей Александрович, ИУ7-82Б

(фамилия, имя, отчество; инициалы; индекс группы)

Во время прохождения производственной практики студент должен:

1. В первой части подробно описать шаги разрабатываемой рекомендательной системы с учетом используемого алгоритма нечеткой кластеризации Гауссовой смеси. Разработать структуру программного приложения, определить требования к формату входных и выходных данных. Описать этапы работы программы их работу и взаимодействие, а также схему алгоритма разработанного метода Гауссовой смеси и рекомендательной системы на ее основе.
2. Во второй части необходимо обосновать выбор программных средств реализации метода оптимизации планирования грузоперевозок. Разработать графический интерфейс пользователя для ввода данных и отображения результатов работы программы. Привести примеры работы программы. Описать используемые методы тестирования программного обеспечения и привести его результаты.
3. В третьей части необходимо проверить правильность работы разработанного метода Гауссовй смеси, а также исследовать зависимость результатов работы рекомендательной системы от различных параметров системы.

Дата выдачи задания « ____ » _____ 20__ г.

Руководитель практики от кафедры _____ / Строганов Ю. В.
(подпись, дата) (Фамилия И.О.)

Студент _____ / _____
(подпись, дата) (Фамилия И.О.)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

За последние несколько десятилетий, с появлением Youtube, Amazon, Netflix и многих других подобных веб–сервисов, системы рекомендаций стали занимать все больше места в нашей жизни. Начиная с электронной коммерции (предлагая статьи, которые могут заинтересовать людей) и заканчивая рекламой в Интернете.

Многие современные сервисы создают рекомендательные системы, которые основываясь на информации о пользователе и его поведении в системе, пытаются определить какие объекты, будь то товары, новости, услуги и т.д. Яркими примерами служат такие сервисы или сайты, как «КиноПоиск», «Яндекс.Дзен», «Яндекс.Новости» и многие другие. «КиноПоиск» — российский веб-сайт, предлагающий пользователю к просмотру фильмы на основе его предпочтений. «Яндекс.Дзен» — веб-сайт и расширение для браузера от компании «Яндекс», ищущее в интернете информацию, которая может быть интересна пользователю, и собирающее ее в персональную ленту. «Яндекс.Новости» — российский веб-сайт, предлагающий к просмотру новости от партнеров службы, в числе которых ведущие российские и зарубежные СМИ. Поступающая информация автоматически группируется в сюжеты. На их основе формируется информационная картинка дня.

Как видно из примеров, рекомендательные системы, улучшают пользовательский опыт, упрощают нахождение наиболее интересного для пользователя контента. Поэтому со временем, количество сервисов применяющих рекомендательные системы растет и начинает широко применятся во многих сферах, таких как электронная коммерция, при поиске фильмов, музыки, научных статей, а также на новостных сайтах и в справочных центрах, а задача разработки эффективных рекомендательных систем является актуальной.

Выделяют два основных метода построения рекомендательных систем — метод фильтрации на основе содержания и метод коллаборативной фильтрации.

Методы фильтрации на основе содержания основаны на описании объекта и профиле предпочтений пользователя. Данный подход пытается подобрать объекты, похожие на те, что нравились пользователю ранее, и опирается на методы информационного поиска и машинного обучения.

Метод коллаборативной фильтрации базируется на информации об истории поведения всех пользователей в системе. К примеру, если это сайт по продаже электроники, то рекомендация по покупке товаров основывается на пользователях со схожей историей и их отношениях к объекту.

Одним из направлений обработки данных различной структуры и свойств является кластеризация. Существует множество методов кластеризации, которые можно классифицировать как четкие и нечеткие. Четкие методы кластеризации разбивают исходное множество объектов на несколько непересекающихся подмножеств. При этом любой объект принадлежит только одному кластеру. Нечеткие методы кластеризации позволяют одному и тому же объекту принадлежать одновременно нескольким (или даже всем) кластерам, но с различной степенью принадлежности. Нечеткая кластеризация во многих ситуациях более “естественна”, чем четкая, например, для объектов расположенных на границе кластеров.

Целью работы является классифицирование и сравнение существующих подходов реализации рекомендательных систем, а также выбор наиболее подходящего метода. Также требуется рассмотреть существующие алгоритмы нечеткой кластеризации и выбрать наиболее подходящий для решения поставленной задачи.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области, выделить основные определения;
- провести анализ существующих подходов реализации рекомендательных систем;
- выделить основные критерии для сравнения и выбора наиболее подходя-

щего подхода рекомендательной системы и алгоритма нечеткой кластеризации для решения проблемы;

- в результате полученных во время анализа данных разработать рекомендательную систему на основе нечеткой кластеризации;
- реализовать выбранный алгоритм нечеткой кластеризации;
- реализовать рекомендательную систему в программном продукте;
- провести исследование работоспособности реализованной рекомендательной системы.

1 Конструкторская часть

В данном разделе будут представлена IDEF0 диаграмма разрабатываемой рекомендательной системы, а также будет рассмотрен и подробно описан каждый этап разработки, приведена схема, разрабатываемой рекомендательной системы, а также описан метод тестирования модели нечеткой кластеризации.

1.1 Декомпозиция разрабатываемой рекомендательной системы

Разрабатываемая рекомендательная система состоит из этапов.

- предобработка входных данных;
- векторизация предобработанных данных;
- понижение размерности полученной матрицы;
- разделение новостей на кластеры;
- рекомендация новостей на основе данных, выделенных в кластеры.

Ниже представлена IDEF0-диаграмма разрабатываемого метода на рисунке 1.1.

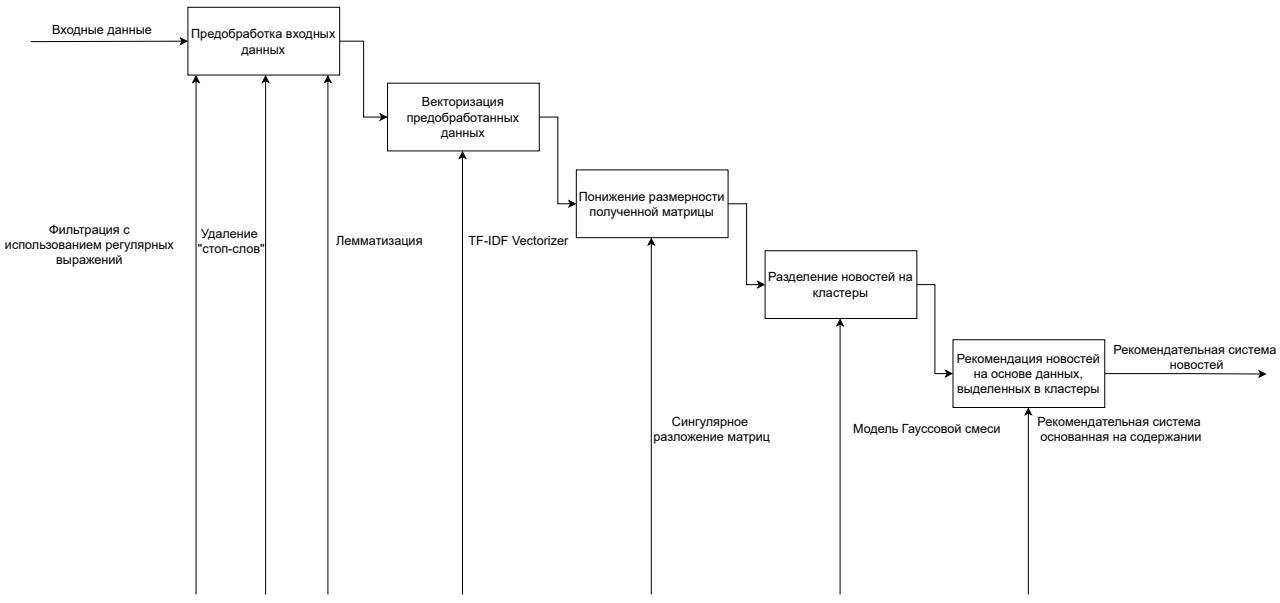

Рисунок 1.1 – IDEF0-диаграмма

1.2 Предобработка входных данных

Данный этап предназначен для подготовки к дальнейшему обучению входных данных. На вход поступают структуры из json-файла, которые содержат

информацию о новостях. Пример входных данных приведен в таблице 1

Таблица 1 – Пример входных данных.

ID	Заголовок	Абстрактное описание	Новость	Дата публикации
N45436	Walmart Slashes Prices on Last-Generation iPads	Apple’s new iPad releases bring big deals on last year’s models...	This year, Walmart’s not waiting until to offer steep deals on tech. Right now, you can save big on since new models for 2019...	10/29/2019
N23144	50 Worst Habits For Belly Fat	These seemingly harmless habits are holding you back and keeping you from shedding that unwanted belly fat for good.	When you first start dieting and exercising, the pounds seem to melt off. But, we all hit that stagnant point where the last few pounds of belly fat just don’t want to leave...	5/7/2019

Для того, чтобы корректно произвести векторизацию и последующее обучение модели, данные следует предобработать следующим образом:

- объединить столбцы с заголовком, абстрактным описанием и самой новостью;
- удалить все символы, кроме кириллических;
- удалить все ”стоп-слова”;
- провести лемматизацию.

После выполнения данного этапа будет получен массив предложений в которых содержится только необходимая для обучения информация, а все данные не несущие смысловую нагрузку удалены.

1.3 Векторизация предобработанных данных

Для векторизации полученных данных используется терм-документная частота (TF-IDF). Вычисление TF-IDF состоит из трех этапов

- вычисление TF;
- вычисление IDF;
- Произведение TF и IDF и получение TF-IDF

TF (term frequency) позволяет оценить важность терма в отдельно взятом документе. TF вычисляется по формуле 1

$$TF(w, d) = \frac{n_w}{\sum_i n_i}, \tag{1}$$

где n_w — количество вхождений терма w в документ d, $\sum_i n_i$ — количество слов в документе.

Инверсная документная частота (Inverse Document Frequency (IDF)) необходима для уменьшения веса широко употребляемых слов. Вычисляется по формуле 2.

$$IDF(w_i, D) = \log\left(\frac{|D|}{|D_i|}\right), \tag{2}$$

где $|D|$ — это общее количество документов, а $|D_i|$ — это число докумен-

тов, где w_i встретилось хотя бы раз.

Получаем что TF-IDF вычисляется по формуле 3

$$TF - IDF(w_i, D) = TF(w, d) * IDF(w_i d, D) \tag{3}$$

Векторизация подобным образом очень эффективна для последующей задачи кластеризации, так как значимые термы, встречающиеся в пределах одного документа, но редко употребляемые во всем корпусе, имеют наибольший вес.

Результатом выполнения данного этапа является матрица, строки которой — это документ (новость), а столбцы — это все термы документов (новостей). В каждой ячейке хранятся TF-IDF для конкретного терма. Так как корпус состоит из большого количества уникальных слов, то матрица получается разреженная, а также слишком большого размера.

1.4 Понижение размерности матрицы признаков

На данном этапе производится сингулярное разложение, полученной на предыдущем шаге матрицы признаков. Данное действие обусловлено тем, что вычислительные мощности моего оборудования не позволяют обрабатывать таблицу подобного размера при проведении кластеризации. Также это сделано для увеличения скорости работы алгоритма нечеткой кластеризации, что немаловажно.

1.5 Разделение новостей на кластеры

После понижения размерности выполняется этап нечеткой кластеризации новостей из полученной матрицы. В качестве алгоритма нечеткой кластеризации используется модель Гауссовой смеси (Gaussian Mixture Model (GMM)), алгоритм работы которого приведен ниже.

В данном алгоритме каждый кластер представляется параметрическим распределением, а весь набор данных моделируется смесью этих распределений, следовательно для Гауссовой смеси получаем формулу 4

$$P(\mathbf{x}|\Theta) = \sum_{i=1}^{K} \alpha_i p_i(\mathbf{x}|\theta_i), \tag{4}$$

где параметры $\Theta = (\alpha_1, ..., \alpha_K, \theta_1, ..., \theta_K)$ такие что $\sum_{i=1}^{K} \alpha_i = 1$ и каждое p_i является функцией плотности Гаусса параметризованной по θ_i. Другими словами, мы предполагаем, что у нас есть K плотностей компонентов, смешанных вместе с K коэффициентами смешения α_i.

Пусть $\mathcal{X} = (x_1, ..., x_m)$ — это набор точек данных. Требуется найти такое Θ, чтобы $p(\mathcal{X}|\Theta)$ было максимальным. Подобная задача известна как оценка максимального правдоподобия для Θ. Для оценки Θ обычно вводят логарифмическую функцию правдоподобия, определяемая по формуле 5.

$$\mathcal{L}(\Theta) = \log P(\mathcal{X}|\Theta) = \log \prod_{i=1}^{m} P(\mathbf{x}_i|\Theta) = \sum_{i=1}^{m} \log \left(\sum_{j=1}^{K} \alpha_j p_j(\mathbf{x}_i|\theta_j) \right) \tag{5}$$

Подобную функцию трудно оптимизировать, поскольку она содержит логарифм суммы. Для упрощения выражение правдоподобия, пусть $y_i \epsilon 1, \ldots, K$ обозначает, из какого Гауссиана x_i, и $\mathcal{Y} = (y1, ..., ym)$. Если мы знаем значение $\mathcal{Y}$, получаем формулу 6.

$$\begin{gathered} \mathcal{L}(\Theta) = \log P(\mathcal{X}, \mathcal{Y}|\Theta) = \log \textstyle\prod_{i=1}^{m} P(\mathbf{x}_i, y_i|\Theta) = \\ = \textstyle\sum_{i=1}^{m} \log P(\mathbf{x}_i|y_i) P(y_i) = \sum_{i=1}^{m} \log(\alpha_{y_i} p_{y_i}(\mathbf{x}_i|\theta_{y_i})) \end{gathered} \tag{6}$$

которая впоследствии оптимизируется с помощью различных методов, самым популярным из которых является алгоритм максимизации ожидания.

Исходя из названия становится ясно, что алгоритм состоит из двух частей, а именно вычисления ожидания (E), которое приведено в формулах 8, и вычисления максимизации (M), которое приведено в формулах 10, 11 и 12. Также используются вспомогательные формулы такие как 9 и 7.

$$f(x; \mu, \sum) = \frac{1}{\sqrt{(2\pi)^d * \det(\sum)}} * e^{(\frac{-1}{2} * ((x-\mu)^T * inv(\sum) * (x-\mu)))}, \tag{7}$$

где d — длина вектора x, x это выходной вектор, μ вектор средних, а $\sum$ это матрица ковариаций.

$$r_{n,k} = \frac{\pi_k N(x_n|\mu_k, \sum_k)}{\sum_j \pi_j N(x_n|\mu_j, \sum_j)}, \tag{8}$$

где π_k и pi_j — это отношения количества элементов в кластере k и j соответственно ко всем элементам данных, x_n это входной вектор данных, μ_k и μ_j это вектор средних для столбцов k и j, вычисляемый по формуле 7, а $\sum_k$ и $\sum_j$ это матрица ковариаций элементов в кластерах k и j.

$$N = \sum_k r_{n,k} \tag{9}$$

По формуле ожидания видно, что мы получаем матрицу в которой строки — это каждый элемент данных, а столбец представляет кластер, следовательно каждый элемент данной матрицы это вероятность принадлежности элемента данных к столбцу. После того как алгоритм сойдется, данные из этой матрицы будут использованы в качестве предсказания точки кластера. Также на данном шаге вычисляется N по формуле 9, которое представляет из себя список сумм столбцов матрицы $r_{n,k}$.

Схема алгоритма шага ожидания (E) приведена на рисунке 1.2.

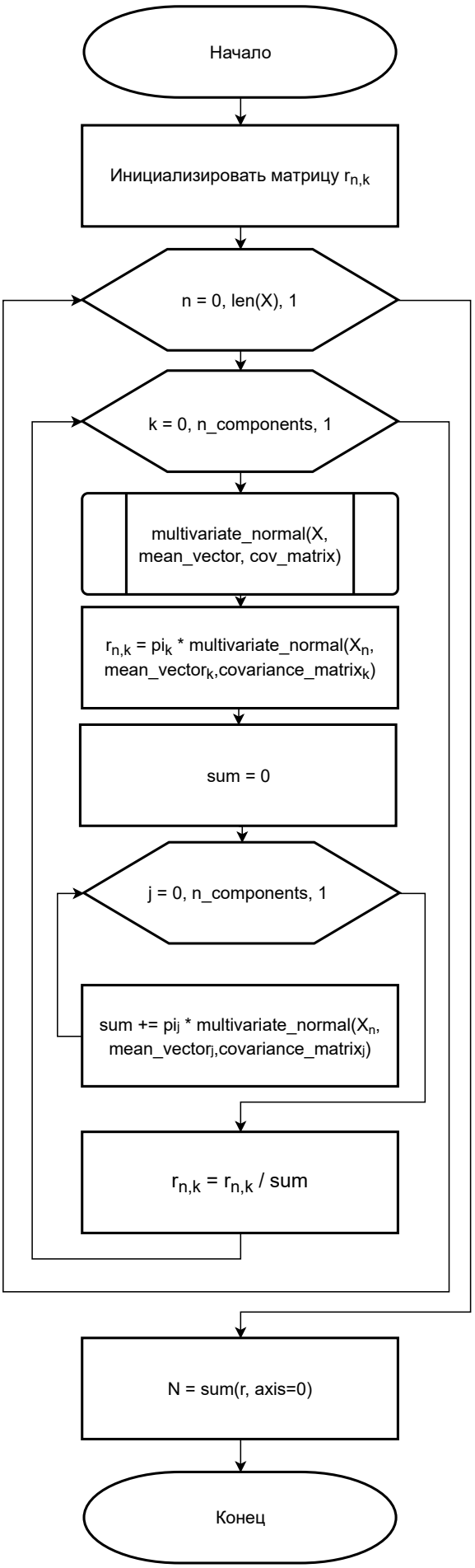

Рисунок 1.2 – Схема алгоритма шага ожидания

Вычисления максимизации (M), приведено в формулах 10, 11 и 12.

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{n,k} x_n, \tag{10}$$

$$\sum_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{n,k} (x_n - \mu_k)(x_n - \mu_k)^T, \tag{11}$$

$$\pi_k = \frac{N_k}{N}, \tag{12}$$

В формуле 10 вычисляется новый вектор средних для каждого столбца, в формуле 11 происходит обновление матрицы ковариаций для каждого столбца, а в формуле 12 обновляется список отношений количества элементов в кластере ко всем элементам данных.

Схема алгоритма шага максимизации (M) приведена на рисунке 1.3.

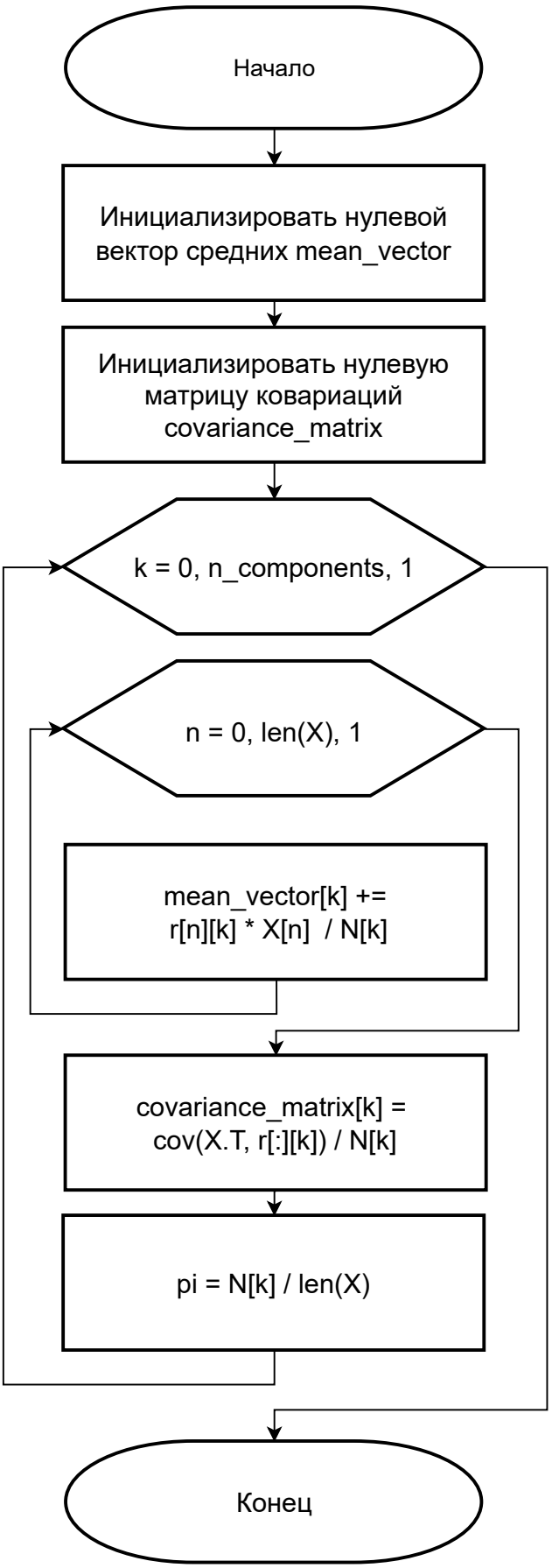

Рисунок 1.3 – Схема алгоритма шага максимизации

Схема алгоритма метода максимизации ожидания для модели Гауссовой смеси представлена на рисунке 1.4

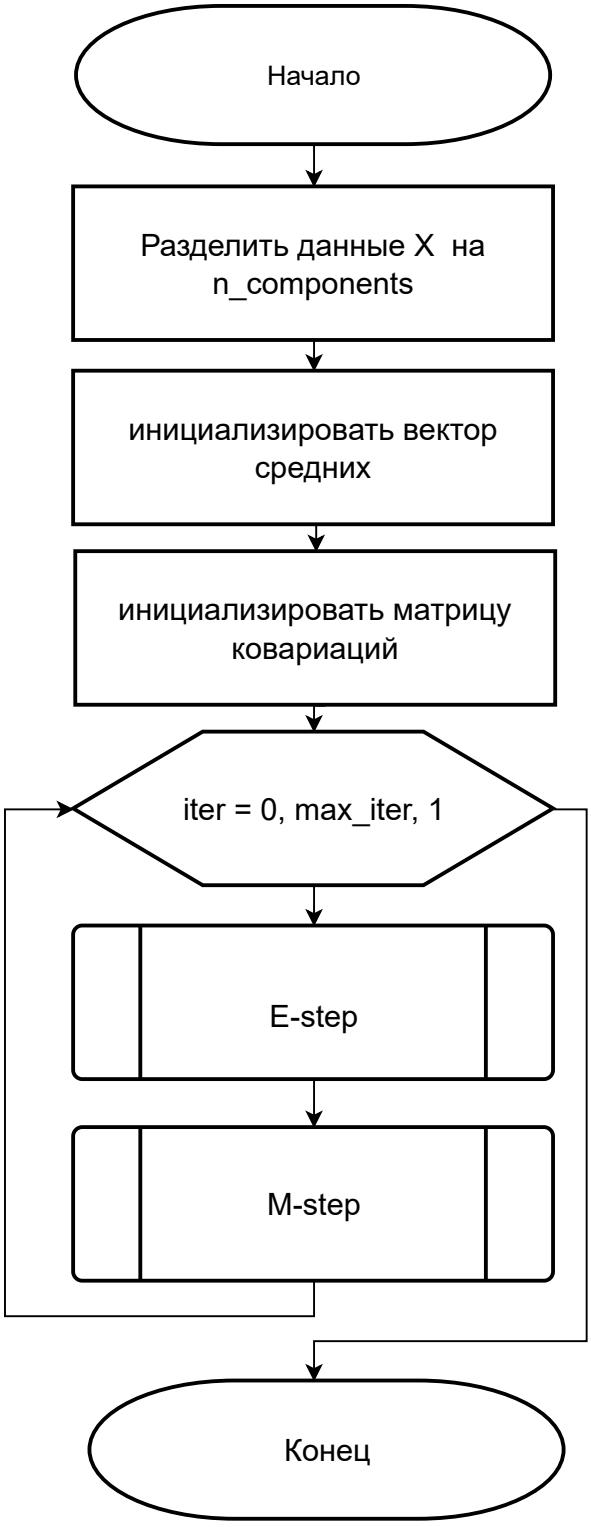

Рисунок 1.4 – Схема алгоритма метода максимизации ожидания для модели Гауссовой смеси

1.6 Рекомендация новостей на основе данных, выделенных в кластеры

Заключительным этапом разработки приложения, является создание рекомендательной системы новостей на основе данных, полученных после проведения нечеткой кластеризации, а именно предоставление пользователю набора новостей из того же кластера, что и выбранная им новость и кластеров, центры которых наиболее близки к текущему. Результатом данного этапа является рекомендательная ситема, предоставляющая пользователю рекомендации новостей на основе его персональных предпочтений.

1.6.1 Принцип работы рекомендательной системы

Принцип работы рекомендательной системы приведен на схеме алгоритма, изображенной на рисунке 1.5.

Рисунок 1.5 – Схема рекомендательной системы на основе контента

Для того, чтобы выполнить корректную рекомендацию требуется предварительно провести действия, описанные выше. Следовательно перед самой

рекомендацией новости требовалось предобработать для корректной работы алгоритма, а также провести понижение размерности для ускорения скорости работы, пожертвовав незначительным количество информации

1.7 Тестирование обученной модели

Поскольку у нас нет априорных данных о принадлежности новостей к категориям, требуется выбрать такой критерий оценки, который позволит оценить насколько объект похож на свой кластер по сравнению с другими кластерами. Данную задачу наилучшим образом решает метод оценки качества под названием Силуэт (Silhouette). Метод силуэтов — способ изучения разделительного расстояния между результирующими кластерами наблюдений, данная мера имеет диапазон [-1, 1]. Коэффициенты силуэта около +1 указывают на то, что образец находится далеко от соседних кластеров. Значение, близкое к нулю указывает, что выборка находится на границе принятия решения между двумя соседними кластерами или очень близко к ней, а отрицательные значения указывают на то, что эти выборки могли быть назначены неправильному кластеру.

Оценка для всей кластерной структуры приведена в формуле 13

$$Sil(C) = \frac{1}{N} \sum_{c_k \epsilon C} \sum_{x_i \epsilon c_k} \frac{b(x_i, c_k) - a(x_i, c_k)}{max\{a(x_i, c_k), b(x_i, c_k)\}}, \tag{13}$$

где $a(x_i, c_k) = \frac{1}{|c_k|} \sum_{x_j \epsilon c_k} ||x_i - x_j||$ — среднее расстояние от $x_i \epsilon c_k$ до других объектов из кластера c_k (компактность),

$b(x_i, c_k) = min_{c_l \epsilon C}\{\frac{1}{|c_l|} \sum_{x_j \epsilon c_l} ||x_i - x_j||\}$ — среднее расстояние от $x_i \epsilon c_k$ до объектов из другого кластера $c_l : k \neq l$ (отделимость).

По формуле 13 можно заметить, что

$-1 \leqslant Sil(C) \leqslant 1.$

И чем ближе данная оценка к 1, тем лучше.

Более того для тестирования реализованной модели нечеткой кластеризации и ее сравнения с моделью из библиотеки scikit-learn, потребуется оценка способная дать более точный результат называемая V-мерой, так как она ис-

пользует информацию о том к каким кластерам принадлежат данные.

V-мера представляет из себя гармоническое среднее оценки однородности и полноты. Вычисление V-меры представлено в формуле **??**

$$V - measure = 2 * \frac{h * c}{h + c}, \tag{14}$$

где h — это однородность, представленная в формуле 16, а c — полнота и представлена она в формуле 17.

Однородность измеряет, насколько образцы в кластере похожи и измеряется с помощью энтропии Шеннона. Вычисление однородности приведено в формуле 16, энтропия Шеннона для образцов с назначенным кластером C в кластере K приведена в формуле 15.

$$H(C|K) = -\sum \frac{n_{ck}}{N} \log(\frac{n_{ck}}{n_k}), \tag{15}$$

где n_{ck} — это количество образцов из кластера c в кластере k, n_k это общее количество образцов в кластере c, а N размер набора данных.

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{16}$$

Как можно заметить, если все образцы в кластере k имеют одинаковый назначенный кластер c, то однородность равна 1.

Полнота же измеряет, сколько похожих образцов объединяется алгоритмом кластеризации. Формула вычисления полноты приведена в формуле 17.

$$c = 1 - \frac{H(K|C)}{H(K)}, \tag{17}$$

где $H(K|C)$ отражает энтропию отношения образцов из кластера c в кластере k к общему количеству образцов c.

Если все образцы с назначенным кластером c назначены одному кластеру k, то полнота равна 1.

1.8 Выводы из конструкторского раздела

В данном разделе была представлена IDEF0 диаграмма разрабатываемой рекомендательной системы, далее были рассмотрены все этапы разработки, а именно:

- этап предобработки входных данных;
- этап векторизации предобработанных данных;
- этап понижения размерности матрицы признаков;
- этап разделения новостей на кластеры;
- этап рекомендации новостей.

Также была приведена схема алгоритма метода максимизации ожидания для модели Гауссовой смеси, а также схема алгоритма как шага ожидания, так и шага максимизации данного алгоритма. Приведена схема рекомендательной системы и описан метод тестирования обученной модели с использованием разных методов оценки.

2 Технологическая часть

В технологическом разделе описываются средства реализации, которые используются для разработки программного продукта, а также преимущества выбранных инструментов. Описывается структура разработанного ПО и приведен пользовательский интерфейс

2.1 Выбор средств разработки

В данном разделе рассматриваются такие инструменты разработки, как язык программирования, среда разработки и используемые библиотеки.

В первом подразделе будет описан язык программирования, который был выбран для реализации программного продукта, а также были описаны необходимые библиотеки для реализации поставленной задачи. Как для языка, так и для используемых библиотек были описаны преимущества их использования и обоснованность выбора именно данного языка и библиотек.

2.1.1 Язык программирования и используемые библиотеки

В качестве языка программирования для разработки программного продукта было принято решение использовать язык Python v3.9.2. Преимуществом данного языка является большое разнообразие представленных библиотек и фреймворков, а также обладает большой гибкостью и удобством использования.

При разработке программного обеспечения использовались следующие библиотеки и фреймворки:

- NumPy — библиотека, поддерживающая работу с большими многомерными массивами и матрицами, а также предоставляющая набор математических функций для работы с этими данными [2];
- Pandas — библиотека, предоставляющая удобные структуры и операции для работы с табличными даннами [3];
- Scikit-learn — данная библиотека включает в себя различные алгоритмы машинного обучения и подготовку данных для последующей; классифи-

кации, поддерживает взаимодействие с NumPy и Pandas [4];

- SciPy — библиотека для математического и числового анализа, такого как вычисление косинусной близости [5];
- NLTK — библиотека для работы с текстовыми данными, предоставляющая возможности обработки и лемматизации текста [6];
- Django — бесплатный высокоуровневый веб-фреймворк [7].

2.1.2 Среда разработки

В качестве среды разработки модуля рекомендательной системой было решено использовать IDE Jupyter Notebook так как он позволяет выполнять код по ячейкам, что очень удобно для разработки модулей, использующих методы машинного обучения. Для разработки графического интерфейса рекомендательной системы была использована IDE PyCharm, которая предоставляется студентам бесплатно по лицензии, а также предоставляет возможности создания виртуальной среды разработки для быстрой установки новых библиотек и модулей.

2.2 Структура разработанного ПО

Структура разработанного ПО в виде UML диаграммы представлена на рисунке 2.6

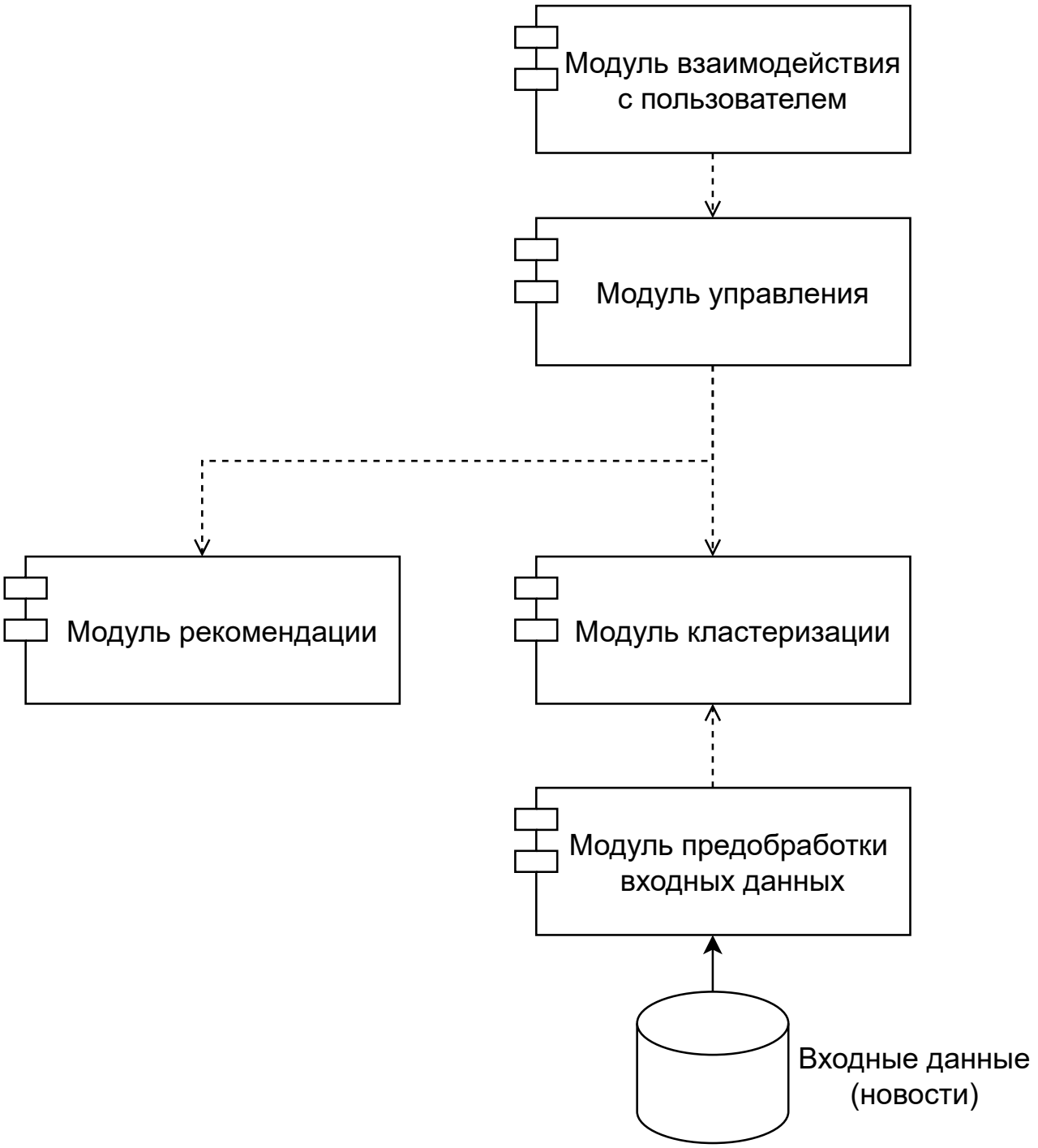

Рисунок 2.6 – UML диаграмма разработанного ПО.

Каждый из модулей, изображенный на диаграмме, содержит сгруппированные по функциональному значению соответствующие классы. Модуль взаимодействия с пользователем отвечает за пользовательский интерфейс. Модуль управления объединяет модуль рекомендации и модуль кластеризации и координирует их работу.

2.3 Пользовательский интерфейс

Интерфейс представляет из себя две веб-страницы с новостями. Веб-страница реализована на HTML с использованием CSS стилей. В качестве веб-фреймворка использовался Django.

Пример работы программы в случае если пользователь находится на главной странице и в случае если он находится на странице с новостью, представлены на рисунках 2.7, 2.8

Рисунок 2.7 – Пример работы программы главной страницы новостного сайта.

Рисунок 2.8 – Пример работы программы страницы с новостью.

2.4 Выводы из технологического раздела

В данном разделе были описаны средства реализации, которые были использованы для разработки ПО, после чего были приведены выбранные инструменты, обоснованы причины их использования и преимущества.

Была описана структура разработанного программного обеспечения и приведено описание каждого из модулей, приведенного на UML диаграмме. Также был приведен пользовательский интерфейс программы и продемонстрирован пример работы

3 Исследовательский раздел

В данном разделе проводится параметризация выбранных алгоритмов векторизации, понижения размерности и модели нечеткой кластеризации. Проведено сравнение модели Гауссовой смеси собственной реализации и модели из библиотеки scikit-learn [8]. Также приведены эксперименты, на основе которых были выбраны наиболее оптимальные параметры для решения данной задачи.

3.1 Выборка данных

Датасет собран из новостей на английском языке с сайта MSN.com. [1]

На вход векторизатору подается выборка данных размером 98247 документов, каждая из которых представляет собой предобработанную строку на английском языке, а также дату публикации.

Данные хранятся в файле имеющем формат tsv — это формат для представления таблиц баз данных. Для последующего обучения модели нечеткой кластеризации использовалась полная выборка данных, так как каждой новости требуется сопоставить кластер для корректной работы алгоритма.

3.2 Сравнение методов

В данном разделе будет произведено сравнение собственной реализации модели Гауссовой смеси и модели из библиотеки scikit-learn. Будет создан искусственный набор данных имеющий 3 кластера. После выполнения кластеризации собственным методом и методом из библиотеки scikit-learn будет произведена оценка кластеризации V-мерой, которая представляет из себя гармоническое среднее оценки однородности и полноты.

На рисунке 3.9 приведен исходный размеченный набор данных размером 400, а на рисунках 3.10 и 3.11 приведен набор данных кластеризованный собственным методом Гауссовой смеси и методом из библиотеки scikit-learn.

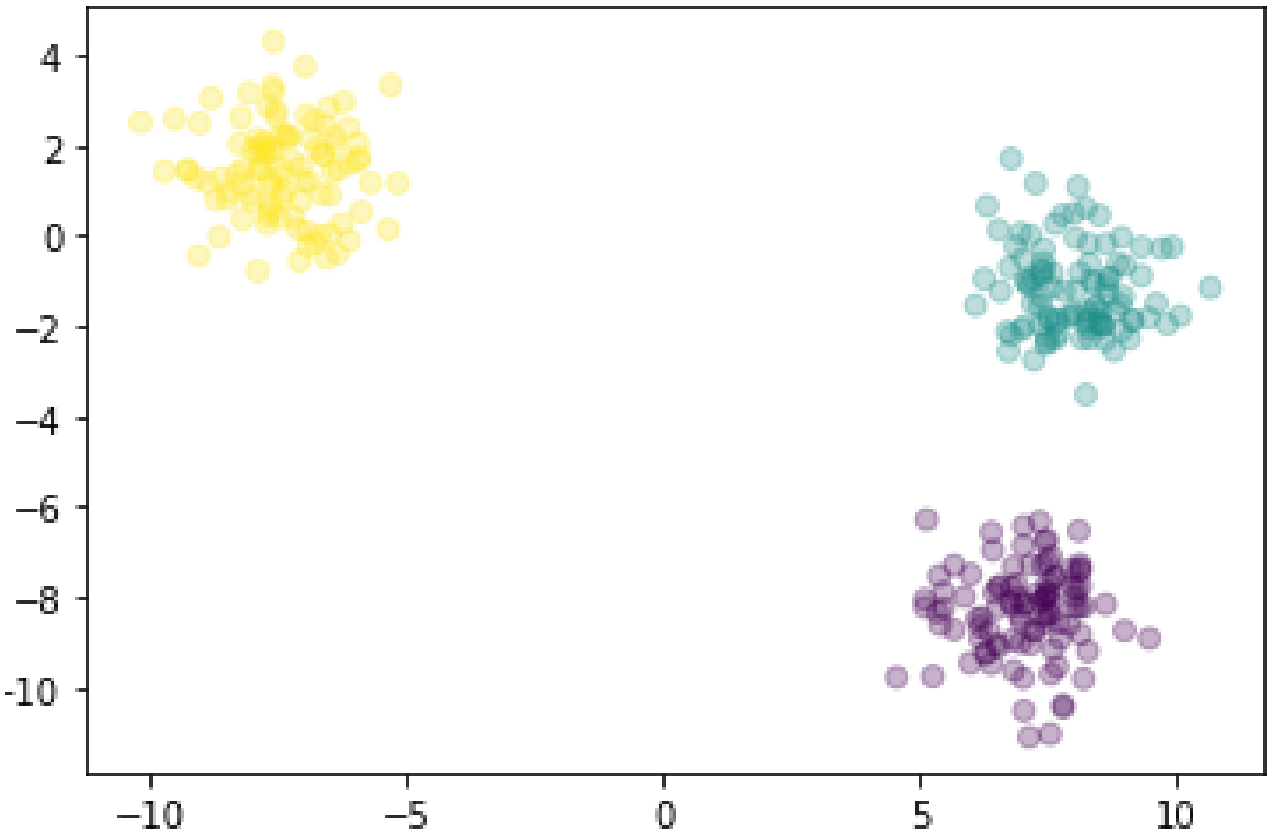

Рисунок 3.9 – График исходного набора данных. Цветами выделены кластеры.

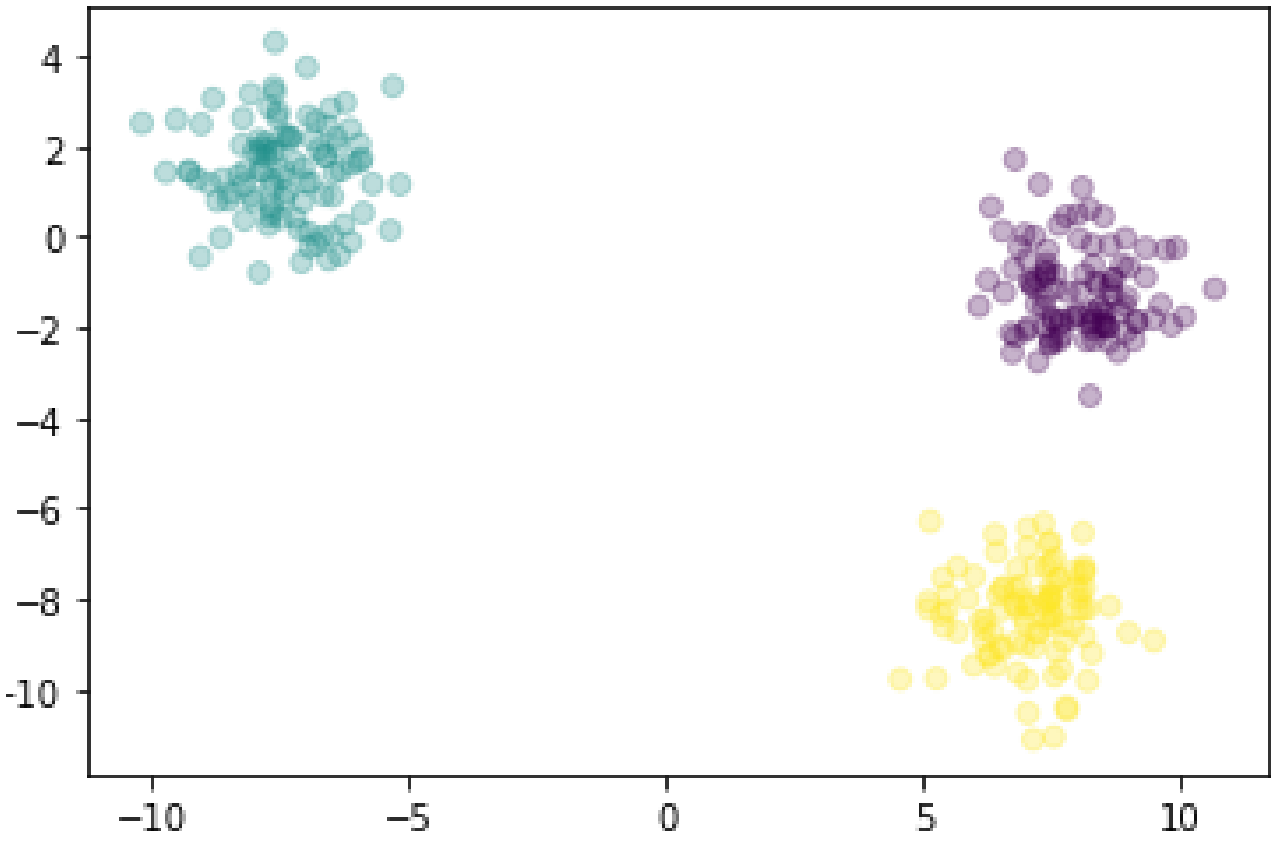

Рисунок 3.10 – График набора данных, кластеризованный собственным методом Гауссовой смеси.

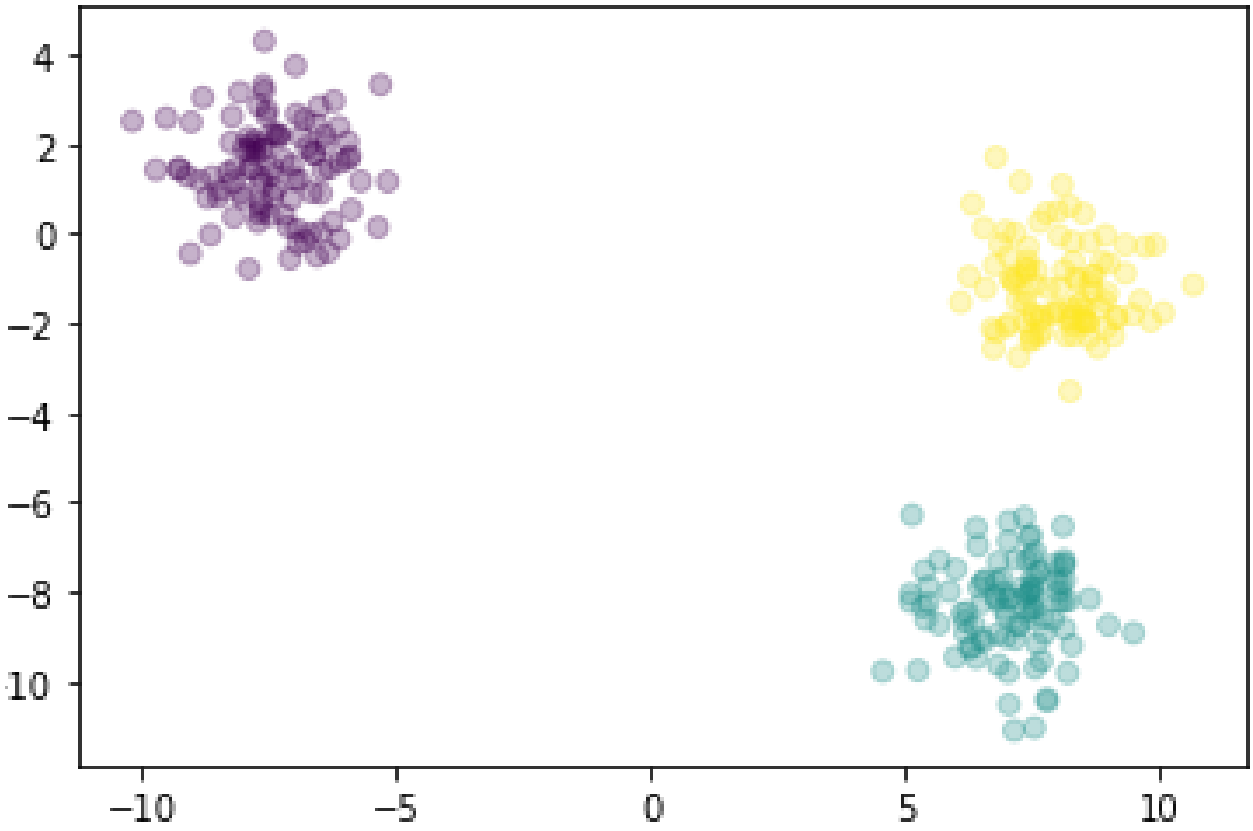

Рисунок 3.11 – График набора данных, кластеризованный методом Gaussian Mixture Model из библиотеки scikit-learn.

Как можно заметить алгоритм собственной реализации корректно разделяет данные. Заключительным этапом является оценка кластеризации V-мерой, результат оценки приведен в таблице 2

Таблица 2 – Сравнение реализованного метода Гауссовой смеси с методом из библиотеки scikit-learn

Количество образцов	Значение V-меры реализованного метода	Значение V-меры метода из scikit-learn
100	0.930	0.965
250	0.930	0.930
300	0.940	0.965

Как можно заметить по таблице, метод собственной реализации лишь незначительно уступает в точности методу из библиотеки scikit-learn, а при значении 2500 совпадает.

3.3 Порядок параметризации

Для достижения наилучшего качества нечеткой кластеризации и последующих рекомендаций в первую очередь следует определить оптимальные параметры векторизатора TF-IDF, после чего требуется определить оптимальное количество компонентов, которые следует оставить для того, чтобы доля объясненной дисперсии после понижения размерности была не меньше 0.95. Заключительным этапом является подбор оптимальных параметров модели Гауссовой смеси для достижения наилучшего качества.

3.3.1 Параметризация векторизатора TF-IDF

Для получения наилучших параметров векторизатора будем варьировать максимальную величину параметра DF (документная частота, то есть максимальное допустимое число документов, в которых встретился термин t), термины выше порогового значения не будут использоваться при векторизации, также будет варьироваться диапазон N-грамм, то есть будут ли признаки формироваться из отдельных слов или из нескольких.

Качество векторизатора будем проверять с помощью оценки последующей нечеткой кластеризации мерой Силуэт. Параметры метода понижения размерности и нечеткой кластеризации заданы по умолчанию. Результат исследования приведен на таблице 3.

Таблица 3 – Значение критерия Силуэт от параметров векторизатора TF-IDF

используемые n-граммы / Max_df	Слова по отдельности	Слова по отдельности и пары слов
0.25	0.912	0.923
0.30	0.905	0.919
0.35	0.903	0.918

Посмотрев на таблицу выше можно сделать вывод о том, что наилучшими значениями параметров векторизатора являются 0.25 для параметра max_df и использование как всех слов по отдельности так и пар слов.

3.3.2 Параметризация метода понижения размерности SVD

Так как исходная выборка обладает большой размерностью, а также является разреженной требуется применить метод понижения размерности в целях экономии памяти и увеличения скорости работы. Целевым показателем является доля объясненной дисперсии, ее значение должно равняться не менее 0.95.

На рисунке 3.12 приведена зависимость количества компонентов после понижения размерности и долю объясненной дисперсии для данного количества компонент.

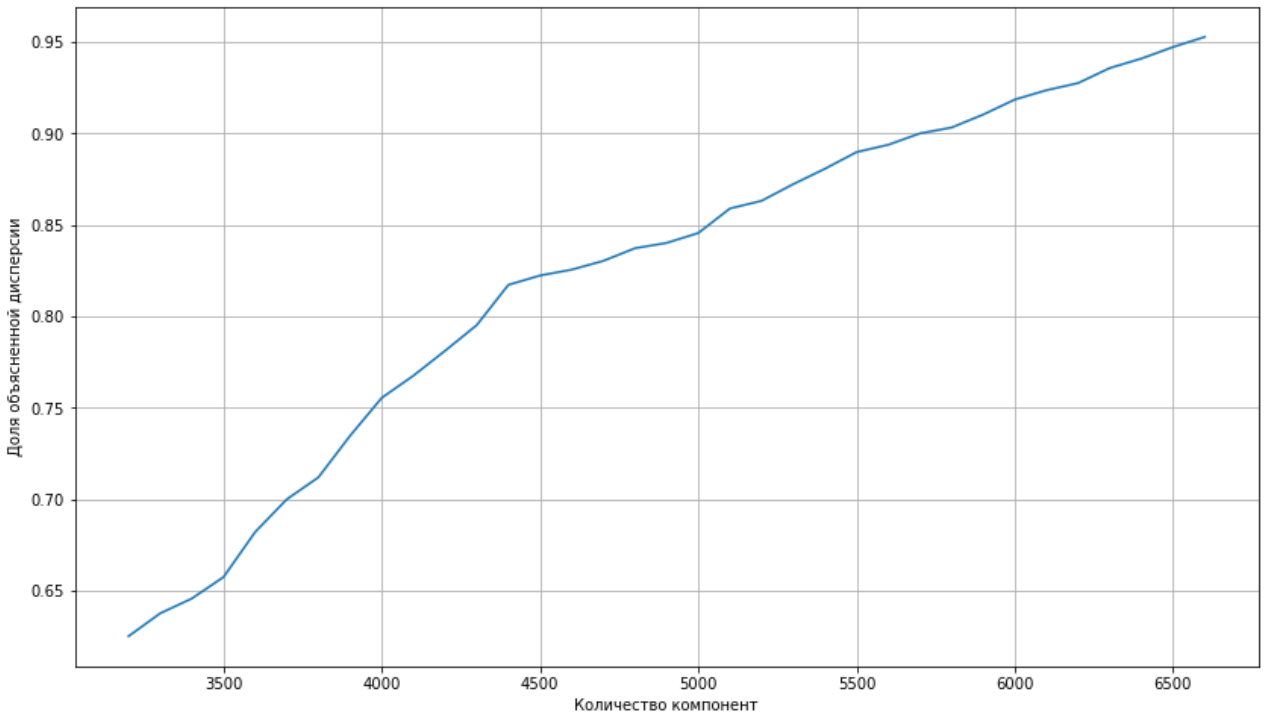

Рисунок 3.12 – Доля объясненной дисперсии в зависимости от количества компонент.

Как можно заметить на картинке вышее необходимое количество компонент составляет 6700, при данном количество доля объясненной дисперсии составляет чуть выше 0.95. Данный этап позволил сохранить информативность данных и сэкономил большой количество ресурсов памяти.

3.3.3 Параметризация метода Гауссовой смеси

Для достижения наилучшего результата нечеткой кластеризации требуется варьировать ключевой параметр, которым является количество кластеров, требуется получить такое количество кластеров при котором оценка методом Силуэта будет давать наивысший результат (быть как можно более близким к 1). Для достижения необходимого результата был проведен эксперимент при котором варьировалась количество кластеров. Зависимость полученной оценки от количества кластеров можно увидеть на рисунке 3.13

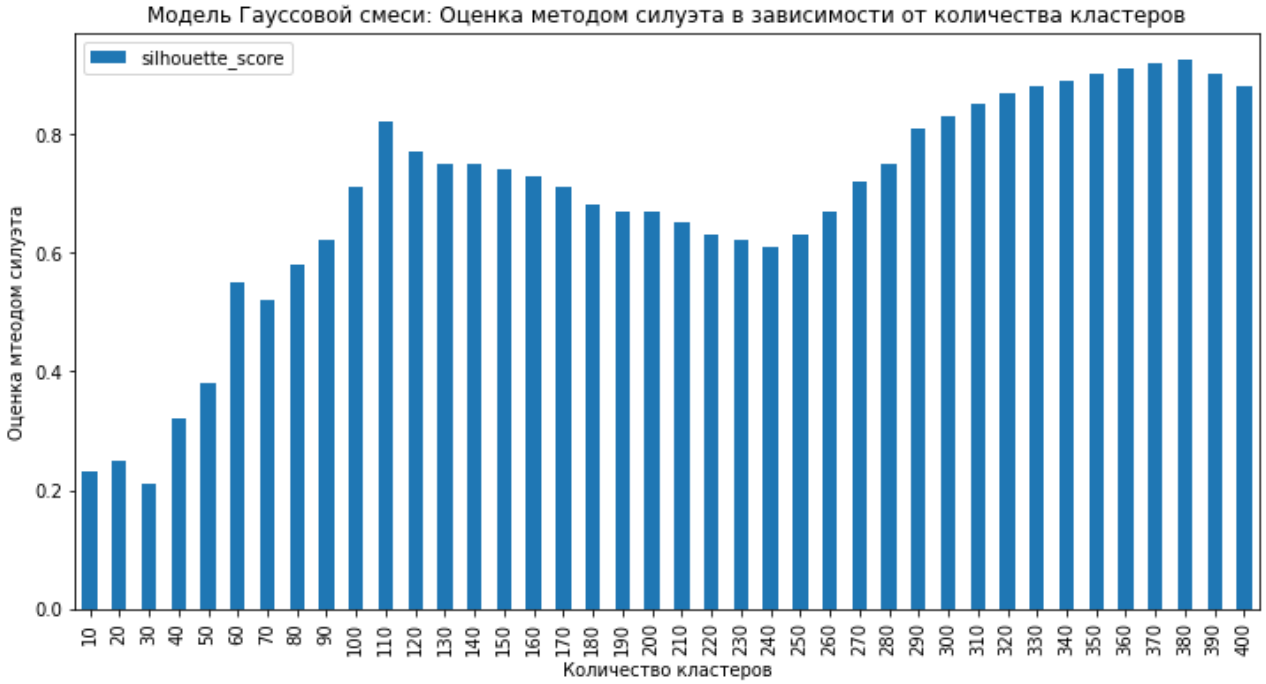

Рисунок 3.13 – Полученная оценка методом Силуэта в зависимости от количества кластеров.

Видно, что явные пики на графике это 110 кластеров и 380 кластеров. При количестве кластеров равным 380 оценка достигает порядка 0.92 по методу силуэта, что является очень хорошим показателем, именно данное количество кластеров и было использовано при построении финальной версии рекомендательной системы.

3.4 Рекомендации к применению рекомендательной системы

После проведения исследования и параметризации, полученной рекомендательной системы можно сделать вывод о том, что она работает эффективно и исправно. Данную систему можно применять при создании собственных новостных сайтов, но стоит учесть, что первоначальное обучение модели на уже имеющихся данных должно выполняться на мощной машине и кластеры новостям должны быть определены заранее, поскольку векторизатор, метод понижения размерности и сама нечеткая кластеризация методом Гауссовой смеси являются трудоемкими операциями в случае большого объема данных.

3.5 Выводы из исследовательского раздела

В данном разделе была проведена парметризация алгоритмов из ключевых этапов построения рекомендательной системы, а также проведена проверка качества работы разработанного метода в сравнении с методом из библиотеки scikit-learn с помощью V-меры, в ходе которой было выявлено, что разработанный метод в большинстве случаев не уступает методу из библиотеки. Параметризовав алгоритм векторизации был сделан вывод о том, что наилучшим значением для параметра max_df векторизатора Tf-IDF является 0.25, а в качестве n-грамм следует брать как слова по отдельности так и пары слов. Для метода понижения размерности наилучшим количеством компонент является 6700, поскольку доля объясненной дисперсии при данном количестве превышает 0.95, что очень хорошо описывает данные, экономя при этом память. Заключительным этапом было выявление оптимального количества кластеров в модели Гауссовой смеси, в качестве результата была получена зависимость оценки метода Силуэта от количества кластеров, исходя из которой можно сделать вывод о том, что наилучшее количество кластеров для построения рекомендательной системы на исходных данных равно 380. В конце данного раздела была описана применимость разработанной рекомендательной системы.

ЗАКЛЮЧЕНИЕ

В рамках данной выпускной квалификационной работы была разработана и реализована рекомендательная система новостей на основе нечеткой кластеризации.

В результат проделанной работы были выполнены следующие задачи и достигнуты следующие результаты:

- разработана рекомендательная система на основе нечеткой кластеризации методом Гауссовой смеси;
- сконструировано и разработан программное обеспечение, демонстрирующее работы данного метода;
- проведена параметризация ключевых алгоритмов, используемых при разработке программного обеспечения;
- разработанная система исследована на предмет применимости.

В качестве направлений дальнейшей работы над методом можно выделить следующие:

- использовать другие, более ресурсоемкие способы векторизации, такие как Doc2Vec;
- применить нейронную сеть, взяв в качестве слоя веса, являющиеся центрами кластеров полученные после обучения модели Гауссовой смеси.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. MSNnews [Электронный ресурс]. – Режим доступа: https://msnews.github.io – Дата обращения: 03.04.2022

2. NumPy [Электронный ресурс]. – Режим доступа: https://numpy.org – Дата обращения: 05.04.2022

3. Pandas [Электронный ресурс]. – Режим доступа: https://pandas.pydata.org – Дата обращения: 05.04.2022

4. Scikit-learn [Электронный ресурс]. – Режим доступа: https://scikit-learn.org – Дата обращения: 05.04.2022

5. SciPy [Электронный ресурс]. – Режим доступа: https://www.scipy.org – Дата обращения: 05.04.2022

6. NLTK [Электронный ресурс]. – Режим доступа: https://www.nltk.org – Дата обращения: 05.04.2022

7. Django [Электронный ресурс]. – Режим доступа: https://www.djangoproject.com – Дата обращения: 05.04.2022

8. Scikit-learn Gaussian Mixture [Электронный ресурс]. – Режим доступа: https://scikit-learn.org/modules/generated/sklearn.mixture.GaussianMixture.html – Дата обращения: 05.04.2022