Исследование и разработка программного обеспечения для поддержки принятия решения при выборе вина.

2019г.

Оглавление

Введение

С каждым годом становится все больше интеллектуальных систем, призванных не только помочь человеку, но и заменить его как в профессиональной среде, так и в обычной жизни. Наиболее яркими примерами являются рекомендательные системы магазинов, которые помогают человеку подобрать подходящий продукт.

Аналогичная ситуация обстоит и с рынком вина. За последние 20 лет потребление выросло в несколько раз, и связано это с тем, что вино, в отличие от других спиртных напитков, имеет наименьшее негативное влияние на здоровье человека, более того, в малых количествах оно оказывает положительное влияние на сердечно-сосудистую систему человека. Однако из-за сложности выбора вина, связанного с большим количеством факторов, влияющих на характеристики вина, многие потребители алкоголя предпочитают вину другие спиртосодержащие продукты.

Существует множество методов, позволяющих разрабатывать рекомендательные системы, однако многие из них не гарантируют высокое качество рекомендации. В области вина ситуация с программным обеспечение, которое работает с винными данными весьма плачевна. На сегодняшний день существует популярное приложение “Vivino”, которое при помощи камеры мобильного устройства позволяет получить информацию о вине, считывая изображение этикетки бутылки. Однако среди приложений с рекомендательной системой существует лишь одно программное обеспечение под название “Hello Vino”

Рекомендательная система – это система, которая, использую пользовательскую информацию и данные исходных объектов, пытается предсказать, какие из этих объектов будут интересны пользователю.

Целью данной работы является создание возможности автоматизированного подбора вина на основе ранее оцененных вин. Для достижения поставленной цели необходимо решить следующие задачи:

1. Подготовить теоретическую базу для рекомендательной подсистемы.
2. Выбрать методы распознавания текста.
3. Определить какие данные будут влиять на рекомендации.

4. Выбрать средства реализации поставленных задач.

5. Выбрать источники данных о винах.

6. Разработать рекомендательную систему.

7. Разработать интерфейс для взаимодействия с пользователем.

8. Добавить возможность поиска данных о вине при помощи камеры.

9. Выполнить тестирование системы.

Актуальность работы состоит в том, что существующие системы рекомендаций имеют низкое качество результатов работы, а в такой популярной сфере, как вино, и вовсе являются не работающими. Новизна работы заключается в создании нового метода автоматизированной рекомендации вин.

Главным результатом работы является разработанная интеллектуальная система, способная производить обработку неоднозначных критериев для дальнейшей рекомендации вин.

Объектами исследования являются системы поддержки принятия решений и энология [1]. Практическая значимость заключается в применении полученного метода на практике.

1 Анализ существующих технических решений

1.1 Wine Spectator WineRatings+

Данное приложение является бесплатным с возможностью покупки дополнительного контента. Приложение доступно на Android [2] и IOS [3] и содержит следующие функции [4]:

Бесплатный контент:

— Новости.

— Винтажные диаграммы.

Премиум контент:

— Регулярно обновляемые рейтинги критиков.

— Быстрый и легкий поиск рейтингов вина.

— Экспертный выбор.

— Виртуальный погреб.

1.2 Delectable Wine

Данное программное обеспечение имеет схожий с предыдущим функционал, за исключением того, что данное приложение направлено на получение информации о конкретном вине, а не о культуре вина в целом. Отличительными чертами являются [5]:

— Возможность сканирование этикетки через камеру.

— Работа не только с вином, но и с другими спиртосодержащими продуктами.

1.3 Vivino Wine Scanner

Самое популярное приложение о вине на сегодняшний день. Данное приложение призвано простым языком объяснить сложные винодельческие термины. Приложение базируется на отзывах пользователей, что, по мнению разработчиков [6], позволяет охватить практически все вина мира и избавиться от сложных для простого человека терминов. Основные функции данного приложения [7]:

— Возможность сканирования этикетке на бутылке, чтобы моментально получить всю информацию о вине

— Возможность сканирования винной карты в ресторане, чтобы выбрать лучшее предложение

— Работа с социальными сетями Facebook [8], Twitter [9] и Gmail [10], чтобы отслеживать выбор друзей.

— Список желаний.

— Поиском вин на основе оценок и рецензий от 20 миллионов пользователей по всему миру.

1.4 Hello Vino

Единственное из представленных на рынке приложений, которое имеет функцию рекомендации вина пользователю [11]. Данная возможность построена на вкусовых предпочтениях пользователя, основанного на любимых ягодах, а также его история поиска (рис. 1).

Рисунок 1. Выбор ягод в Hello Vino.

Данное приложение не имеет однозначной оценки, так как данное программное обеспечение ориентированно на американский рынок и содержит, по большей части, вина штата Калифорния. Более того, подход, основанный на любимых ягодах, не способен однозначно порекомендовать вино, так как существует множество

факторов, которые влияют на вино [12]. Однако данный недостаток исправляется использованием истории поиска, но и это не может гарантировать качество рекомендаций, если использовать приложение в кампании других людей.

К сожалению, отсутствует какая-либо информация о том, как выбор ягод и история просмотров влияет на рекомендации, и не имеется возможность проверить корректность работы данного приложения, но на основе отзывов пользователей данного приложения, взятые из магазинов App Store и Google Play, можно сделать вывод, что данный метод с одной стороны не способен выдавать рекомендации, а с другой стороны имеет высокое качество рекомендаций. Статистика приведена ниже (рис. 2).

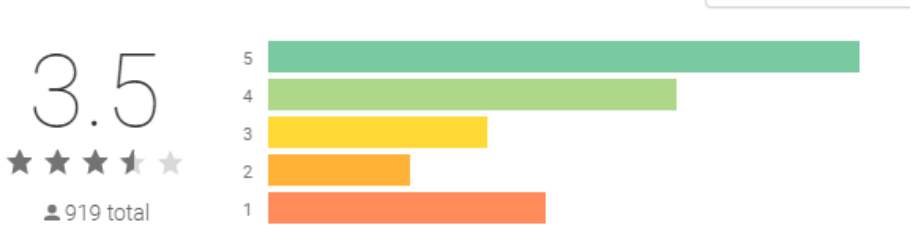

Рисунок 2. Статистика приложения в Google Play.

Существует еще несколько приложений, которые так или иначе взаимодействуют с вином, однако они являются более простыми аналогами уже описанных приложений, без функционала рекомендательной системы. Более того, несколько более-менее популярных в свое время приложений удалены из магазинов, или не поддерживаются разработчиками.

1.5 Выводы

В ходе проведенного анализа выявлено, что на сегодняшний день приложения, которые позволяют пользователю так или иначе выбрать вино являются актуальными. Лишь единственно приложение Hello Vino способно выдавать рекомендации, однако оно не имеет однозначной оценки в качестве рекомендаций.

2 Аналитический обзор методов построения рекомендательных систем

Все существующие на сегодняшний момент рекомендательные системы построены при помощи науки о данных и многие из них попадают под понятие искусственный интеллект. Главной основой любой модели рекомендательной системы является алгоритм, основанный на различных математических теоремах. Главными математическими областями являются статистика и теория вероятностей.

На сегодняшний день проведен обзор множества алгоритмов машинного обучения для задач разработки систем поддержки принятия решений, каждый из которых имеет свои преимущества и недостатки. Рассмотрим наиболее популярные из них.

2.1 Описание различных подходов

Существует несколько подходов работы рекомендательной системы [13]:

- Коллаборативный метод (collaborative filtering);
- Метод, основанный на содержании (content - based);
- Подход, основанный на знаниях (knowledge - based);
- Гибридная фильтрация (hybrid).

2.1.1 Коллаборативнаый метод

Данный тип фильтрации строит прогнозы на основе модели уже совершенных действий на сайте или определенных характеристик пользователя. Эта модель может быть построена не только на поведении конкретного пользователя, но и с учетом поведения пользователей со схожими параметрами.

Данный метод работает на основе действий, которые совершил пользователь ранее. Помимо этих действий, анализируются действия пользователей, которые имеют схожие параметры.

Коллаборативную фильтрацию можно разделить на три типа [13]:

- Соседство (neighborhood - based);
- Модель (model - based);
- Гибридные модели.

Принцип соседства является первым подходом, который строит рекомендации на основе оценок пользователя. При этом сама рекомендация строится исходя из вычисления меры схожести данных, которые были получены при проведении анализа оценок пользователя.

Этот метод можно разделить на два подтипа [13]:

— На основе пользователей (user - based);

— На основе элементов (item - based).

В первом случае система сравнивает пользователей между собой и использую их сходства предоставляет рекомендацию на основе выбора другого пользователя. Рассмотрим пример пользователей и их оценок в виде матрицы в таблице 1.

Таблица 1. Пример матрицы предпочтений.

Пользователи	Музыка		
	Рок	Рэп	Классическая музыка
Николай	5	-	8
Софья	5	7	3

В данной таблице по строкам находится вектор пользовательских оценок, а по столбцам огладываются оценки пользователей для каждого типа объекта. Следовательно, для нахождения рекомендации берется вектор одного пользователя и сравнивается с вектором других пользователей. Существует три шага вычисления рекомендации:

1) Вычисление совпадений пользователей (вес);
2) Составление списка пользователей с похожими весами;
3) Вычисление рекомендации для пользователя по оценкам из списка пользователей с похожими весами.

Для первого этапа применяется метод кластерного анализа. Количество сходств и различий определяется как метрическое расстояние между точками значений оценки. Для вычисления списка схожих пользователей существует множество алгоритмов, рассмотрим наиболее популярные [13]:

— евклидово расстояние;

— коэффициент корреляции Пирсона;

— манхэттенское расстояние;

— коэффициент Жаккара;

— расстояние Чебышева.

Евклидово расстояние – геометрическое расстояние между двумя точками в многомерном пространстве, вычисляемое по теореме Пифагора и выглядит он следующим образом:

$$r1(X1, X2) = \sqrt{\sum_{k=1}^{m}(X1_k - X2_k)^2},$$

где X1 и X2 – пользователи, k – конкретный объект, m – количество объектов, $X1_k$ и $X2_k$ – оценка пользователя для k-го объекта (как и для формул далее). Более того, есть возможность применить квадрат евклидова расстояния для того, чтобы установить широкий диапазон весов схожести пользователей:

$$r2(X1, X2) = \sum_{k=1}^{m}(X1_k - X2_k)^2,$$

Коэффициент корреляции Пирсона – является точнее, чем евклидово расстояние и определяет линейную зависимость между:

$$r3(X1, X2) = \frac{\sum_{k=1}^{m}(X1_k - \overline{X1}) * (X2_k - \overline{X2})}{\sqrt{\sum_{k=1}^{m}(X1_k - \overline{X1})^2 * \sum_{k=1}^{m}(X2_k - \overline{X2})^2}},$$

Значение может меняться от -1 до 1 и чем ближе к единице, тем больше схожесть двух пользователей.

Манхэттенское расстояние – метод, который достаточно просто вычислить:

$$r4(X1, X2) = \sum_{k=1}^{m}|X1_k - X2_k|,$$

Коэффициент Жаккара – применяется для конечных неотрицательных множеств:

$$r5(X1, X2) = \frac{\sum_{k=1}^{m}\min(X1_k, X2_k)}{(\sum_{k=1}^{m}(X1_k) + \sum_{k=1}^{m}(X) - \sum_{k=1}^{m}\min(X1_k, X2_k)},$$

Расстояние Чебышева – максимум модуля разности соответствующих признаков двух объектов:

$$r6(X1, X2) = \max_{k=1..m}|X1_k - X2_k|,$$

На втором шаге используется метод машинного обучения - кластеризация. Различают иерархические и не иерархические методы. Но иногда проще ввести порог меры близости, который находит «соседей» среди тех, кто переходит данный порог

Иерархическую кластеризацию можно представить как дерево взвешенных кластеров (рисунок 3).

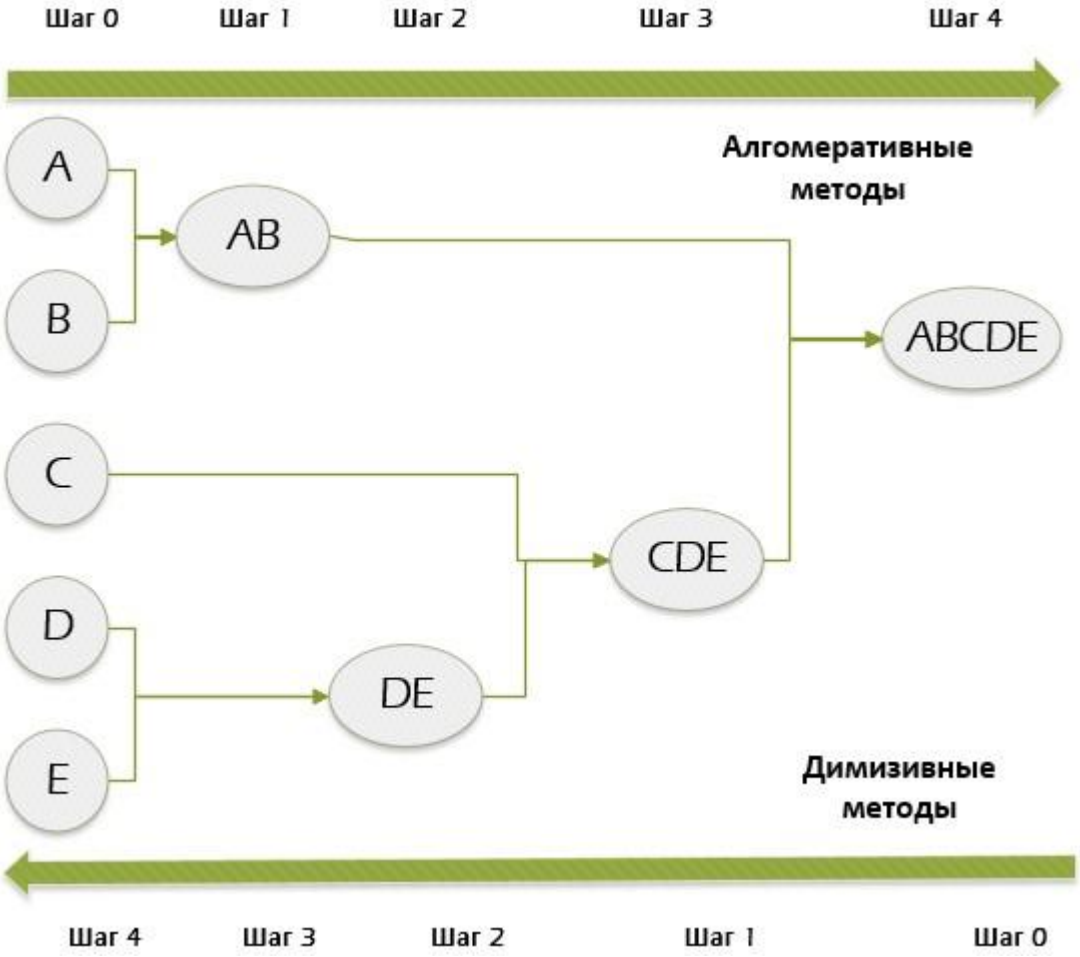

Рисунок 3. Пример дерева взвешенных кластеров.

Дерево взвешенных кластеров бывает двух типов [13]:

— Агломеративным (снизу - вверх),

— дивизимным (сверху - вниз).

Агломеративной кластеризации начинает разбиение с кластеров, содержащих по одному объекту, и осуществляет последовательное объединение ближайших друг к другу кластеров.

Дивизимная кластеризация начинается с объекта, в котором находятся все кластеры и затем последовательно осуществляется отделение наиболее отдаленных

объектов. Для расчета расстояний между кластерами применяется несколько алгоритмов:

— Метод полной связи, где расстояние между кластерами является расстоянием между возможными парами объектов, находящихся в разных кластерах:

$$Kr(X1, X2) = \max r(X1, X2),$$

Визуализация кластерной структуры метода (рисунок 4 и рисунок 5)

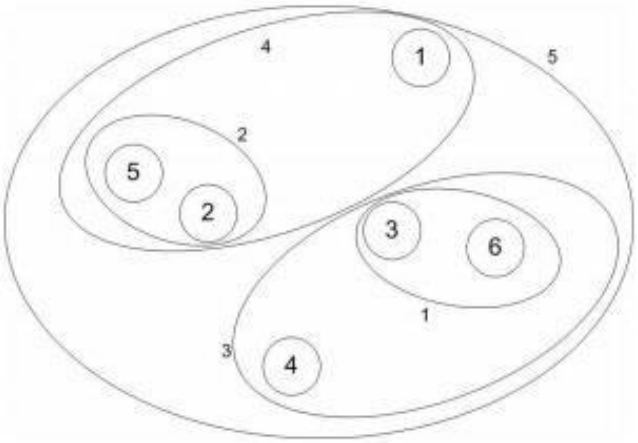

Рисунок 4. Вложенная диаграмма.

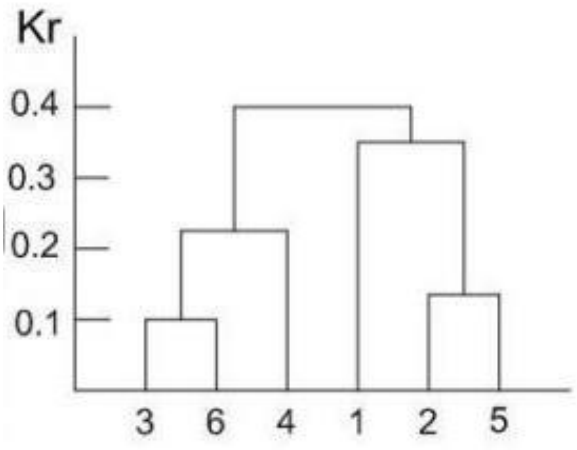

Рисунок 5. Дендрограмма.

— Метод одиночной связи, в котором, в отличие от метода полной связи, принимается минимальное расстояние:

$$Kr(X1, X2) = \min r(X1, X2),$$

Визуализация кластерной структуры метода (рисунок 6 и рисунок 7)

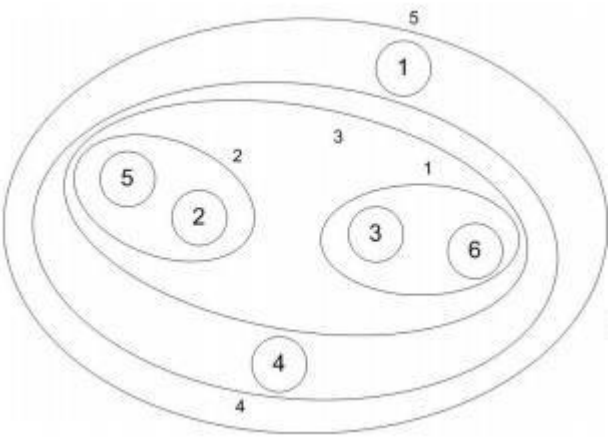

Рисунок 6. Вложенная диаграмма.

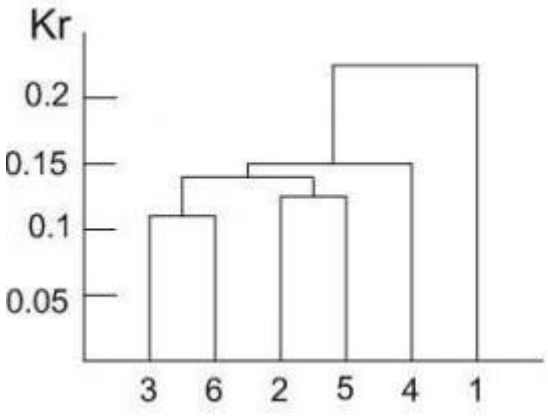

Рисунок 7. Дендрограмма.

— Методе средней связи расстояние между объектами рассчитывается, как среднее расстояние между всеми возможными парами объектов, принадлежащих разным кластерам (рисунок 8 и рисунок 9):

$$Kr(X1, X2) = \text{avgr}(X1, X2),$$

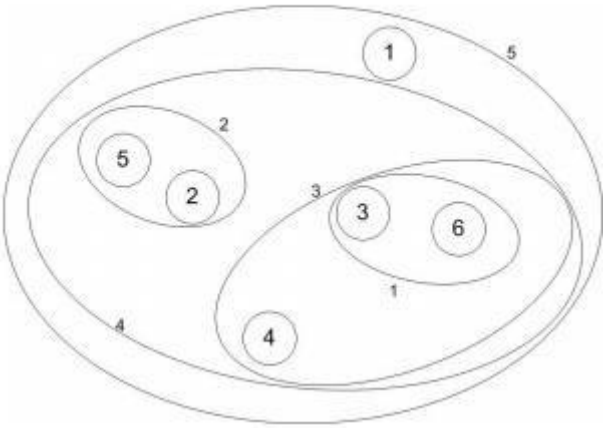

Рисунок 8. Вложенная диаграмма.

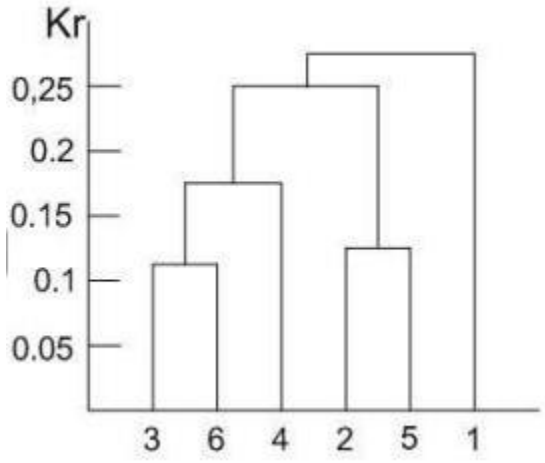

Рисунок 9. Дендрограмма.

— Метод Уорда, в котором расстояние между кластерами является приростом суммы квадратов расстояний объектов до центров кластеров, получаемых в результате объединения (рисунок 10 и рисунок 11):

$$Kr(X1, X2) = \frac{|X1 * X2|}{|X1| + |X2|} r^2 \left(\sum_{x1 \in X1} \frac{x1}{|X1|} * \sum_{x2 \in X2} \frac{x2}{|X2|}\right),$$

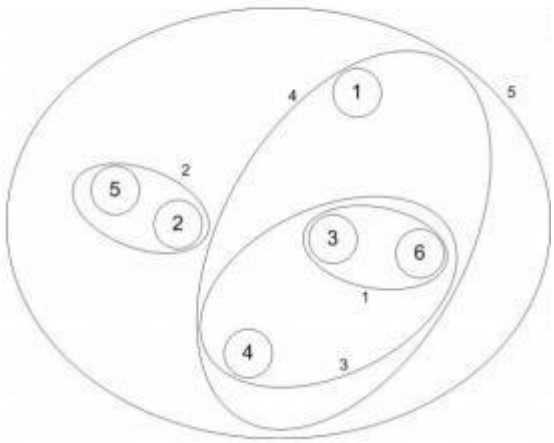

Рисунок 10. Вложенная диаграмма.

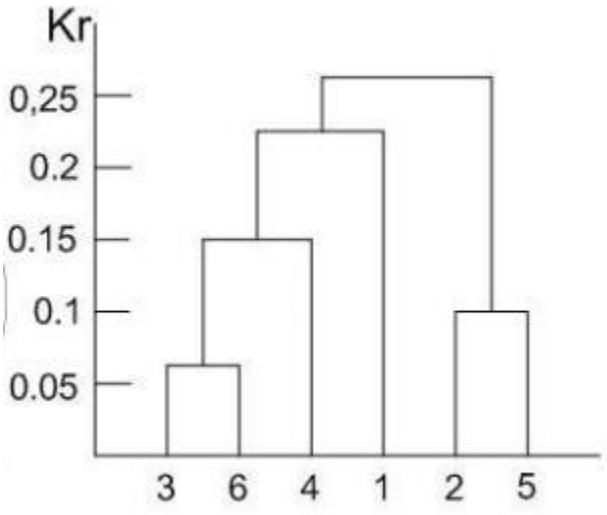

Рисунок 11. Дендрограмма.

Для работы с не иерархичным кластеризациями применяются следующие методы [13]:

1. К – средних:
2. Выбор количества пользователей;
3. Выбор начальных центров кластеров;
4. Распределение объектов по ближайшим центрам;
5. Перерасчёт центров кластеров;
6. Проверка граничных условий;
7. Если граничное условие не достигнуто, то повторять этапы 3,4,5.

1) K - metoids. Центром кластера является точка, которая равноудалена от других точек.

2) Алгоритм QT:

1. Выбор радиуса кластера;

2. Вычисляются кластеры – кандидаты. В каждом кластере-кандидате одна из точек является центром. В кластере – кандидате попадают точки, отстающие от центра не более, чем на радиус кластера;

3. Выбор кластера – кандидата, содержащего наибольшее число точек. Выбранный кластер является построенным в точке кластера, исключающейся из анализа;

4. Повторять шаги 2 и 3, пока не будут обработаны все точки;

Теперь рассмотрим метод на основе схожести элементов. Метод аналогичен рассматриваемому ранее, однако вместо пользователей рассматриваются объекты и вычисляется средняя оценка аналогичных объектов:

$$O = \frac{\sum_{c=1}^{m} X_c * r(k1,kc)}{\sum_{c=1}^{m} r(k1,kc)},$$

где k1 и kc являются объектами, m – количество объектов, а Xc – оценка аналогичных объектов.

Коллаборативная фильтрация на основе модели заключается в том, что строится модель по совокупности оценок, на основании которых формируются рекомендации. Модель может быть реализована при помощи следующих методов:

— Модель Байеса;

— Кластерный анализ (описан ранее);

— Латентный семантический анализа (LSA);

— Сингулярное разложение (SVD).

1) Модель Байеса создается для каждого пользователя:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)},$$

Рассмотрим пример в таблице 2.

Таблица 2. Пример оценок пользователя.

Порядковый номер	Объект	Оценка
1	Тетрадь	5
2	Ручка	3
3	Карандаш	9

Продолжение таблицы 2

Порядковый номер	Объект	Оценка
4	Карандаш	6
5	Ручка	7
6	Карандаш	2
7	Карандаш	8
8	Ручка	4
9	Тетрадь	4
10	Тетрадь	10

Преобразуем ее в частотную таблицу с тремя диапазонами оценок (таблица 3).

Таблица 3. Пример частотной таблицы.

	<5	≥5, <8	≥8
Карандаш	1	1	2
Ручка	2	1	0
Тетрадь	1	1	1

Затем постоим таблицу вероятностей (таблица 4).

Таблица 4. Пример таблицы вероятностей.

	<5	≥5, <8	≥8	Всего
Карандаш	¼	1/3	2/3	0,4
Ручка	2/4	1/3	-	0,3
Тетрадь	¼	1/3	1/3	0,3
Всего	0,4	0,3	0,3	

Посчитаем вероятность того, что пользователь поставит выше восьми баллов объекту «Карандаш».

$$\text{P}(\geq 8|\text{Карандаш}) = \frac{\text{P}(\text{Карандаш}|\geq 8)*\text{P}(\geq 8)}{\text{P}(\text{Карандаш})},$$

$$\text{P}(\text{Карандаш}| \geq 8) = \frac{2}{3},$$

$$\text{P}(\geq 8) = \frac{3}{10},$$

$$\text{P}(\text{Карандаш}) = \frac{3}{10},$$

$$\text{P}(\geq 8|\text{Карандаш}) = \frac{\frac{2}{3}*0{,}3}{0{,}3},$$

Отсюда следует, что объект будет «Карандаш» с вероятностью 0,3. А вероятность того, что он поставит в нем больше восьми баллов примерно равно 0,67.

2) Методы латентного семантического анализа применяется для работы со словами и терминами объекта в матрицу, в которой по строкам идут слова и термины, а по столбцам объекты. Для работы с этой матрицей используется сингулярное разложение.

3) Сингулярное разложение использует матрицу, о которой было сказано в LSA и рассчитывается по формуле:

$$A = UWV^T,$$

где A – матрица размера NxM, U – ортогональная матрица с размером MxM, V - ортогональная матрица с размером NxN, W – матрица размера MxN, на главной диагонали которой лежат сингулярные числа, во всех остальных лежат нули.

Сингулярные числа – это неотрицательные вещественные числа, следующего вида:

$$A\alpha = \sigma\beta \text{ и } A\beta = \sigma\alpha,$$

где α и β – векторы единичной длины

Далее, чтобы предсказать рейтинг пользователя i объекту j, берем их скалярное произведение:

$$U_i * V_j = U_i^T V_j,$$

2.1.2 Рекомендательная система, основанная на методе содержания

Данный метод использует данные объекта и данные о просмотрах, скачиваниях и так далее [14]. Параметры, которые сопоставляются при фильтрации, зависят от типа. После фильтрации система рекомендует тот объект, который похож по

описанию на понравившийся объект. Далее, чтобы определить сходство объектов, применяется коэффициент Дайса

$$rd(A_i, C_j) = \frac{|parameter(A_i) \cap parameter(C_j)|}{|parameter(A_i)| + |parameter(C_j)|},$$

Алгоритм TF-IDF предназначен для выделения ключевых слов, а затем высчитывает влияние каждого из них, учитывая частоту использования их. TF (term frequency) вычисляет частоту в описании объекта и находится в диапазоне от 0 до 1, а IDF (inverse document frequency) показывает частоту объектов. Сам же метод является их произведением

$$TF(x, A) = \frac{fr(x,A)}{\max\limits_{y \in A} fr(y,A)},$$

$$IDF(x) = \frac{N}{n(x)},$$

где fr(x,A) – количество слов x в документе A; N- количество документов в наборе, n(x) – количество документов, в которых встречается слово x.

2.1.3 Рекомендательная система, основанная на знаниях

Данный метод схож с контентной фильтрацией, однако дополнительно рассматриваются взаимосвязи групп объектов. Фильтрация на основе знаний применяют в том случае, когда информации о поведении пользователя малы и редки. Для решения существует два метода основанных на знаниях:

— использование ограничений;

— выбор близких объектов.

Для первого метода используются объекты, которые полно соответствуют требованиям пользователя. Чтобы ограничить количество вопросов пользователю, применяются значения характеристик по умолчанию:

— Статические. Используются самые популярный объекты.

— Зависимые. Устанавливаются ограничения, которые характерны для объектов.

— Производные. Вычисляются по запросам пользователей.

Для метода близких объектов ищутся и рекомендуются те объекты, которые близки к требованиям пользователя. Для него используется мера схожести между требованиями и параметрами.

2.1.4 Гибридный метод

Гибридный метод состоит из двух других фильтраций: коллаборативная и контентная. Это самый сложный и эффективный метод из всех методов, так как он закрывает недостаток одного метода другим, используя множество алгоритмов.

2.2 Измерение точности рекомендаций

Для измерения точности чаще всего используется среднеквадратичная ошибки (RMSE), так как она поддается оптимизации при помощи градиентного спуска, и рассчитывается по формуле:

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{(u,i)\in T}(p_{ui} - r_{ui})^2},$$

Где u — пользователь, i — объект, r — оценка, p — предсказанная оценка, T — общее количество тестовых оценок.

Данные в таблице 5 приведены качества рекомендательных систем, основанных на различных методах [15].

Таблица 5. Оценка точности методов рекомендательных систем.

Метод	Оценка RMSE
Метод основанный на содержании	1,48
Коллаборативная фильтрация	1,35
Метод основанный на знаниях	1,41
Гибридный метод	1,40

Чем меньше результат, тем точнее метод составляет рекомендации. Как видно из таблицы, наилучший результат у коллаборативной фильтрации – 1,35.

2.3 Недостатки и достоинства

Недостатки и достоинства методов рекомендательной системы продемонстрированы в виде схем, изображенных на рисунках 12 и 13 соответственно.

Рисунок 12. Недостатки методов рекомендательной системы.

Недостатками коллаборативной фильтрации является зависимость от количества оценок. Если по каким-то объектам отсутствуют оценки, то система должна вычислить остальные элементы матрицы. А также для нового пользователя, у которого еще нет оцененных объектов или их слишком мало для анализа, сложно что – либо рекомендовать. Пассивное поведение пользователей может привести к простою системы.

Контентная фильтрация рассматривает только однотипные объекты, что может не работать в том случае, если пользователь искал объект, например автомобиль, и уже купил его, а система дальше рекомендует автомобили. Ярким примером является реклама в интернете.

Рекомендательные системы, основанные на знаниях тяжело реализовать, так как необходимо соединить различные группы объектов, чтобы система не повторяла ошибок контентной фильтрации.

Гибридная рекомендательная система сложна тем, что в них вложены различные алгоритмы из разных методов.

Рисунок 13. Достоинства методов рекомендательной системы.

2.4 Выводы

Данные методы проектирование систем поддержки принятия решений используются при определенных условиях. Исходя из потребностей, стоит выбирать самый подходящий метод проектирования. Исходя из поставленной задачи, необходимо будет применить гибридный метод, с элементами всех существующих методов. Безусловно, данный подход будет сложен в реализации, но выбор связан с тем, что вина не имеют сходства между собой даже при сходстве множества параметров. Как пример, если два вина произведены во Франции, из одного сорта винограда, одной винодельней и даже имеют одно название, но произведены с разницей в год, то они могут иметь абсолютно разный вкус из-за такого фактора, как год. В тоже время, требуется постоянное выставление пользователем оценок и пояснений к ним, чтобы система постоянно подстраивалась под персональные вкусовые определения пользователя, так как для разных пользователей одно и тоже вино может быть слишком вяжущим, а для других слишком сладким (в случае с полусухим вином). Тем самым, это позволит разработать систему лишь с недостатком того, что пользователь должен всегда взаимодействовать с системой, однако это нужно лишь для улучшения рекомендаций.

3 Аналитический обзор методов распознавания текста

Оптическое распознавание символов – это процесс, реализующий перевод изображения печатного, машинописного или рукописного текста в текстовые данные, представленные в электронном виде [16]. Качество распознавания от того, как данные подаются системе. Основные факторы, влияющие на качество работы определяются следующим образом:

— шрифт;

— размер символов;

— освещение.

На сегодняшний день существует несколько систем, которые позволяют решить данную задачу, однако все они имеют свои минусы и недостатки в первую очередь, связанную с методом реализации данных систем. Однако, единым для всех остается процесс распознавания текста.

3.1 Восприятие

Человек, читая текст, способен узнавать буквы и на основе полученный букв строить слова. Более того, зачастую человек способен распознавать слова в тексте, который не разделен пробелами. Этого он добивается, обучаясь в школе и читая, увеличивая свой словарный запас. Однако, иногда человек, ввиду проблем со зрением, не всегда может прочесть какую-то букву, но в конце концов способен прочитать слово использую видимые ему буквы. Зачастую, мы этого не замечаем, так как мозг достаточно быстро способен обрабатывать данную информация, но если человеку с проблемным зрением дать текст на другом языке, то он начнет выдвигать гипотезы о возможных пропущенных буквах (например, при попытке перевода текста в переводчике). Так же себя ведет техническая система, которая ищет для каждой буквы соответствующий класс буквы. Правда, для качественной работы, необходимо выполнение свойства отображаемости, то есть каждый класс буквы должен быть описан так, чтобы в него попадал объект данного класса и не попадал объект другого класса. Это обеспечивает эталон системы (изображение эталонных букв).

3.2 Предобработка.

На данном этапе необходимо минимизировать влияние сторонних факторов, таких как фон и цвет. В большей части проектов используется приведение фона к белому цвету, а слов к черному, более того, возможна размытость буквы на более серый цвет, что только улучшает качество работы системы.

3.3 Сегментация

На данном этапе текст делится на строки, при помощи междустрочных белых линий, а после, строки разбиваются по буквам. Затем, буквы нормализуются до размеров эталонов.

3.4 Распознавание

3.4.1 Распознавание при помощи метрик.

Метрика – некоторое условное значение функции, определяющее положение объекта в пространстве. Наиболее популярной метрикой при распознавании текста является метрика Хэмминга, которая показывает, насколько объекты не похожи друг на друга.

Принцип распознавания при помощи метрик имеет плюсы и минусы:

Плюсы:

- легкая реализация;
- высокая точность при одинаковых параметрах шрифта.

Минусы:

- шибка при распознавании схожих букв (“i”,”j”);
- низкое качество при различных параметрах шрифта.

3.4.2 Распознавание при помощи нейронной сети.

Для данной задачи используется большое количество нейронных сетей. На вход им подается изображение NxM (размер входа нейронной сети). Для каждого входа задан определенный коэффициент, полученный в ходе обучения. Чей заряд по итогу распознавания будет больше, тот нейрон и испустит импульс. Что касается качества

работы такого метода, то он напрямую зависит от размера обучающей выборки и ее качества.

3.5 Проблемы

Рассмотрим подробнее каждую проблему.

3.5.1 Проблема наличия неоднородности освещения

Неоднородность освещения влияет на контраст каждой буквы и для ее решения необходимо выполнить операцию бинаризации. На сегодняшний день можно выделить несколько основных методов:

— среднего порогового значения;
— Бернсена [17];
— Ниблэка [18];
— Саувола [19];
— Вульфа [20];
— Брэдли-Рота [21].

В работе 22 проведен анализ данных методов и были получены следующие результаты:

Метод Брэдли-Рота содержит высокие показатели оценки точности бинаризации (рис. 14) и наименьшее время выполнения (рис. 15), но подвержен влиянию со стороны глобальной контрастности изображения (рис. 16). Методы среднего порогового значения и Вульфа содержат низкие значения коэффициента корректно преобразованных пикселей объекта k_p (рис. 14), но почти не подвержены изменениям контрастности (рис. 16) Методы Бенсена и Ниблэка показывают большой процент шумов печати, что приводит к снижению коэффициента точности преобразования A_p.

Коэффициент корректно преобразованных пикселей объекта:

$$k_p = \frac{c_p}{p},$$

где c_p – число корректно преобразованных пикселей объекта, p- число пикселей объекта;

Точность преобразования:

$$A_p = 1 - \frac{(i_p + d_p)}{pi},$$

где i_p,d_p – число ошибочно вставленных и удаленных пикселей объекта, pi-число пикселей изображения.

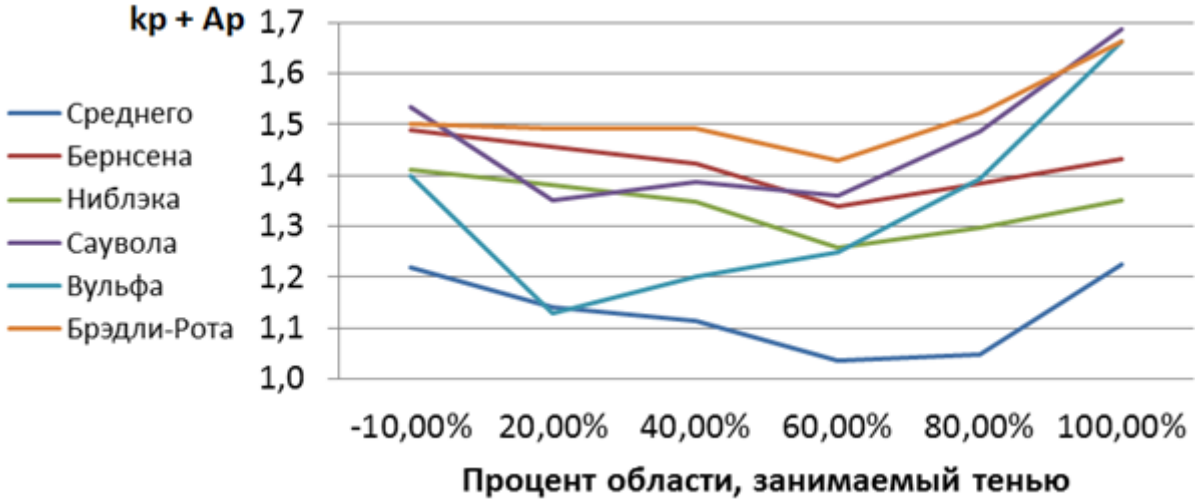

Рисунок 14. Зависимость среднего значения точности бинаризации (*kp*+*Ap*).

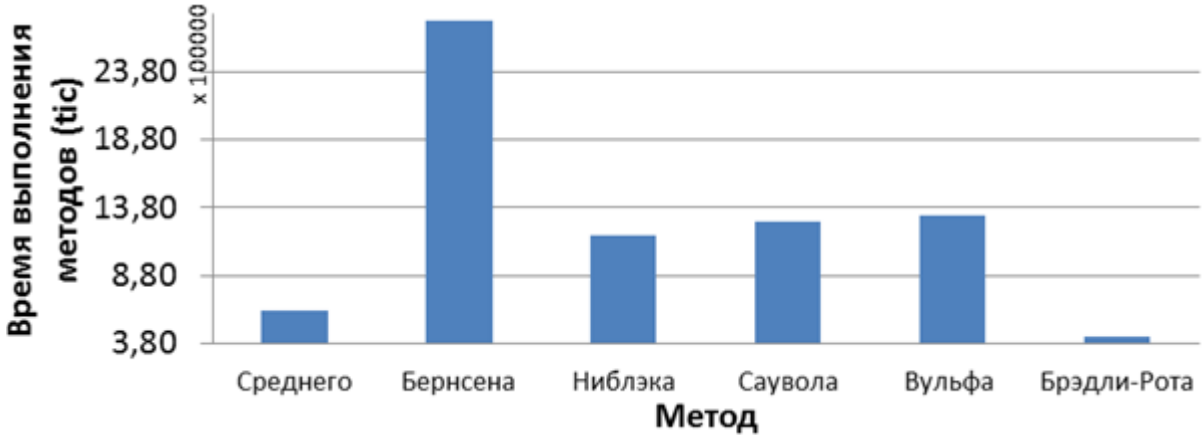

Рисунок 15. Время выполнения методов.

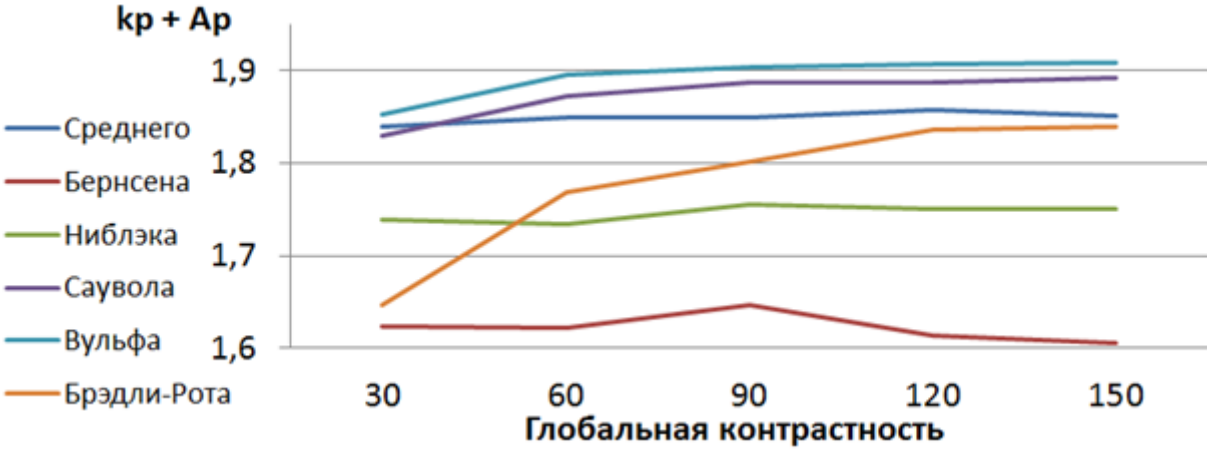

Рисунок 16. Зависимость точности бинаризации (*kp+Ap*) от глобальной контрастности изображения.

Исходя из данного анализа нам подойдет метод Бредли-Рота.

3.5.2 Проблема наличия различных размеров, форм, наклонов символов

Данная проблема решается на этапе распознавания, методом, которые были рассмотрены ранее. Исходя из предыдущего описание методов в нашей задаче необходимо использовать нейронную сеть. Рассмотри наиболее популярные из них (таблица 6).

Таблица 6. Сравнение свойств методов распознавания.

Метод	Входное значение – изображение	Инвариантность к			
		искажениям	углу	положению	размеру
Дерево решений	–	–	–	–	–
Генетические алгоритмы	–	–	–	–	–
НС Хопфилда	+	+	–	–	–
НС высокого порядка	+	–	+	–	+
Сверточная НС (рис.17)	+	+	+	+	–

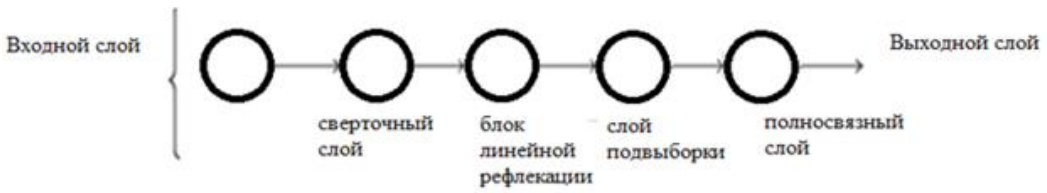

Рисунок 17. Архитектура сверточной нейронной сети.

На основе полученных данных следует выбирать сверточную нейронную сеть, так как размер текста названия на винных этикетках пишется единым размером.

3.6 Выводы

На основании анализа предлагается использовать метод Брэдли-Рота, так как он имеет высокие показатели оценки точности бинаризации и наименьшее время выполнения, а глобальная контрастность изображения будет иметь малый эффект на этикетку вина.

Так же предлагается использовать сверточную нейронную сеть, так как винные этикетки содержат однородный размер в названии для одной бутылки, но могут быть написаны различными шрифтами и под разными углами на различных бутылках. Более того, человек может под разным углом сканировать бутылку.

В итоге получается следующий алгоритм:

- сегментация:
  - оценка контрастности изображения;
  - увеличение контрастности при необходимости;
  - применение метода Брэдли-Рота;
  - сегментация символов;
- распознавание:
  - сверточная нейронная сеть.

4 Обоснование выбора языков программирования и инструментальных средств

4.1 Выбор операционной системы

Данная система реализуется под устройства на ОС Android и связано это с тем, что согласно статистике StatCounter [22] данная операционная система занимает лидирующие позиции на рынке носимых устройств (рис. 23).

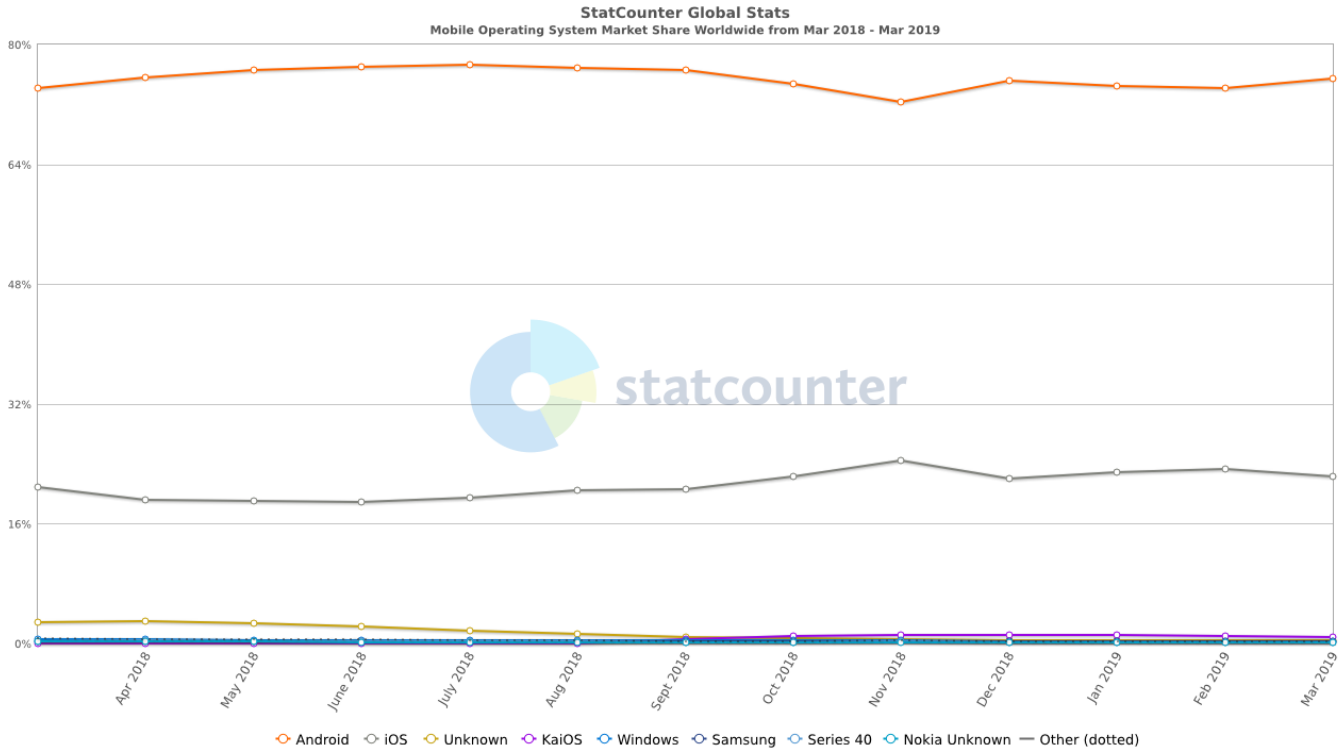

Рисунок 18. Статистика рынка ОС носимых устройств.

1) Android - 75.39%

2) iOS - 22.35%

3) KaiOS [23] - 0.84%

4) Unknown (неизвестные) - 0.36%

5) Windows [24] - 0.28%

6) Samsung [25]- 0.26%

Более того, данная операционная система имеет большое комьюнити и поддержку со стороны лидера ИТ-рынка Google.

4.2 Выбор среды разработки ПО

В качестве интегрированной среды разработки была выбрана Android studio [26], которая является основной средой разработки ПО под Android. В свою очередь,

данная среда поддерживает два языка программирования: Java [27] и Kotlin [28]. Выбор пал на первый из них, так как Java зарекомендовал себя среди разработчиков за много лет.

4.3 Выбор базы данных

После проведения анализа методов распознавания текстов было принято решение использовать базу данных Firebase со встроенным функционалом машинного обучения, в частности распознавания текста. Сама база является No-SQL, соответственно, она является нереляционной и к ней не применимы законы нормальной формы.

4.4 Выбор среды прототипирования

Для сбора информации о винах разработана программа на ЯП Python [29], который в дальнейшем использовался для моделирования рекомендательной системы.

При разработке рекомендательной системы использовался дистрибутив Anaconda3 [30] и Python 3.6. Причиной выбора данного стека является большое количество поддерживаемых библиотек науки о данных и машинного обучения.

4.5 Выводы

Для реализации данной системы были выбраны следующие технологии:

— ОС Android;

— среда разработки Android studio;

— ЯП Java и Python 3.6;

— БД No-SQL Firebase;

— библиотеки Firebase ML-kit;

— дистрибутив Anaconda3.

5 Теоретическая часть

5.1 Сбор данных

На сегодняшний день вино является самым популярным алкогольным напитком, однако не существует каких-либо баз или каталогов вин, где можно найти информацию о винах. Для решения данной проблемы было принято решение использовать онлайн-магазины, которые специализируются на винах. Данные на сайтах должны соответствовать нескольким требованиям:

— Наличие региона производства вина.

— Наличие года вина.

— Наличие описания вина.

— Наличие перечня сортов для каждого вина с процентным соотношением.

К сожалению, этих данных недостаточно для максимально качественной рекомендации, однако их хватает чтобы рекомендовать вина с высокой точностью.

Для реализации используется парсер, который проходя HTML (HyperText Markup Language, [31]) разметку сайта находит нужные нам данные. Необходимо, однако собирать данные так, чтобы они не повторялись. Для этого данные записываются в словарь, а затем в формат JSON (JavaScript Object Notation, [32]). В качестве ключа используется название вина и год производства, что гарантирует оригинальность всех вин.

5.2 Обработка полученных данных

После получения всей информации необходимо убрать символы, которые не поддерживаются базой данных Firebase [33]. Для этого все такие символы заменяются пустым местом, что позволяет не потерять данные и продолжить их разбиение на слова. Так же необходимо убрать выбросы, то есть данные, которые не соответствуют большому количеству данных. Примером таких данных являются безалкогольные вина, которые будут негативно влиять на рекомендательную систему в случае, если найдется такое же по характеристикам вино, но с содержанием алкоголя.

Еще одним этапом обработки информации является заполнение пропусков данных. Проблема этапа заключается в том, что необходимо визуально проанализировать данные и понять, что пропущено и как это исправить. Так,

например, в ходе обработки сортов винограда, было замечен, что некоторые вина не имеют четко прописанного процентного содержания. Однако, при выводе всех таких вин было получено, что процентное содержание не прописано в винах, состоящих из сортов винограда в равных пропорциях.

Так же, довольно часто пропускается год выпуска вина. Это связано с тем, что многие недорогие вина не указывают его, так как из года в год виноград данных сортов не зависит от погодных условий ввиду того, что он выращивается в неестественных условиях или же данный сорт винограда достаточно непривередлив к погодным условиям.

Еще один параметр, который зачастую является пропущенным — это регион производства. Для заполнения пропусков в этом пункте приходится вручную рассматривать возможные значения. Так, использовалось несколько источников.

1) Книга Роберта Паркера.

Из данной книги были взяты все возможные места производства вин, и если в стране производится вино из текущего сорта винограда, и только в нем, то регион проставлялся для всех вин с данным сортом и страной одинаковым.

2) Исходя из закона.

Гораздо меньше вин получается заполнить данным способом, и применяется исключительно для европейских и американских вин, а также австралийских. Это обусловлено тем, что некоторые названия вин содержат регион производства (например шампанское), который нельзя использовать при производстве вин вне данного региона. В свою очередь в РФ данное правило не исполняется, так, например, российские производители выпускают игристое вино с названием Российское шампанское. Однако такую проблему можно решить при помощи страны производства.

5.3 Взаимодействие с базой данных

В качестве БД используется Firebase, это встроенная база данных в Android Studio, которая позволяет в реальном времени пополнять базу, и все пользователи могут получать актуальные данные. База является No-SQL (not only structured query language, [34]), что не предоставляет возможность использовать нормализацию. При

каждом взаимодействии с БД пользователь скачивает необходимые ему данные. Это позволяет экономить память устройства, так как база скачивается лишь временно и хранится в оперативной памяти, и экономить трафик, так как скачивается определенное количество информации, и в случае необходимости, дополняется новой информацией с базой.

База не содержит в себе таблиц, как в привычных реляционных база данных, а содержит каталоги. Всего в базе три каталога:

1. Каталог с винами.
2. Каталог пользователей.
3. Каталог вин пользователя.

В случае открытия пользователем списка вин, первым делом скачиваются данные вин с оценками, а затем дополняются остальными винами.

В базе пользователей содержатся логин и пароль каждого пользователя. Пример базы содержится в приложении 1.

5.4 Распознавание текста этикетки

Исходя из анализа методов распознавания текстов, необходима сверточная нейронная сеть с методами сегментации. Процесс обучения проходил следующим образом:

- Берется изображение буквы из обучающей выборки;
- Анализируются позиции черных пикселей;
- Выравниваются коэффициенты;
- Минимизируется ошибка совпадения методом градиента;
- Каждому нейрону сопоставляется изображение.

По окончанию обучения было получено подобия холста, где чем темнее пиксели, тем чаще встречаются данные пиксели в словах (рис. 22).

Рисунок 19. Итог обучения буквы "А".

Однако данный метод не позволяет считывать буквы, которые написаны курсивом так далее. Многие производители используют дизайнерские стили, которые тяжело распознавать. Для этого необходимо увеличение разнообразия и количества выборки. Однако многие шрифты не предоставляется возможным учесть и соответственно колоссально увеличивается время обучения. Также данный метод не позволяет отличать русский и английский текст.

Для определения текста использовались буквы в слове, и если они входили в русскую группу, то остальные буквы слова представлялись русскими. Однако некоторые русские слова могут одержать исключительно те буквы, которые встречаются как в кириллице, так и в латинице (например, вино Аристео). Еще один способ исходя из анализа, использовать готовые слова. Их можно брать из базы вин.

Однако, в процессе разработки было получено, что российские вина, а также вина стран СНГ, не стоит использовать, так как исходя из ранних описаний, множество проблем возникает с регионом, который на данных винах часто не написан, или же в целях маркетинга, не является правильным (шампанское Российское). Так же некоторые вина не имеют четко определенного названия. Как пример, «Вино Грузинское Домашнее». Как итог, в связи с их небольшим количеством, было принято решения не использовать данные вина.

Так как вина на русском языке были выкинуты, уменьшилась выборка данных и уменьшилось время обучения нейронной сети, однако оно все равно остается достаточно высоким. В связи с этим было принято решение провести тестирование нейронной сети от Google, входящую в состав Firebase ML-kit. Исходя из описания

данной библиотеки [33] она достаточно обучена на большом количестве данных и может распознать практически любой печатный текст. По результатам проверки на сложных этикетках и даже рукописном тексте, было получено распознавание всех текстов, правда в редких случаях не с первого раза. Эта проблема решена тем, что при использовании данной технологии, на короткое время сохраняется правильно распознанное слово.

5.5 Рекомендации

Исходя из количества информации, которая доступна о винах, создавать рекомендации можно на основе сота винограда и температуры региона сорта. В случае с сортом необходимо создать классы сортов, которые максимально похожи друг с другом. Для этого необходимо на основании описания сортов винограда классифицировать их. Так как изначально никаких классов нет, то необходимо использовать метод машинного обучения – кластеризация.

1. Сходство Джаро-Винклера [35]

Данный метод позволяет определить схожесть строк. В нашем случае каждая строка – это слово-признак. Тем самым, определяются все категориальные признаки к единой форме.

$$d_w = d_j + \left(lp\left(1 - d_j\right)\right),$$

где:

- d_j — расстояние Джаро для строк
- l — длина общего префикса от начала строки до максимума 4-х символов
- p— постоянный коэффициент масштабирования, использующийся для того, чтобы скорректировать оценку в сторону повышения для выявления наличия общих префиксов. не должен превышать 0,25, поскольку в противном случае расстояние может стать больше, чем 1. Стандартное значение этой константы в работе Винклера: $p = 0.1$.

Расстояние Джаро:

$$d_j = \begin{cases} 0, & m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right), & \text{в остальных случаях'} \end{cases}$$

где:

— $|s_1|$ — длина строки;

— m — число *совпадающих символов*;

— t — половина числа *транспозиций*.

После приведения к единой форме необходимо создать матрицу признаков сорта винограда. В каждой строке будет содержаться сорт, а в каждом столбце вкусовой параметр. $A_{ij} = 1$, если сорт ”i” содержит параметр “j”. Так же, приводим много сортовые вина. Для этого для каждого «букета» сортов складываются все параметры умноженные на процент содержания сорта в вине, и:

$$A_{ij} = \begin{cases} 0, & A_{ij} >= 0.5 \\ 1, & \text{в остальных случаях} \end{cases},$$

Пример:

Вино D содержит A*30% + B*30% + C*40%

Таблица 7. Пример определения букета сортов.

	Яблоко	Апельсин	Корица	Вишня	Лимон	Малина
A	1	0	0	1	1	0
B	1	1	0	1	0	0
C	0	0	0	1	1	1
D	1 * 0,3 + 1 * 0,3 + 0 * 0,4 = 1	0 * 0,3 + 1 * 0,3 + 0 * 0,4 = 0	0 * 0,3 + 0 * 0,3 + 0 * 0,4 = 0	1 * 0,3 + 1 * 0,3 + 1 * 0,4 = 1	1 * 0,3 + 0 * 0,3 + 1 * 0,4 = 1	0 * 0,3 + 0 * 0,3 + 1 * 0,4 = 0

Затем рассчитывается Евклидово расстояние для всех вин и создаются группы вин:

$$p(x, x') = \sqrt{\sum_i^n (x_i - x_i')^2},$$

где x – критерий первого объекта, x’ – критерий первого объекта.

Следующий матрицей задается средняя температура регионов по годам. Этот параметр влияет на некоторые «капризные» сорта винограда. Исходя из [14], температура должна отличаться не более чем на градус.

В итоге рекомендация состоит из того, что подбираются вина, сорта винограда которых находятся в группе с винами, у которых большая средняя оценка вин, а также температура в году сбора винограда, отличается не более чем на градус от

температуры рекомендуемого вина. В случае, если год отсутствует, то поиск по температуре пропускается.

6 Описание системы

В итоге получается следующая UMl (Unified Modeling Language, [36]) диаграмма использования:

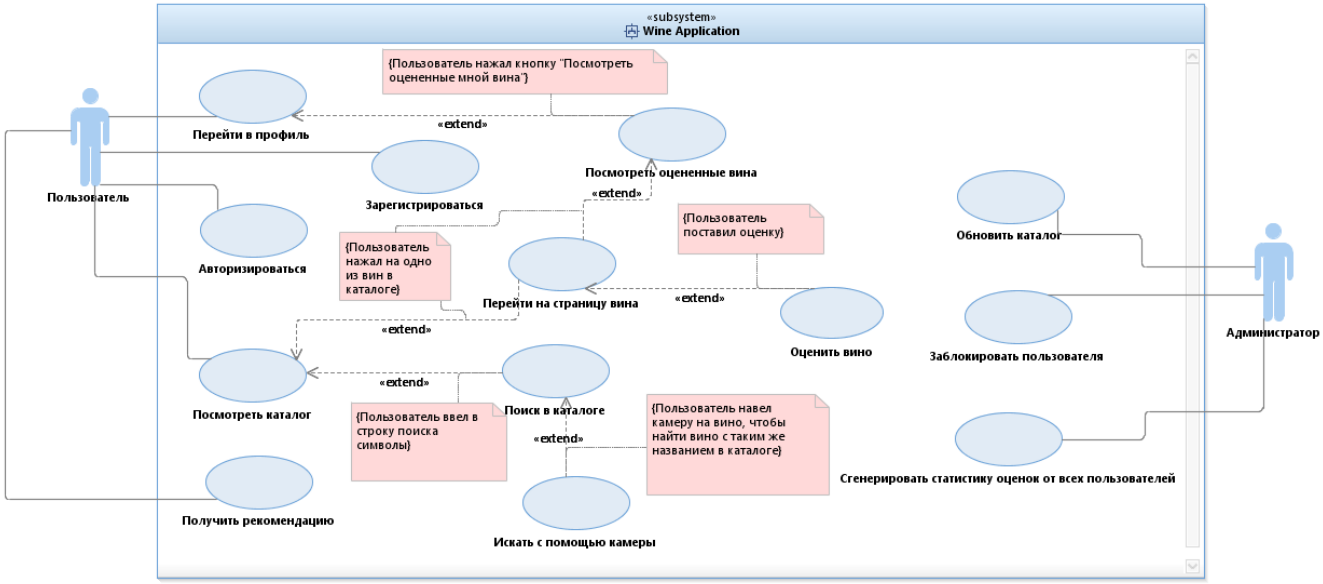

Диаграмма 1. Варианты использования системы.

Деятельность каталога представлена на следующей диаграмме:

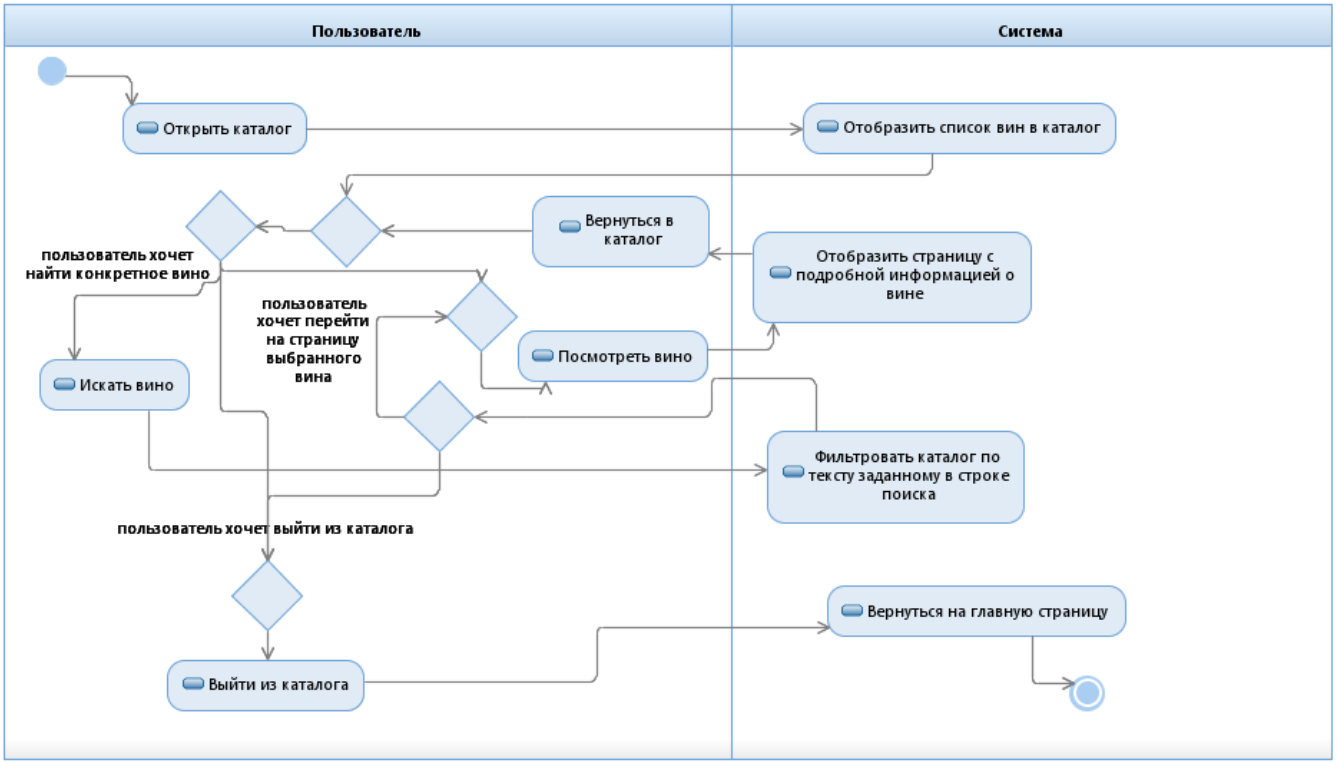

Диаграмма 2. Деятельность каталога.

Алгоритм рекомендательной подсистемы выглядит следующим образом:

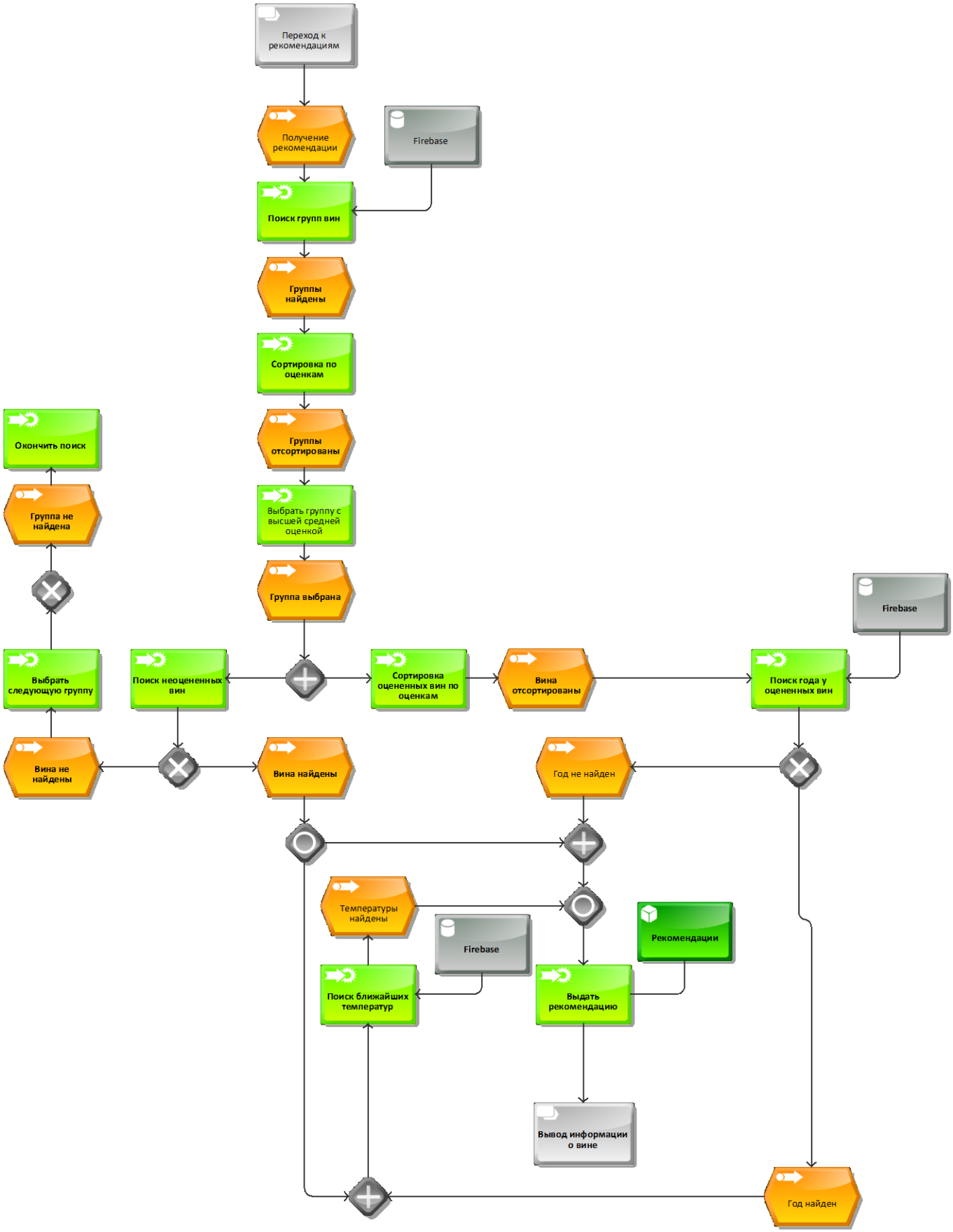

Диаграмма 3. Алгоритм рекомендательной подсистемы.

В итоге получилась следующая функциональная схема:

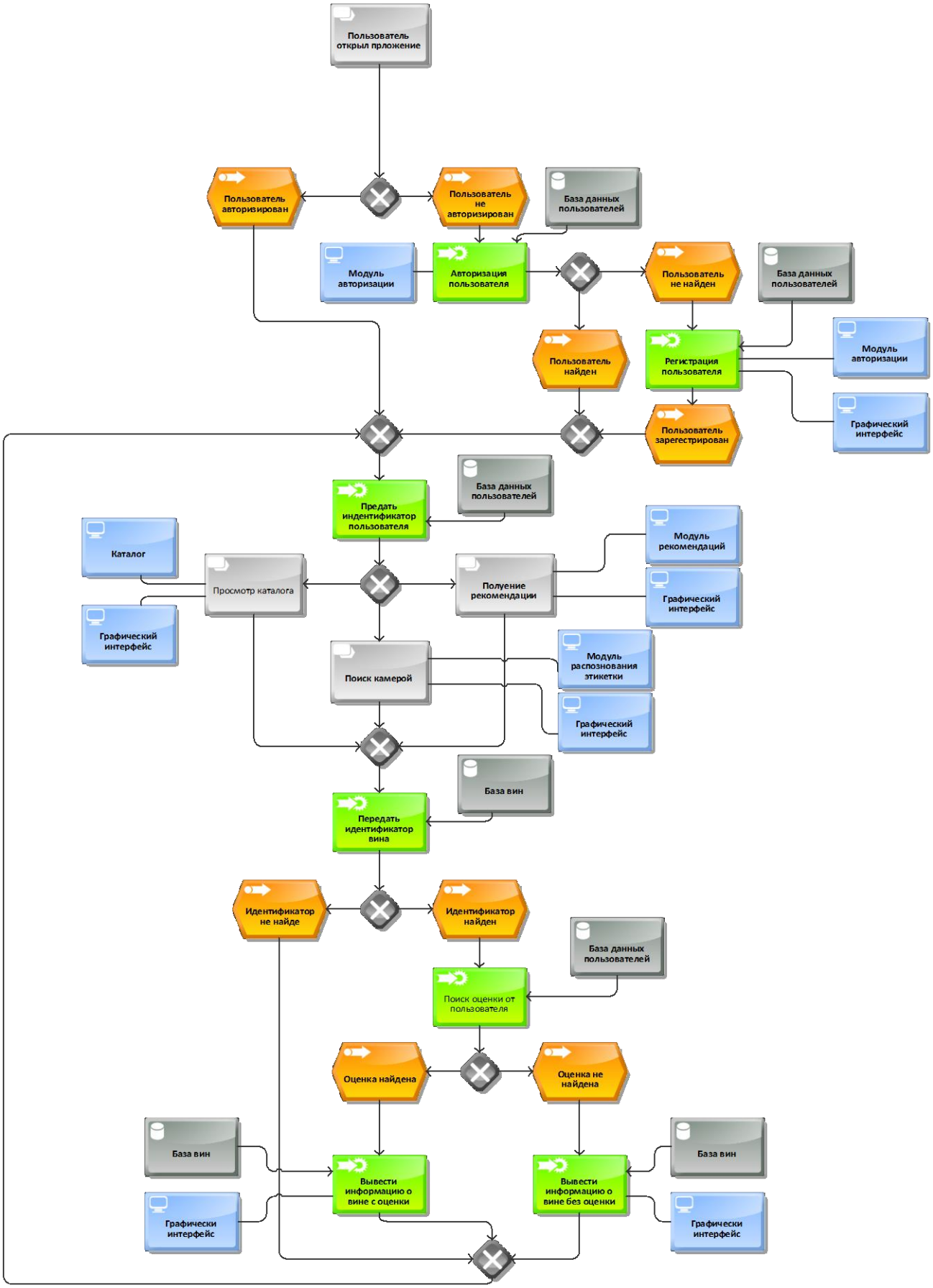

Диаграмма 4. Функциональная схема.

7 Практическая реализация

7.1 Сбор информации.

Сбор информации происходит с интернет-магазинов, так, как только в них содержится максимум информации о винах. Для этого использовалось два магазина:

— АльтаВина;

— WineStyle.

Первым шагом является скачивание JSON файла с уже имеющимися винами из базы данных. Затем, эти данные заносятся в словарь, где название вина и год являются ключами. После этого считывается HTML разметка сайта и из нее вычисляется количество страниц. Следом на каждой странице считывается ссылка на каждое вино в отдельности. Переход по страницам происходит за счет изменения URL (Uniform Resource Locator, [37]) ссылки. Однако необходимо отметить, что загрузка страницы должна происходить дважды. Это связано с тем, что данные сайты имеют ограничения по возрасту, и в случае обращения к сайту на любой странице, будет открыта первая страница с просьбой подтвердить совершеннолетие. Все это выполняется в специальной сессии, чтобы сайт подумал, что это обращается человек. На странице с вином берется разметка и заполняются данные о винах. По итогу получается следующая структура (рис. 20 - 21)

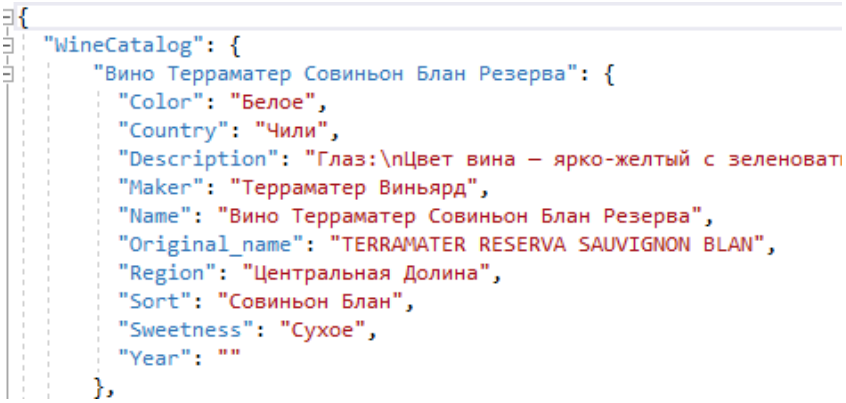

Рисунок 20. Структура каталога вин.

	Color	Country	Description	Maker	Name	Region	Sort	Sweetness	Year
Вино Терраматер Совиньон Блан Резерва	Белое	Чили	Глаз:\nЦвет вина — ярко-желтый с зеленоватым о...	Терраматер Виньярд	Вино Терраматер Совиньон Блан Резерва	Центральная Долина	Совиньон Блан	Сухое	
Вино Аристео	Красное	Италия	Глаз:\nРубиново-красный цвет \n\n\nНос:\nАрома...	Ла Боллина	Вино Аристео	Тоскана	Санджовезе 25 % + Канайоло 25 % + Каберне Сови...	Полусухое	
Вино Марани Киндзмараули	Красное	Грузия	Глаз:\nЦвет вина насыщенный красный с лиловыми...		Вино Марани Киндзмараули	Кахетия	Саперави	Полусладкое	
Вино Анжу Вилляж Экспрессьон	Красное	Франция	Глаз:\nВино насыщенного рубинового цвета \n\n\...	Домен Де Троттьер	Вино Анжу Вилляж Экспрессьон	Долина Луары	Каберне Фран 60 % + Каберне Совиньон 40 %	Сухое	2015
Вино Рислинг Вьей Винь Домен Жан-Марк Бернар	Белое	Франция	Глаз:\nСоломенно-золотистый цвет \n\n\nНос:\nЛ...	Домен Жан-Марк Бернар	Вино Рислинг Вьей Винь Домен Жан-Марк Бернар	Эльзас	Рислинг 100 %	Полусухое	2016

Рисунок 21. Пример выводимых данных.

В свою очередь, использование ключа, описанного выше, позволяет предотвратить повторную запись вин.

7.2 Рекомендательная подсистема

После получения всей информации о винах необходимо узнать, сколько уникальных сортов содержится в базе. Уникальными считаются как отдельные сорта, так и букеты сортов винограда. В итоге их оказалось 1130 (рис. 22).

```
import numpy as np
import pandas as pd

ratings_df = pd.read_json('./Source/repos/RecomendationSystem/PythonApplication2/wine.json',encoding = "utf-8").T

print(len(ratings_df['Sort'].unique()))
```

```
1130
```

Рисунок 22. Количество сортов.

После чего создается словарь со всеми сортами винограда и их описанием (рис. 23).

```
Шардоне
['ЗЕЛЕНОЕ ЯБЛОКО', 'ГРУША', 'ЦИТРУСОВЫЕ', 'ДЫНЯ', 'АНАНАС', 'ПЕРСИК', 'СУХОФРУК
ТЫ', 'МАСЛО', 'ВОСК', 'СМЕСЬ ПРЯНОСТЕЙ', 'СЫРАЯ ШЕРСТЬ', 'МИНЕРАЛЫ', 'ТРОПИЧЕСК
ИЕ ФРУКТЫ']
Совиньон Блан
['СВЕЖЕСКОШЕННАЯ ТРАВА', 'ПЕРСИК', 'КРЫЖОВНИК', 'ПОЧКИ СМОРОДИНЫ', 'ЗЕЛЕНЫЕ БОБ
Ы', 'ИНОГДА МИНЕРАЛЫ']
Семийон
['ТРАВА', 'ЦИТРУСОВЫЕ', 'ЛАНОЛИН', 'МЕД', 'ЖАРЕНЫЙ ХЛЕБ', 'ВОСК']
Мюскадель
['ЯБЛОКИ', 'ТРАВА']
Рислинг
['ХРУСТЯЩИЕ ЗЕЛЕНЫЕ ЯБЛОКИ', 'СУШЕНЫЕ ЯБЛОКИ', 'АЙВА', 'АПЕЛЬСИН', 'ЛИМОН', 'МЕ
Д', 'МИНЕРАЛЫ', 'НЕФТЬ', 'ЖАРЕНЫЙ ХЛЕБ']
Гевюрцтраминер
['ПРЯНОСТИ', 'РОЗА', 'ЛИЧИ', 'МУСКАТНЫЙ ОРЕХ', 'МЕД']
Мускат
['ВИНОГРАД', 'АПЕЛЬСИНЫ', 'РОЗА', 'БЕРГАМОТ', 'ИЗЮМ', 'ЯЧМЕННЫЙ САХАР', 'ПОМЕЛ
О', 'ЛИЧИ']
Пино Гри
['ПЕРСИК', 'АПЕЛЬСИН']
Шенен блан
['МЕД', 'АКАЦИЯ', 'АЙВА', 'ЗАСАХАРЕННЫЕ ФРУКТЫ', 'ЛАКРИЦА']
Вионье
['ПЕРСИК', 'АБРИКОС', 'МЕД', 'СПЕЦИИ']
Марсан
['БОЯРЫШНИК', 'ФИАЛКА', 'АКАЦИЯ', 'МЕД']
Руссан
['МЕД', 'АБРИКОС', 'БОЯРЫШНИК']
Шассла
['ЗЕЛЕНАЯ ТРАВА', 'ЛИПА', 'ЦВЕТЫ']
```

Рисунок 23. Пример описаний сортов.

После чего, описанными ранее методами происходит разбиение на группы.

Реализация таблицы с температурами происходила в ручном режиме. Это связано с тем, что многие регионы не являются административными единицами в области географии, а являются административными единицами в области виноделия. Сбор происходил из [38].

7.3 Модуль авторизации.

Для полноценного использования всех функций приложения пользователю необходимо авторизоваться или зарегистрироваться. Была реализована упрощенная система регистрации пользователей без каких-либо дополнительных данных, а только с адресом электронной почты и паролем. Для пользователя данный экран будет выглядеть следующим образом (рис. 24):

Рисунок 24. Окно авторизации.

При регистрации нового пользователя данные заносятся в Firebase Authentication, где каждому пользователю присваивается уникальный идентификатор, которым впоследствии мы будем пользоваться при проставлении оценок винам и составлении рекомендаций. Firebase Authentication – это специальная упрощенная система для хранения данных пользователей. Она также позволяет легко получать доступ из других модулей Firebase, таких как database. Как можно увидеть на рисунке (рис. 25), администраторам доступна такая информация, как адрес

электронной почты, даты регистрации и последнего входа, идентификатор, но не пароль, в целях защиты индивидуальных данных пользователя.

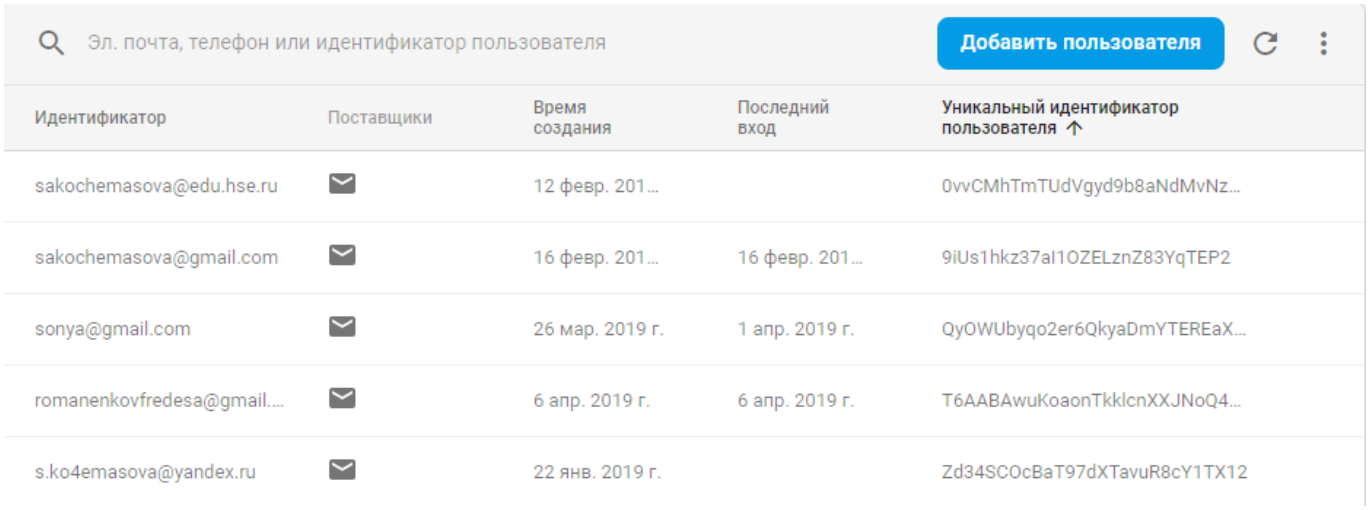

Рисунок 25. Часть базы данных с пользователями.

Программная реализации авторизации и регистрации также становится простой и понятной с Firebase Authentication. Однако ограничения на входные данные были прописаны в коде нами, чтобы поставить свои условия регистрации. Мы ограничили длину пароля (не менее 6 символов), а также добавили ограничение на формат электронной почты (входные данные должны соответствовать маске ‘*@*.*’). Когда пользователь уже зарегистрирован, то он может авторизоваться и получить доступ к личному кабинету, оцениванию вин и получению индивидуальных рекомендаций. Без авторизации пользователь может только посмотреть каталог вин или воспользоваться камерой для распознавания вина.

7.4 Модуль распознавания текста.

Как говорилось ранее, решено использовать встроенный в Firebase модуль распознавания текста. Однако он позволяет лишь считать данные с изображения. Для большего качества функция распознавания этикетки использует камеру мобильного телефона в режиме видеосъемки. Первым делом были установлены необходимые APK и интегрированы в проект. После чего создается объект TextRecognizer, который и позволяет распознать текст. Однако, он может не работать из-за нехватки памяти или не готовности приложения. Для того, чтобы избежать ошибок работы, добавляется проверка того, что объект готов к работе.

Далее создается специальный класс, реализующий управление камерой. Для того, чтобы распознавать текст с этикетки, надо установить максимально возможное разрешение. Исходя из текущих моделей телефонов, было принято решение установить разрешение на отметке HD (1280*720). Частота кадра устанавливается на отметке 30fps. Так же необходимо настроить автофокус.

После чего реализуется процесс обработки изображения. Из TextRecognizer получаются блоки с текстом и передаются в модуль вывода текста на экран и в модуль поиска в базе. В первом случае проверяются все текстовые блоки, которые распознаны камерой, и в случае нахождения названия вина, выводит дополнительную информацию о нем на экран, не прерывая съемку. Для того, чтобы при тряске камеры или при плохом освещении не происходил сброс вывода, вывод задерживается на 2 секунды при не обнаружении текста или пока не появится новое название вина.

Во втором случае данные передаются в модуль, который ожидает нажатия на экран пользователем. Как только тот нажмет на экран, будет произведено название вина и приложение перейдет в каталог, где будет это вино.

При наведении камеры на этикетку вина, пользователь получает информацию о сорте, регионы производства и виде (рис. 26).

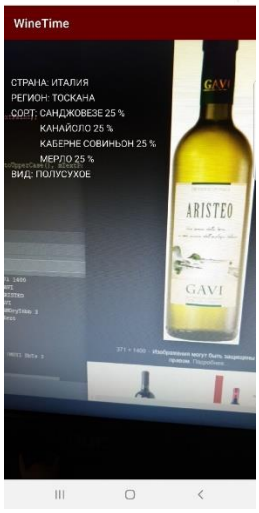

Рисунок 26. Пример работы поиска камерой.

При нажатии на экран, пользователю выводится информация о вине (рис. 27).

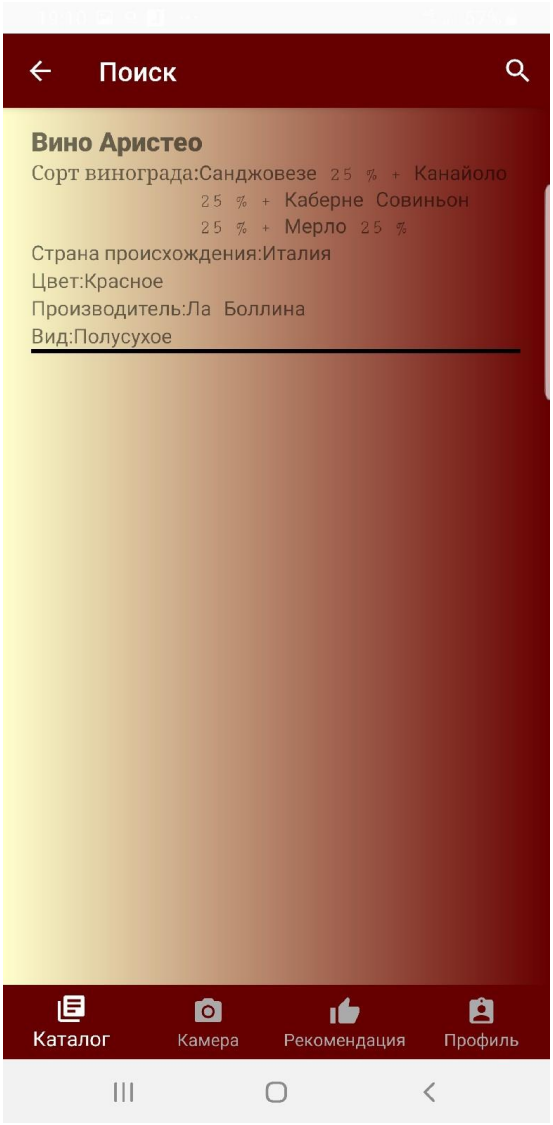

Рисунок 27. Результат нажатия при наведенной камере.

7.5 Графический интерфейс.

Графический интерфейс содержит четыре вкладки:

— Каталог;

— Камера;

— Рекомендации;

— Личный кабинет.

Окно каталога будет содержать информацию, которая подгружается с базы данных, и возможность живого поиска. Для поиска реализована постоянная обработка действий в ленте ввода информации. При добавлении символа, данные, которые содержатся в кэше, который содержит результаты последней обработки

ввода символа, изменяются тем, что удаляются вина, которые перестают подходить под поиск. В случае удаления символа поиск происходит для всего каталога в целом.

Окно камеры позволит подключать модуль распознавания текста и при нажатии переходить в каталог.

Окно рекомендательной системы содержит в себе несколько кнопок для вывода информации из рекомендательной подсистемы. (рис. 28)

Рисунок 28. Окно рекомендательной подсистемы.

Личный кабинет содержит в себе кнопку выхода и просмотр оцененных вин (рис. 29).

Рисунок 29. Личный кабинет.

7.6 Сборка программного обеспечения

Интеграция всех модулей происходила в следующем порядке. В первую очередь было интегрировано взаимодействие с базой данных и выводом всех вин в каталог (рис. 30).

Рисунок 30. Каталог вин.

Для каждого вина создана специальное окно, которое содержит информацию о выбранном вине (рис. 31).

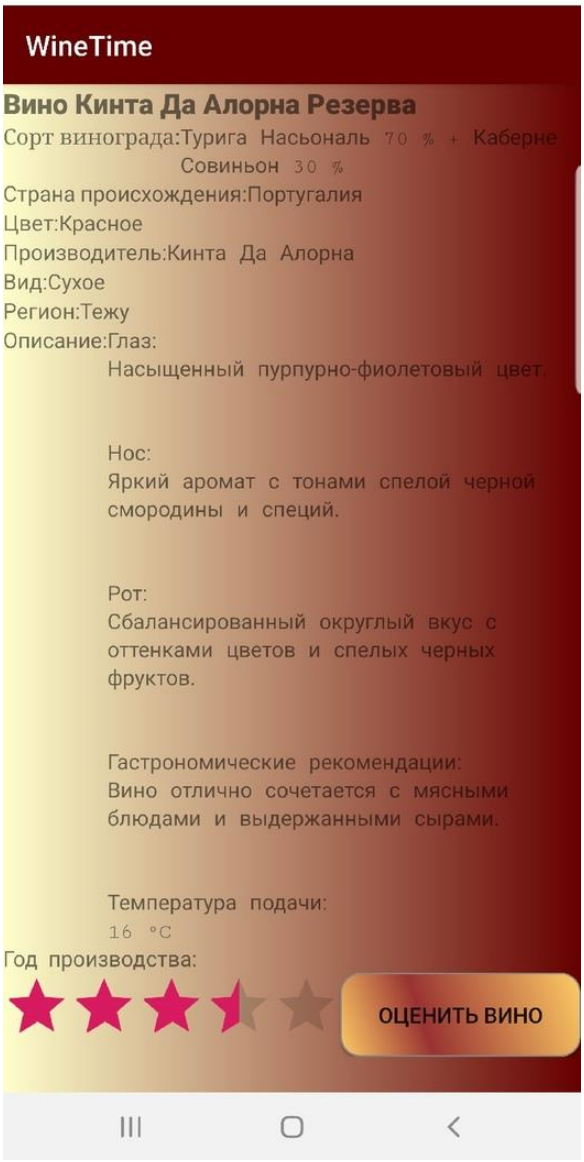

Рисунок 31. Информация о вине.

Затем была добавлена рекомендательная подсистема. Для него было подключено два вида кнопок;

— Получение рекомендаций среди всех вин;

— Получение рекомендаций к блюду.

В первом случае происходила работа рекомендательного модуля в стандартном режиме. Во втором случае, входные данные были уменьшены по объему до тех вин, которые содержат в себе гастрономические предпочтения к выбранным блюдам.

8 Тестирование

1. Тестирование рекомендательной подсистемы

Для тестирования рекомендательной системы сперва проверяется правильное разбиение сортов на группы. Для этого использовался источник [12], где есть несколько сгруппированных между собой сортов винограда. В ходе выполнения группировки была получена следующая группа (рис. 32):

```
String[] { "Каберне Совиньон", "Мерло", "Каберне Фран", "Карменер", "Бордо", "Санджовезе", "Канайоло"});
```

Рисунок 32. Пример полученной группы.

В источнике указано, что вина «Мерло», «Бордо», «Санджовезе» очень схожи между собой. Аналогичная ситуация обстоит и с другими группами. Соответственно, данный метод группировки показывает 100% результат на тестовой выборке, которой является информация из источника.

Тестирование влияния температуры не производится, так как данная информация взята из книги Роберта Паркера, который является главным экспертом в области виноделия.

После тестирования отдельных методов были протестированы рекомендации. Для этого был создан модуль тестирования, который в произвольном порядке выставляет оценки винам, а затем проверяется полученный результат.

2. Тестирование модуля распознавания текста.

Сначала производится автоматизированное тестирование модуля. Для этого, на вход модуля подается множество изображений, взятых из сети интернет. В случае успешного считывания информации с этикетки и нахождения информации в базе, система продолжает работу, однако если происходит какая-либо ошибка, то выдает сообщение. По результатам работы были получены ошибки, связанные с отсутствием вин в базе. На следующем этапе тестировалась правильность считывания, так как существовала вероятность схожести названий. Для этого в модуль загружаются изображения этикеток конкретных вин и проверяются в автоматическом режиме с ожидаемым результатом. По итогу не было получено никаких ошибок.

Следующий этап заключается в ручном тестировании модуля. Для этого было взято изображение нескольких бутылок вина. В результате было получено несколько ошибок при считывании в режиме видеопотока, которые были исправлены.

3. Тестирование графического интерфейса.

3.1 Тестирование поиска по каталогу в программном обеспечении.

При задании в строку поиска символов, результат совпадает с запросом. (рис. 33)

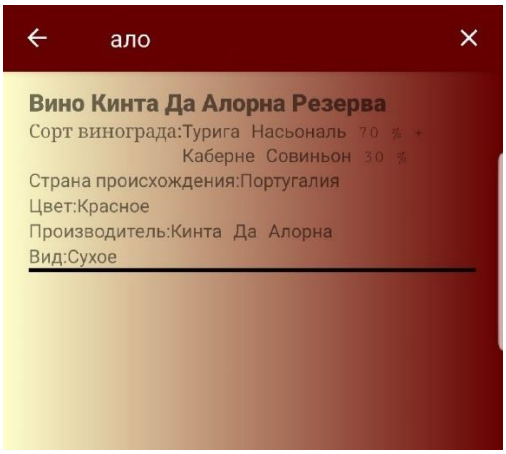

Рисунок 33. Поиск в каталоге.

3.2 Тестирование ограничений на входные данные при регистрации нового пользователя.

При вводе некорректных данных на экране появляется сообщение, которое позволяет пользователю понять, что не так и исправить это. (рис. 34-36)

Рисунок 34. Некорректный пароль.

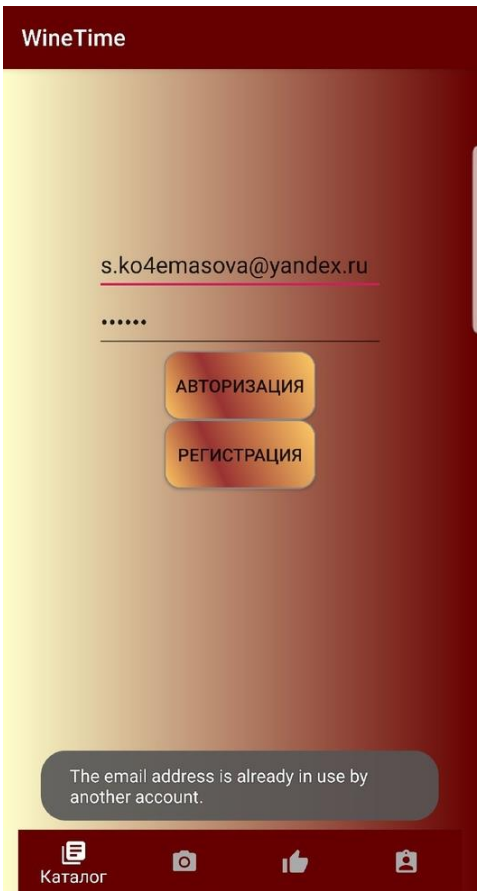

Рисунок 35. Попытка регистрации уже существующего email адреса.

Рисунок 36. Некорректный адрес электронной почты.

Заключение

В данной работе произведён анализ существующих методов построения рекомендательных систем и методов распознавания текста. Достигнута цель данной работы и выполнены все задачи на пути к ее достижению. В результате проделанной работы было получено программное обеспечение на базе операционной системы Android с функционалом распознавания текста и получением рекомендаций при выборе вина. Работа реализована при помощи среды разработки Android Studio и языка Java. Все моделирование происходило при помощи языка Python и дистрибутива Anaconda3.

Выявлен интерес к программным обеспечениям в области энологии и потребность в создании системы поддержки принятия решения при выборе вина, ввиду отсутствия однозначно качественной системы (раздел 1).

Аналитический обзор методов построения рекомендательных систем позволил выбрать для реализации рекомендательной подсистемы гибридный метод, включающий множество других методов построения рекомендательных систем, основанных на содержании и знаниях. В качестве основного показателя при построении рекомендаций используются оценки пользователя (раздел 2).

Для реализации распознавания текста использован встроенный в Firebase модуль, удовлетворяющий требованиям проеденного анализа существующих методов (раздел 3).

В результате проектирования системы разработан алгоритм построения рекомендаций для вин, использующий спроектированные в данной работе группы и матрицу температур (раздел 5.5).

В результате тестирования системы подтвержден теоретически ожидаемый результат рекомендаций.

Проведен обзор существующих технических решений и доказана актуальность задачи (раздел 1.5). Произведен обзор методов распознавания текста, а именно, каким образом он воспринимается системами, какую необходимо выполнить предобработку и сегментацию текста, а также каким способом строить данную подсистему в рамках поставленных задач (раздел 3) и спроектирована подсистема распознавания текста с

этикетки и вывода информации о вине на экран (раздел 5.4 и раздел 7.4), а также пользовательский интерфейс (раздел 7.5).

Произведен анализ методов построения рекомендательных систем и на основе него выбран гибридный метод, с функционалом других методов (раздел 2). Произведен анализ данных, связанных с вином, и спроектирована база данных (разделы 5.1-5.3 и раздел 7.1). На основе проведенного анализа в разделе 2, спроектирована рекомендательная подсистема (раздел 5.5 и раздел 7.2). Кроме того, для работы с данными пользователя был спроектирован модуль авторизации (раздел 7.3).

В дальнейшем планируется расширить каталог вин, снабдив его изображениями, и разработать метод поиска вина через камеру используя не только чтение текста этикетки, но и изображение этикетки, что позволит однозначно определять вина производства стран СНГ. Так же, планируется воспользоваться консультацией юриста, для дальнейшего выпуска приложения в магазин Google Play.

Список использованных источников

1. Иукуридзе Э. Ж., Егоров Б. В., Ткаченко О. Б. Современные представления о развитии технологии вина как науки.

2. Zhou Y., Jiang X. Dissecting android malware: Characterization and evolution //2012 IEEE symposium on security and privacy. – IEEE, 2012. – С. 95-109.

3. Ahmad M. S. et al. Comparison between android and iOS Operating System in terms of security //2013 8th International Conference on Information Technology in Asia (CITA). – IEEE, 2013. – С. 1-4.

4. WineSpectator [Электронный ресурс]: Главная страница. – Электрон. текст. дан. – Режим доступа: http://apps.winespectator.com/wineratingsplus/, свободный. – (Дата обращения: 20.02.2019).

5. Google Play [Электронный ресурс]: Delectable Wine. – Электрон. текст. дан. – Режим доступа: https://play.google.com/store/apps/details?id=com.delectable.mobile&hl=ru, свободный. – (Дата обращения: 20.02.2019).

6. ПОБОКАЛАМ [Электронный ресурс]: Зарисовки с MUST: основатель Vivino о самом популярном винном приложении мира. – Электрон. текст. дан. – Режим доступа: https://by-the-glass.ru/must-vivino/, свободный. – (Дата обращения: 20.02.2019).

7. Vivino [Электронный ресурс]: Главная страница. – Электрон. текст. дан. – Режим доступа: https://www.vivino.com/, свободный. – (Дата обращения: 20.02.2019).

8. Facebook [Электронный ресурс]: About us. – Электрон. текст. дан. – Режим доступа: https://www.facebook.com/pg/facebook/about/?ref=page_internal, свободный. – (Дата обращения: 20.02.2019).

9. Twitter [Электронный ресурс]: About us. – Электрон. текст. дан. – Режим доступа: https://about.twitter.com/ru.html, свободный. – (Дата обращения: 20.02.2019).

10. Google [Электронный ресурс]: Gmail. – Электрон. текст. дан. – Режим доступа: https://www.google.com/intl/ru/gmail/about/, свободный. – (Дата обращения: 20.02.2019).

11. Hello Vino [Электронный ресурс]: Главная страница. – Электрон. текст. дан. – Режим доступа: http://www.hellovino.com/, свободный. – (Дата обращения: 20.02.2019).

12. Parker R. M., Rovani P. A. Parker's wine buyer's guide. – Simon and Schuster, 2010.

13. Гомзин А. Г., Коршунов А. В. Системы рекомендаций: обзор современных подходов [Электронный ресурс]: труды ИСП РАН. 2012. URL: http://cyberleninka.ru/article/n/sistemy-rekomendatsiyobzorsovremennyh-podhodov (дата обращения 20.02.2019).

14. Лекция в Яндексе [Электронный ресурс]: Как работают рекомендательные системы. – Электрон. текст. дан. – Режим доступа: http://habrahabr.ru/company/yandex/blog/241455/ (Дата обращения 22.02.2019).

15. Habr [Электронный ресурс]: Рекомендательные системы. – Электрон. текст. дан. – Режим доступа: https://habr.com/post/176549/ , свободный. – (Дата обращения: 24.02.2019).

16. Cheriet M. Character recognition systems: a guide for students and practioners / M. Cheriet. – John Wiley & Sons, 2007. – 326 p.

17. Bernsen J. Dynamic thresholding of grey-level images / J. Bernsen // Proc. 8th ICPR. – 1986. – Vol.1 – P. 1251-1255.

18. Niblack W. An Introduction to Digital image processing / W. Niblack. – Prentice Hall, 1986. – 215 p.

19. Sauvola J. Adaptive document image binarization / J. Sauvola, M. Pietikainen // Pattern Recognition. – 2000. –Vol. 33 – P. 225–236.

20. Wolf C. Text localization, enhancement and binarization in multimedia documents / C. Wolf, J. M. Jolion, F. Chassaing // International Conference on Pattern Recognition. – 2002. –Vol. 4 – P. 1037–1040.

21. Bradley Adaptive Thresholding Using the Integral Image / D. Bradley, G. Roth // Journal of Graphics Tools. – 2007. – Vol. 12(2). – P. 13-21.

22. Stats-Browser S. C. G. OS, Search Engine including Mobile Usage Share.

23. KaiOS [Электронный ресурс]: Главная страница. – Электрон. текст. дан. – Режим доступа: https://www.kaiostech.com/, свободный. – (Дата обращения: 30.03.2019).

24. Casey E., Bann M., Doyle J. Introduction to windows mobile forensics. – 2010.

25. Raja H. Q. Tizen OS: Brief History, Roots, and Current Status. – 2014.

26. Studio A. Android Studio //The Official IDE for Android. – 2017.

27. Gosling J. et al. The Java language specification. – Addison-Wesley Professional, 2000.

28. Moskala M., Wojda I. Android Development with Kotlin. – Packt Publishing Ltd, 2017.

29. Van Rossum G., Drake F. L. The python language reference manual. – Network Theory Ltd., 2011.

30. Anaconda [Электронный ресурс]: Anconda distribution. – Электрон. текст. дан. – Режим доступа: https://www.anaconda.com/distribution/, свободный. – (Дата обращения: 20.02.2019).

31. Raggett D. et al. HTML 4.01 Specification //W3C recommendation. – 1999. – Т. 24.

32. Bray T. The javascript object notation (json) data interchange format. – 2014.

33. FireBase [Электронный ресурс]: ML-kit. – Электрон. текст. дан. – Режим доступа: https://firebase.google.com/docs/ml-kit/recognize-text , свободный. – (Дата обращения: 24.03.2019).

34. Dourish P. No SQL: The shifting materialities of database technology. – 2014.

35. Погребняк И. В., Тропченко А. Ю. РАСПОЗНАВАНИЕ СИМВОЛОВ НА ИЗОБРАЖЕНИЯХ, СОДЕРЖАЩИХ ИСКАЖЕНИЯ //Международный научно-исследовательский журнал. – 2017. – №. 5-3. – С. 81-86.

36. Fowler M., Kobryn C. UML distilled: a brief guide to the standard object modeling language. – Addison-Wesley Professional, 2004.

37. Berners-Lee T., Masinter L., McCahill M. Uniform resource locators (URL). – 1994. – №. RFC 1738.

38. Погода и климат [Электронный ресурс]: Архив погоды. – Электрон. текст. дан. – Режим доступа: http://www.pogodaiklimat.ru/ , свободный. – (Дата обращения: 24.04.2019).

Приложение 1

Часть базы данных вин