Санкт-Петербургский государственный университет

Топпер Алина Михайловна

Выпускная квалификационная работа

«Нейросетевые технологии и методы математической статистики для прогнозирования рейтингов анимационных произведений»

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2020 «Прикладная математика, фундаментальная информатика и программирование»

Научный руководитель:

доцент, кафедра технологии программирования,

кандидат физико-математических наук,

Сергеев Сергей Львович

Рецензент:

генеральный директор,

общество с ограниченной ответственностью «Капитал Программ»,

кандидат физико-математических наук,

Пашкевич Василий Эрикович

Санкт-Петербург,
2024 г.

Saint Petersburg State University
Department of Technology of Programming

Topper Alina

Final qualification paper

Neural Network Technologies and Mathematical Statistics Methods for Forecasting Ratings of Animated Works

Education level: Bachelor's degree

Education program 01.03.02 «Applied Mathematics, Fundamental Informatics and Programming»

\

Supervisor:

Associate Professor, Department of Programming Technology,

candidate of physical and mathematical sciences,

S. L. Sergeev

Reviewer:

General Director,

Limited Liability Company “Capital Program”,

Candidate of Physical and Mathematical Sciences,

Pashkevich Vasilii Ericovich

Saint Petersburg 2024

Содержание

Введение

Рейтинг - это основной критерий, по которому судят практически любое произведение. Ожидаемый рейтинг показывает окупаемость товара и, следственно, необходимость его производства. Именно поэтому так важно уметь прогнозировать популярность будущего продукта. Продукт с высоким прогнозируемым рейтингом стоит производить в первую очередь, потому что на него будет спрос.

Чаще всего рейтинг встречается у кинофильмов, сериалов, шоу и анимационных произведений. В данной работе будет представлена модель, прогнозирующая оценку японских мультипликационных шоу - аниме. Этот вид анимации было решено выбрать из-за особенностей выпуска работ. Каждый квартал почти одновременно выходят около 30 произведений, в то время как у других кинолент нет такого строгого графика. Также японская анимация не имеет такую же освещенность в СМИ, как фильмы. Предполагается, что рейтинги аниме более корректные, так как на них не влияют общественное мнение и рецензии именитых критиков. Наконец, популярность аниме зависит от меньшего количества параметров.

Известность японской мультипликации во всем мире растет с каждым годом. И так как эта ниша стремительно развивается заграницей, а не только в родной Японии, то она стала одной из значимых категорий во внешней экономике родной страны. Международный рынок в 2021 улучшился на 109,4% в сравнении с 2019 годом [1]. Популярность аниме продолжает расти с 2000-х годов, когда такой контент, как «Pokémon», распространился за границу. После этого индустрия пережила немало трудностей, таких как крах рынка видео, повальное интернет-пиратство, ограничения на вещание в Китае, финансовый кризис 2008 года и рост курса иены. Тем не менее, с улучшением качества интернет-услуг в середине 2010-х годов, китайской массовостью покупок легального контента под руководством правительства

и быстрым развитием американских стриминговых платформ международное поле стало стремительно развиваться. Кроме того, этот импульс сохранялся даже во время пандемии COVID-19, поскольку количество людей, остающихся дома, стало стимулом для дальнейшего роста, и в 2020 году он обогнал внутренний рынок. Международный рынок вырос в три раза за период с 2015 по 2017 год и продолжает расти. Все больше популярных западных стриминговых сервисов лицензируют аниме, а также спонсируют свои собственные производства. С таким успехом аниме привлекает огромное количество международного внимание, что заставляет аниме-студии пересмотреть свои взгляды на то, как обратиться к более глобальному рынку, если они хотят добиться международного успеха.

В данной работе представлена математическая модель с 4-мя входными параметрами и 3-мя дополнительными параметрами, которая предсказывает рейтинг произведения по десятибалльной шкале, основываясь на средних значениях параметров в предыдущем сезоне, со среднеквадратическим отклонением 0.3 (на момент 21.04.24). Эту модель можно экстраполировать на другие типы прогнозов, для которых известны исходные данные до выхода продукта, заданные дискретно в некотором временном интервале и имеющие либо категориальные, либо числовые значения.

Это исследование может быть полезно, как и для создателей, так и для целевой аудитории. Аниме-студиям необходимо понять, какими признаками должна обладать их продукция, чтобы в будущем привлекать международную аудиторию, а не только японскую [2, 3]. Командам, занимающимся локализацией, также необходимо предсказать, какие аниме, вышедшие недавно, будут достаточно успешными, чтобы их стоило локализовать на свой язык. Зрители, скорее всего, сначала проверят описание аниме, прежде чем решат его посмотреть, поэтому понимание того, какие параметры наиболее привлекательны для аудитории, принесет этим группам наибольшую пользу.

Постановка задачи

Целью данной дипломной работы является разработка программного продукта, прогнозирующего рейтинг аниме, с помощью методов математической статистики и нейронных сетей. Также для более удобного использования было создано приложения на базе ОС Android. Для достижения поставленной цели решаются следующие основные задачи:

- Обзор предметной области
- Создание математической модели прогнозирования
- Написание программы для сбора данных. Формирование корпуса данных
- Анализ данных
- Разработка мобильного приложения, реализующего расчет конечного прогноза рейтинга

Обзор Литературы

Работ, связанных с прогнозированием рейтингов именно для аниме, крайне мало, потому что эта сфера только начала обретать популярность и привлекать внимание исследователей. Обсудим несколько работ, наиболее похожих на данное исследование.

В [4] была опубликована работа, которая предсказывает успех аниме по отзывам зрителей, с помощью векторного представления слов (word2vec) и сверточных нейронных сетей (CNN). В данной статье основное внимание уделяется созданию точной нейросетевой модели с использованием отзывов с «MyAnimeList» (MAL) для обучения и оценки. В результате исследования итоговая модель имеет оценку F1 (точность, когда классы не сбалансированы) равную 0,97.

В другой статье [5] авторы пытаются предсказать и дать прямые ответы на вопрос о том, что может стать популярным в ближайшие годы, анализируя тенденции изменения данных с помощью временных рядов из библиотеки Prophet для языка R. Исследователи используют такие данные как: индекс произведения, жанр, рейтинг и момент выхода этого аниме. Точность их прогноза составляет более 80%. В отличие от них, в текущей работе рассматривается более сложная конфигурация параметров, а также предпринимается попытка выявить более долгосрочные тенденции в изменении важности параметров, а не только за последние несколько лет.

В исследовании [6] успех аниме был предсказан по краткому описанию. Синопсисы были исследованы путем их преобразования в векторное представление с использованием n-грамм и деревьев зависимостей. Это позволило применить метод опорных векторов (SVM), гауссовский наивный байесовский классификатор и логистическую регрессию для изучения связи между кратким описанием аниме и его успехом.

В статье [7] пытались выявить наиболее значимые параметры для прогнозирования и предсказать рейтинг анимационных произведений с помощью линейной регрессии. Было выявлено, что жанр анимации больше всего влияет на рейтинг. По данному исследованию ни источник адаптации, ни сценарист не оказывают существенного влияния на рейтинг. На данном этапе точность рассматриваемой модели невысока из-за недостаточного объема данных. В данной работе, наоборот, предполагается, что оценка первоисточника имеет существенной влияния на конечный рейтинг.

Теперь обратимся к другим статьям с похожей задачей, но на базе фильмов. В этой сфере прогнозирование рейтингов имеет более широкое представление, а также рассматриваются различные подходы для решения этой задачи. В работе [8] представляется сравнение различных моделей машинного обучения, методов кластеризации и нейронных сетей. Указанные модели сравнивались по целому ряду факторов, включая их точность на обучающих и проверочных наборах данных, а также на тестовом наборе данных, наличие новых характеристик фильмов и ряд других статистических показателей. Исследователи смогли выявить наиболее значимые характеристики для успеха фильма. Нейронная сеть с точностью 86% показала наилучший результат среди всех рассмотренных моделей.

В статье [9] была цель найти лучшую модель для прогнозирования успеха фильмов и шоу Netflix с использованием различных атрибутов: тип, год выпуска, возрастное ограничение, время просмотра, жанр, страна производства. Для прогнозирования успеха фильма или шоу применялись алгоритмы машинного обучения такие, как метод случайного леса (Random Forest), наивный Байес (Naive Bayes) и метод k-ближайших соседей (k-Nearest Neighbors). Лучшим алгоритмом в этом случае стал метод случайного леса.

И в другой работе, которая тоже основывалась только на данных доступных до производства фильма, [10] лучшим алгоритмом оказался метод

случайного леса. Также особенностью данной работы является исследование различий между результатами самых коротких и самых длинных выборок с точки зрения времени. Сравнение было проведено между методом случайного леса, методом опорных векторов и нейронной сетью.

В одном отечественном сборнике приведено 2 статьи, в которых отчетливо наблюдается проблема влияния на зрительский рейтинг неопределенного количества факторов. В одной из них [11] было выявлено 43 значимых параметра, в другой же [12] - 11 факторов. Помимо нерешенной проблемы количества параметров и их веса в итоговой формуле, существует задача о минимизации отклонения прогнозируемого результата и реального. В статье [13] достигается одна из самых высоких возможных точностей - 80.29%. Но такая точность, в общем случае, все же не может гарантировать истинность предсказанного рейтинга.

Как упоминалось ранее, в задаче прогнозирования одна из проблем - неизвестное число параметров, которые надо учитывать в модели. Эта проблема решается в работе [14], в которой особенное внимание уделяют выявлению наиболее значимых факторов и построению модели, основанной именно на них. В большинстве предыдущих исследований, где использовались методы машинного обучения, основное внимание уделялось увеличению предсказательной способности модели, без учета важности выбранных характеристик.

Рассмотрим еще одну популярную архитектуру в нейронных сетях - трансформеры. Это архитектура глубоких нейронных сетей, основанная на механизме внимания без использования рекуррентных нейронных сетей. В задаче прогнозирования рейтинга трансформеры чаще используются для обработки текстовых данных фильма, например, преобразование краткого содержания в числовые вектора. А затем другая сеть с помощью полученных векторов прогнозирует рейтинг, к примеру, такой подход предложен в работе [15]. Была предложена новая задача - предсказание рейтинга фильмов с

учетом мульти-модальных данных (MAMRP), то есть текстовых, визуальных и звуковых. Для этого была создана новая модель с несколькими модулями. Один из этих модулей основан на трансформере, который извлекает информацию о фильмах. Другие модули объединяют данные из разных источников и анализируют их для более точного предсказания. Сама модель основана на дереве, в котором используется графовая нейронная сеть. В экспериментах было показано, что метод работает лучше на 24% по сравнению с уже существующими методами прогнозирования.

Глава 1. Обзор методов

В данной работе для задачи прогнозирования предлагается использовать сочетание нейронной сети, аппроксимации методом наименьших квадратов (МНК) и дисперсии для определения веса параметров. В первой главе будут рассмотрены часто используемые методы машинного обучения и нейронные сети. Также рассмотрим метрики качества, которые понадобятся в дальнейшем.

1.1 Методы машинного обучения

А. SVR (Регрессия опорных векторов)

Основная идея метода регрессии опорных векторов заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным способом. Алгоритм работает в предположении, что чем больше расстояние между разделяющей гиперплоскостью и объектами разделяемых классов, тем меньше будет средняя ошибка классификатора. При использовании SVR модели представляется большая гибкость и можно самим определить, насколько допустима ошибка [16]. Пока предсказания находятся в приемлемом диапазоне, можно свободно найти наилучший вариант для исследуемых данных.

Б. Random Forest (Метод случайного леса)

Random Forest - это метод, который заключается в построение ансамбля независимых решающих деревьев. Решающее дерево предсказывает значение целевой переменной с помощью применения последовательности простых решающих правил, которые называются предикатами. Как упоминается в [17], каждое дерево зависит от значений случайного вектора, который имеет распределение, общее для всех деревьев в лесу, но является независимым в отношении выборки.

B. k-NN (Метод k-ближайших соседей)

kNN представляет собой непараметрический алгоритм, который в задаче регрессии использует среднее целевое значение k ближайших соседей для прогнозирования неизвестных данных. Кроме того, соседям могут быть присвоены веса так, чтобы более близкие соседи вносили больший вклад в прогнозирование. Метод использует идею близости для принятия решений о регрессии новых объектов на основе их сходства с ближайшими соседями из обучающего набора данных. Недостатком этого метода является высокая сложность одного прогноза.

Г. Lasso (Лассо регрессия) и Ridge (Гребневая регрессия)

Lasso и Ridge основаны на простой линейной регрессии в виде метода наименьших квадратов, к которой добавляют дополнительный штрафной параметр в качестве регуляризации для повышения точности. В Lasso этим штрафом является сумма квадратов весов. С другой стороны, Ridge вычисляет сумму абсолютных значений весов.

Д. Gradient boost (Градиентный бустинг) и XGBoost (Экстремальный градиентный бустинг)

Градиентный бустинг [18] использует слабые обучающие элементы в виде деревьев решений и последовательно преобразует их в более сильные. После оценки каждого дерева и соответствующих ему весов, веса следующего дерева настраиваются на основе наблюдаемых результатов. Более надежная и эффективная реализация с открытым исходным кодом Gradient Boost является XGBoost, как описано в [19].

1.2 Нейронные Сети

A. Сверточные нейронные сети (CNN)

Идея сверточных нейронных сетей заключается в чередовании сверточных слоев (convolution layers) и объединяющих слоев (pooling layers).

CNN может быть адаптирована для регрессии, изменяя последний слой так, чтобы он имел один выходной узел. Сверточный слой - основной тип слоев модели, здесь происходит применение операции свертки. В сверточном слое каждый «нейрон» соединяется с небольшим локальным участком предыдущего слоя, а не со всеми нейронами. Это позволяет учесть локальные свойства данных и снизить количество параметров сети. Слой объединения выполняет операцию уменьшения размерности получаемых данных. Слой объединения устойчив к небольшим трансформациям входных данных, что помогает избежать переобучения и уменьшает число параметров сети.

Входные данные передаются через сверточные слои, в которых каждый узел соединен только со своими ближайшими соседями. Эти слои имеют свойство сжиматься с глубиной, причём обычно они уменьшаются на какой-нибудь из делителей количества входных данных, часто используются степени двойки.

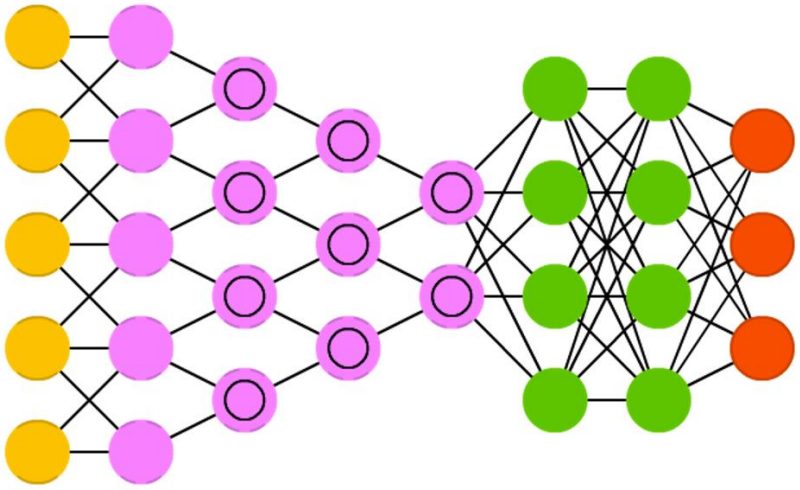

Рисунок 1.1 — Структура CNN

Сверточная нейронная сеть является одной из самых часто используемых в задаче прогнозирования успеха фильма.

Например, в статье [20] для прогнозирования кассовых доходов фильмов предлагается объединение нескольких нейронных сетей. Одной из них была CNN, включающая в себя входные данные об особенностях

постеров фильмов. Сверточная нейронная сеть была использована для извлечения признаков из постеров фильмов. Путем предварительного обучения CNN можно изучить функции, имеющие отношение к кассовым сборам фильмов. Затем была создана другая нейронная сеть, которая использовала как функции постеров фильмов, так и другие данные, связанные с фильмами, для прогнозирования кассовых доходов фильмов. Таким образом, предложенная модель использовала контент постеров, а не только традиционные данные для прогнозирования кассовых сборов фильмов.

Чтобы устранить ограничения предыдущих исследований, в работе [21] в качестве параметров были использованы исторические ценности фильма - факторы, известные до релиза. Эти факторы могут быть сгенерированы из элементов предыдущего фильма, и созданы на основе отношений между фильмами, таких как актер, режиссер, жанры, рейтинг контента и производственные компании. Используя исторические значения, можно сделать объективное предсказание еще до выхода фильма в прокат. Предложенный метод предназначен для более точного и общего прогнозирования рейтинга фильмов. В данном исследовании использование исторических признаков и CNN в качестве модели показало многообещающий результат.

В статье [22] в качестве объекта исследования взято прогнозирование оценки фильма, а также создана модель для этой задачи, основанная на сверточной нейронной сети. С помощью экспериментов был получен результат, что десятислойная сверточная нейронная сеть имела наилучшую производительность. Было установлено, что точность работы данной модели на тестовом наборе всегда составляла около 56%, а точность на обучающемся наборе увеличивалась с увеличением времени обучения. Исследователи сравнивали её с моделью прогнозирования, основанной на дереве решений.

Б. Рекуррентные нейронные сети (RNN)

RNN — сети с циклами, которые хорошо подходят для обработки последовательностей. Нейроны получают информацию не только из предыдущего слоя, но и от себя из предыдущего прохода. Это означает, что порядок, в котором передаются входные данные и обучается сеть, имеет значение. Поскольку одни и те же параметры используются на всех временных этапах в сети, градиент на каждом выходе зависит от предыдущих временных шагов. Одной из больших проблем с RNN является проблема исчезновения (или взрыва) градиента, когда, в зависимости от используемых функций активации, информация быстро теряется с течением времени. В принципе, RNN можно использовать во многих областях, поскольку большинство форм данных, которые фактически не имеют временной шкалы, могут быть представлены в виде последовательности.

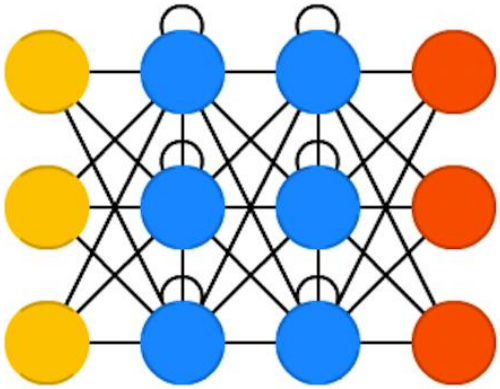

Рисунок 1.2 — Структура RNN

В задаче прогнозирования успеха фильма чаще используют LSTM (долгая краткосрочная память) - она является расширением RNN лишь с тем отличием, что пробуют побороть проблему взрывного градиента, используя фильтры и блоки памяти. У каждого нейрона есть три фильтра: входной фильтр, выходной фильтр и фильтр забывания. Задача этих фильтров — сохранять информацию, останавливая и возобновляя ее поток. Входной фильтр определяет количество информации с предыдущего шага, которое будет храниться в блоке памяти. Выходной фильтр занят тем, что определяет,

сколько информации о текущем состоянии узла получит следующий слой. Наличие фильтра может быть полезно в некоторых ситуациях, например, если нейросеть запоминает книгу, в начале новой главы может быть необходимо забыть некоторых героев из предыдущей.

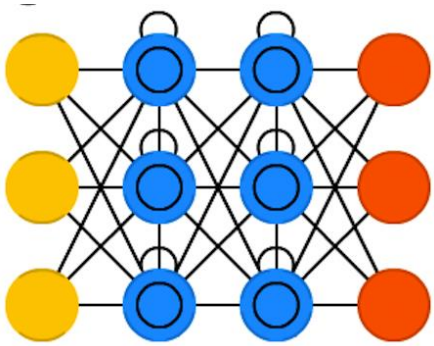

Рисунок 1.3 — Структура LSTM

Использование LSTM в задаче прогнозирования фильмов крайне редко встречается, чаще эта модель используется для какой-то подзадачи, например обработка рецензий фильма. В одной из немногих статей [23], в которых LSTM используется для предсказания успеха фильма, была построена модель на основе различных особенностей фильма, таких как съемочная группа, сюжет фильма, кассовые сборы, обзоры зрителей и критиков. В этой работе было проведено детальное исследование некоторых алгоритмов машинного обучения и нейронных сетей, в числе которых были LSTM и CNN. По результатам классификатор XGBoost показал наилучшую точность.

В. Многослойный персептрон (MLP)

MLP состоит минимум из трех слоев: входного, скрытого и выходного. За исключением входных, все нейроны использует нелинейную функцию активации. При обучении MLP используется обучение с учителем и алгоритм обратного распространения ошибки. Связи между нейронами имеют соответствующие веса, которые изучаются в процессе обучения. Эти веса определяют силу связей и играют решающую роль в способности сети улавливать закономерности в данных.

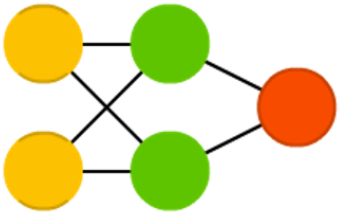

Рисунок 1.4 — Структура MLP

MLP является второй по популярности нейронной сетью, после CNN, для задачи прогнозирования успеха фильма. Рассмотрим несколько статей, использующих данную архитектуру.

В работе [24] используют данные по 241 китайско-американскому фильму и пытаются разделить каждый фильм на 6 классов в зависимости от кассового успеха на основе 11 взвешенных непрерывных переменных. В модели использовалась MLP из 30 узлов в первом скрытом слое и 10 узлов во втором скрытом слое. Результаты были не очень точными при классификации фильмов по их фактическому классу, но имели высокую относительную точность, что указывает на то, что прогнозируемый класс отличался на один класс от фактического класса

Аналогичное исследование провели в [25]. Там использовали 834 фильма 1998–2002 годов и пропускали их через MLP. Исследователи пытались отнести каждый фильм к одному из 9 классов на основе его кассовых сборов, используя 7 непрерывных переменных и 10-кратную перекрестную проверку. В результате уровень точности составил 75,1% в одной категории.

В работе [26] модель прогнозирует приблизительный уровень успеха фильма на основе его прибыльности, анализируя исторические данные из различных источников. Исследователи классифицируют фильмы по 5 категориями, ориентируясь на финансовый успех фильма. Если модель MLP обучается с использованием только параметров, известных до релиза фильма, то она показывает точность 84,1 %.

1.3 Метрики качества

Методы машинного обучения и нейронные сети будут сравниваться с помощью оценки R^2 – это один из показателей оценки эффективности моделей на основе регрессии.

$$R^2 = 1 - \frac{\sum e_t^2}{\sum(y_t - \check{y}t)^2}, \quad (1)$$

R^2 — коэффициент детерминации,

e_t^2 — средняя квадратичная ошибка,

y_t — верное значение,

$\check{y}_t$ — среднее значение.

Для сравнения результатов конечной модели с истинными значениями будем использовать функцию RMSE (корень из среднеквадратической ошибки), т. к. она измеряется в тех же единицах, что и значение ответа, нежели среднеквадратическая ошибка (MSE).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \check{y}i)^2}, \quad (2)$$

где n — количество наблюдений по которым строится модель и количество прогнозов,

y_i — фактические значение зависимой переменной для i-го наблюдения,

$\check{y}_i$ — значение зависимой переменной, предсказанное моделью.

Глава 2. Работа с данными и анализ

2.1 Формирование данных и постобработка

Было проведено множество различных исследований с целью прогнозирования общего рейтинга фильма на основе доступных данных, таких как информация о фильме или даже обзоры в социальных сетях. Однако некоторые из упомянутых исследований использовали метаданные, которые доступны только после выхода фильма. Но лучшим подходом было бы использовать информацию, полученную до даты выхода фильма, поскольку это позволяет продюсерам фильма и инвесторам принимать менее рискованные решения.

А. Извлечение данных

Данные получаем с помощью http-запроса с сайта «jikan.moe» [27]. Он берет данные с другого сайта – MyAnimeList (MAL). MAL представляет собой каталог аниме и манги (литературный первоисточник). Он хранит информацию о более чем 17,5 тысячах аниме. Для получения данных используется встроенный в Java Development Kit (JDK), то есть набор инструментов для разработки на языке Java, пакет «java.net.http»[28].

Из всего набора готовых запросов, представленных на сайте, нам нужно только три: getSeason, getAnimeRelations и getMangaById.

В запросе getSeason можно указывать нужный сезон - зима, весна, лето, осень, - и год, это будет полезно для создания csv таблиц, каждая из которых будет отнесена к конкретному сезону и году. В ответе получаем json-объект, который нужно распарсить и записать в csv таблицы.

Получаем объкты «data» и «pagination». В основном мы будем работать с данными в «data». Из «pagination» нужен ключ «has_next_page», который указывает, существует ли следующая страница, содержащая произведения.

Если такой страницы нет, то заканчиваем работу. На каждой странице отображается по 30 произведений.

Из объекта «data» нам понадобятся следующие ключи: «mal_id», «score», «rating», «studios», «genres», «themes», «demographics».

По запросу getAnimeRelations с помощью полученное по ключу «mal_id» значения получаем ответ, содержащий объект «data», из которого достаем значение по ключу «relation» – это будет строка, в которой написан источник. Если для произведения источником является «Adaptation», то есть произведение имеет литературный первоисточник, то переходим к рассмотрению массиву объектов «entry», из которого по ключу «mal_id» достаем уникальный номер первоисточника.

Наконец, по запросу getMangaById с помощью «mal_id» для источника можем получить ответ, содержащий объект «data». Из него достаем по ключу «score» числовое значение рейтинга манги.

Описание параметров представлено на Рисунке 2.1.1.

Название параметра	Описание	Тип
id anime	Уникальное ID используемое MAL для каждого произведения, есть набор цифр	int
score anime	Рейтинг аниме	double
studio	Название студии, создавшее аниме	string
genre	Жанры, разделяются запятой	string
id manga	Уникальное ID используемое MAL для каждого произведения, есть набор цифр	int
score manga	Рейтинг манги	double
theme	Темы - ключевые слова, разделяются запятой	string
rating	Возрастное ограничение	string
demographic	Разделени по полу и возрасту: Kids - для обоих полов до 12 лет Seinen - для юношей старше 18 Shoujo - для девушек 12-18 лет Shounen - для юношей 12-18 лет Josei - для девушек старше 18	string

Рисунок 2.1.1 — Описание параметров в датасете

Б. Обработка данных

Чтобы улучшить производительность модели на предоставленных данных и подготовить набор данных для дальнейших исследований, используем определенные методы предварительной обработки. Сначала откажемся от тех параметров, которые доступны только после выхода фильма такие, как количество оценок, популярность, количество пользователей, оценивших произведение и так далее. В случае текстовых параметров название фильма и исходное название были удалены. Также не были включены такие параметры, как трейлер, синопсис и постер. Такие параметры как продюсеры, лицензоры и время вещания по телевидению тоже не были учтены. Продюсеров исключили, так как мало кто из зрителей смотрят на их имена, так как за качество истории отвечает первоисточник, а за качество визуализации - студия. Поэтому рассматривать продюсеров нет необходимости. Время вещания тоже не важно, так как все произведения можно смотреть в общем доступе в интернете в любое время.

Было выделено четыре важных параметра: рейтинг аниме, название студии, жанр и рейтинг манги. Дополнительные параметры: тема, возрастной рейтинг и демография.

Уже на этапе создания таблиц была проведена предобработка: удалялись пустые значения для главных параметров, то есть если для произведения хотя бы один параметр был 0 или NaN, то такое аниме не попадало в датасет.

Было создано два набора данных: один, в котором оставляли произведение если у него были пустые дополнительные параметры (тема, возрастной рейтинг, демография) и заполняли вместо пустых значений наиболее часто встречаемыми значениями (например, если неизвестен параметр демография, то заполняем исходя из жанра. Для комедии наиболее встречаемая демография - Seinen, а для жанра романтики - Shoujo). Во

втором наборе данных удалялись произведение хотя бы с одним пустым значением у дополнительных параметров.

Был выбран второй датасет, так как хоть он и меньше первого практически в два раза, но более точный. В первом датасете не учитываются выбросы. Например, для произведения с жанром комедии была записана демография seinen, но в каждом третьем случае это может быть неправильно, из-за чего возникает смещение. Конечный датасет содержит более чем 450 произведений в период с 2015 до 2023 включительно. Также существует расширенный датасет с 2000 годов, содержащий более 1600 наименований, но он считается менее релевантным для прогнозирования, так как аниме стало популярным за рубежом только после 2015 года. То есть оценки произведений с 2000 по 2015 либо сформированы японскими зрителями, либо зрителями, которые смотрели эти аниме намного позже их выхода, что делает их не показательными. Но он лучше подходит для анализа данных.

В результате был получен набор csv таблиц, пример которой представлен на Рисунке 2.1.2.

id_anime	score_anime	studio_name	genres	id_manga	score_manga	theme	rating	demographic
9919	7.49	A-1 Pictures	Action	13492	7.89	Mythology	PG-13 - Teens 13 or older	Shounen
6880	7.15	Manglobe	Action	3986	7.95	Gore	R - 17+ (violence & profanity)	Shounen
10165	8.46	Kyoto Animation	Comedy	3082	8.37	Gag Humor	PG-13 - Teens 13 or older	Shounen
9969	9.04	Sunrise	Action	44	8.62	Gag Humor	PG-13 - Teens 13 or older	Shounen
10080	7.88	Manglobe	Comedy	7519	8.47	Harem	PG-13 - Teens 13 or older	Shounen
9863	8.21	Tatsunoko Production	Comedy	1414	8.38	School	PG-13 - Teens 13 or older	Shounen
10711	7.2	Barnum Studio	Comedy	25675	7.63	School	PG-13 - Teens 13 or older	Seinen
10271	8.25	Madhouse	Suspense	3573	8.4	Adult Cast	R - 17+ (violence & profanity)	Seinen
9736	6.49	DiomedГ©a	Comedy	6778	7.13	Harem	PG-13 - Teens 13 or older	Seinen

Рисунок 2.1.2 — Пример получившейся таблицы

2.2 Анализ данных

А. Анализ данных

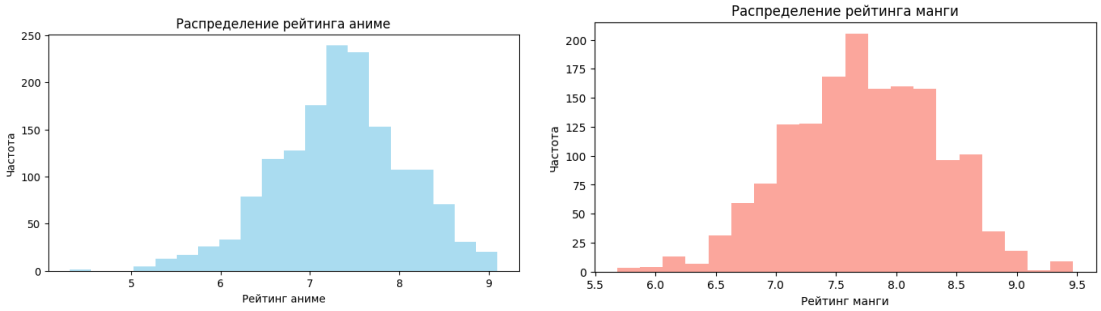

Рисунок 2.2.1 — Распределение рейтингов

Как можно заметить на графиках, рейтинг аниме колеблется в основном от 5 до 9, тогда как рейтинг манги изменяется от 5.5 до 9.5. У манги рейтинг в среднем смещен на 0.5 в большую сторону по сравнению с аниме, это также видно при более детальном анализе конкретных произведений. Этот факт будет важен при анализе результатов в конечной модели прогнозирования рейтинга.

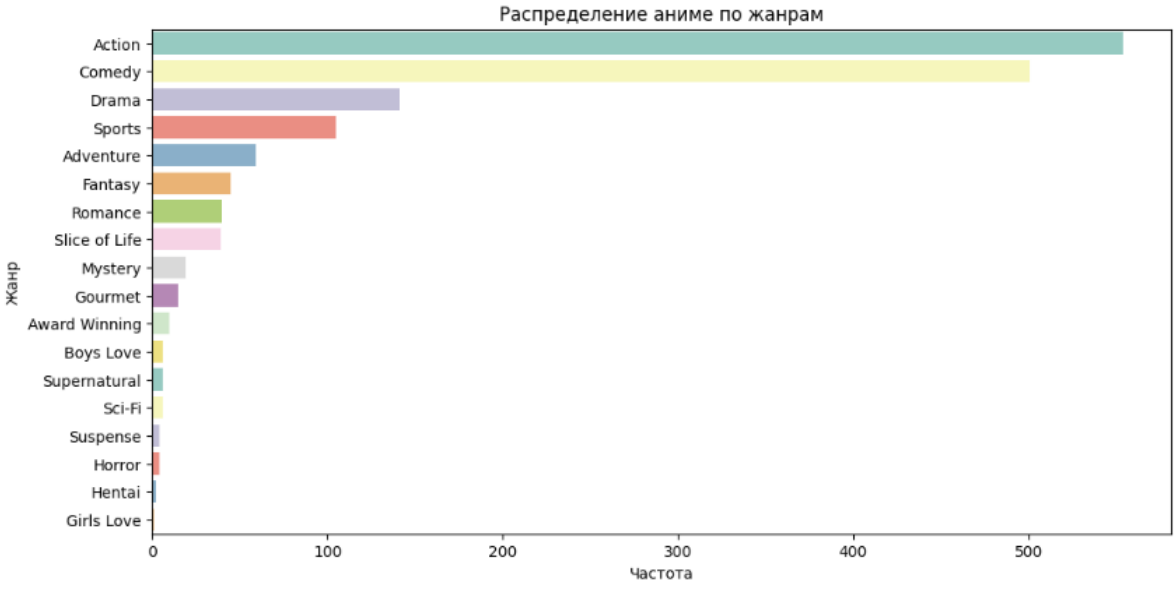

Рисунок 2.2.2 — Распределение аниме по жанрам

Исходя из этой гистограммы видно, что наиболее часто встречаемый жанр – экшен (Action). Очевидно, что чем более распространен жанр, тем менее он важен для конечного расчета. Это связано с тем, что из-за большого количества произведений получается слишком большой разброс по рейтингам.

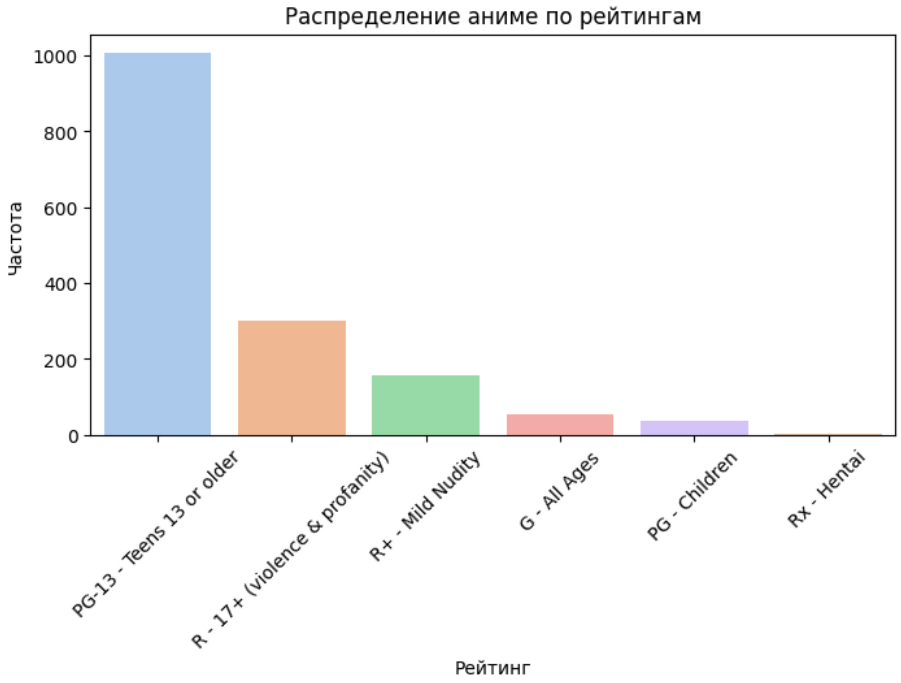

Рисунок 2.2.3 — Распределение аниме по возрастным рейтингам

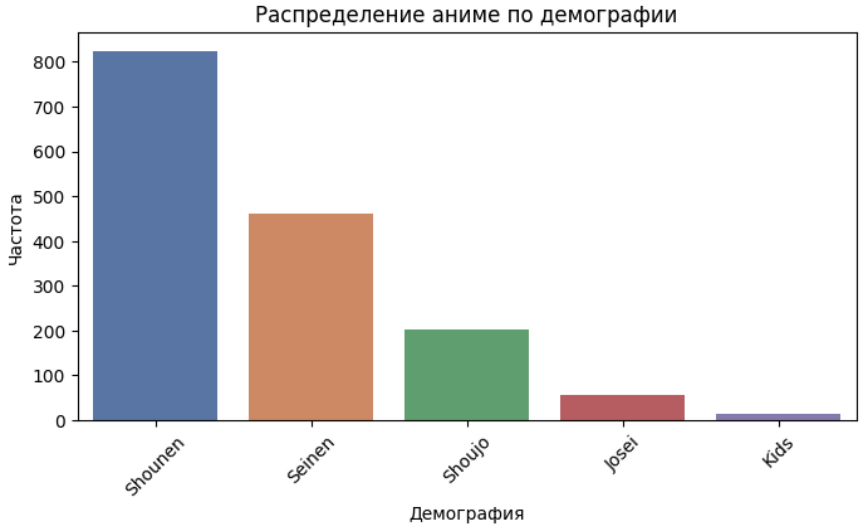

Рисунок 2.2.4 — Распределение аниме по демографии

Подобный вывод аналогичен для Рисунков 2.1.3 и 2.1. 4. Рейтинг «PG-13» и демография «Shounen» наиболее распространены и, следственно, больше всего подвержены случайным выбросам и наименее важны в конечной формуле.

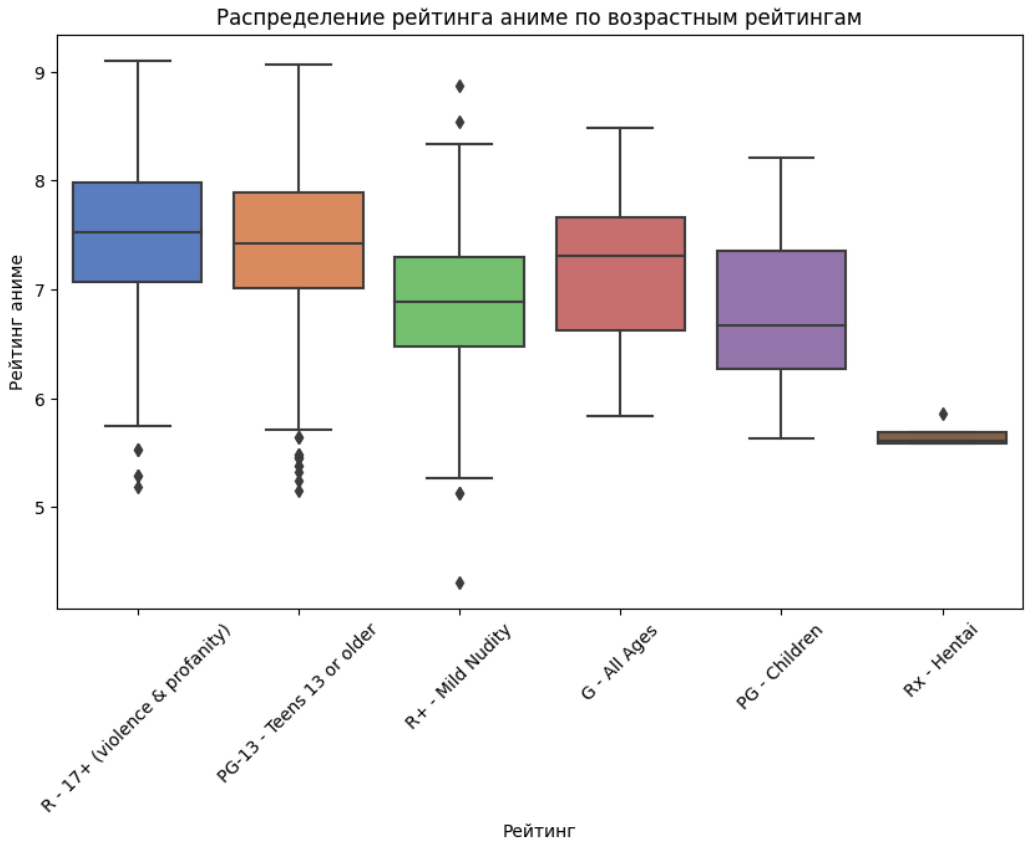

Рисунок 2.2.5 — Распределение рейтинга аниме по возрастным рейтингам

Из Рисунка 2.2.5 можно предположить, что аниме с возрастным рейтингом «R-17+» будет чаще всего иметь оценку от 7 до 8, а с рейтингом «PG» - от 6.5 до 7.2.

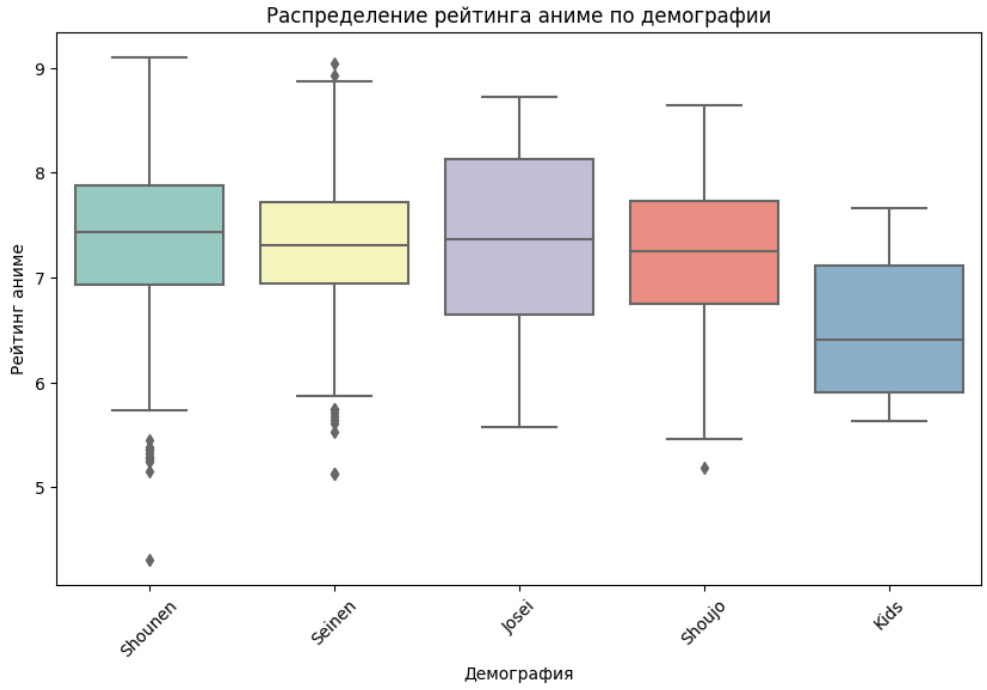

Рисунок 2.1.6 — Распределение рейтинга аниме по демографии

Из Рисунка 2.1.6 можно сделать вывод, что произведения с категорией «Kids» всегда будет иметь оценку от 5.8 до 7.8, а в категории «Shounen» могут быть как и самые низкие оценки - около 4, так и самые высокие - больше 9.

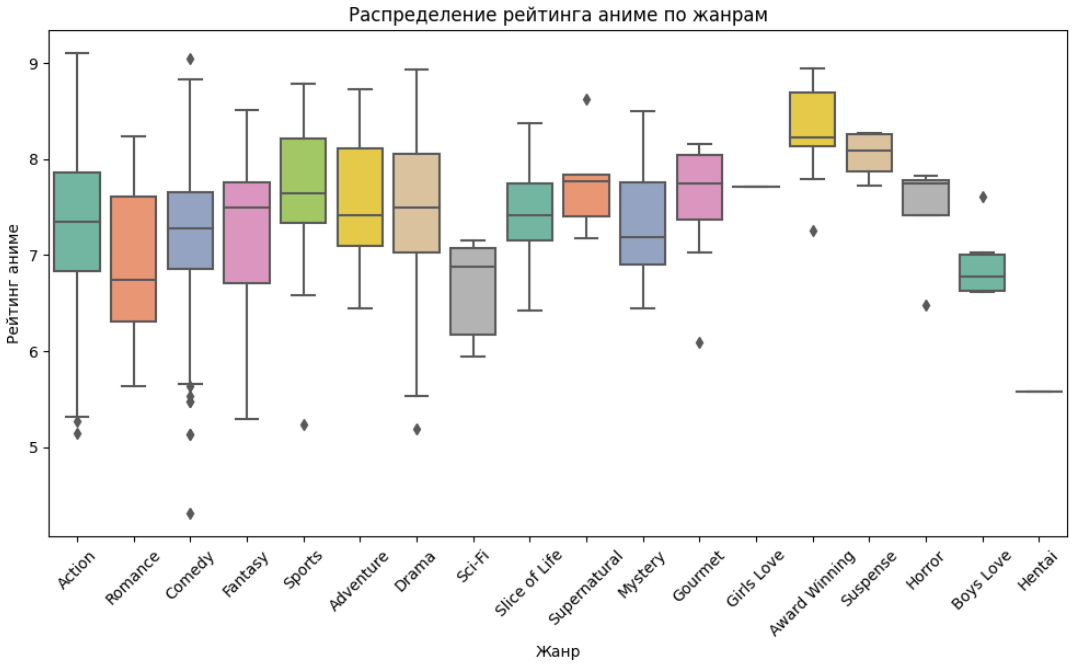

Рисунок 2.1.7 — Распределение рейтинга аниме по жанрам

Из Рисунка 2.1.7 можно выделить жанры, которые менее подвержены выбросам, а значит, будут давать более точную оценку при прогнозе. Например, если мы захотим спрогнозировать рейтинг аниме по таким жанрам, как: «Action», «Comedy» и «Drama», то результаты могут сильно отличаться от истинных. Тогда как прогноз по менее частым жанрам таким как: «Sci-Fi» и «Mystery» покажут результат близкий к действительному.

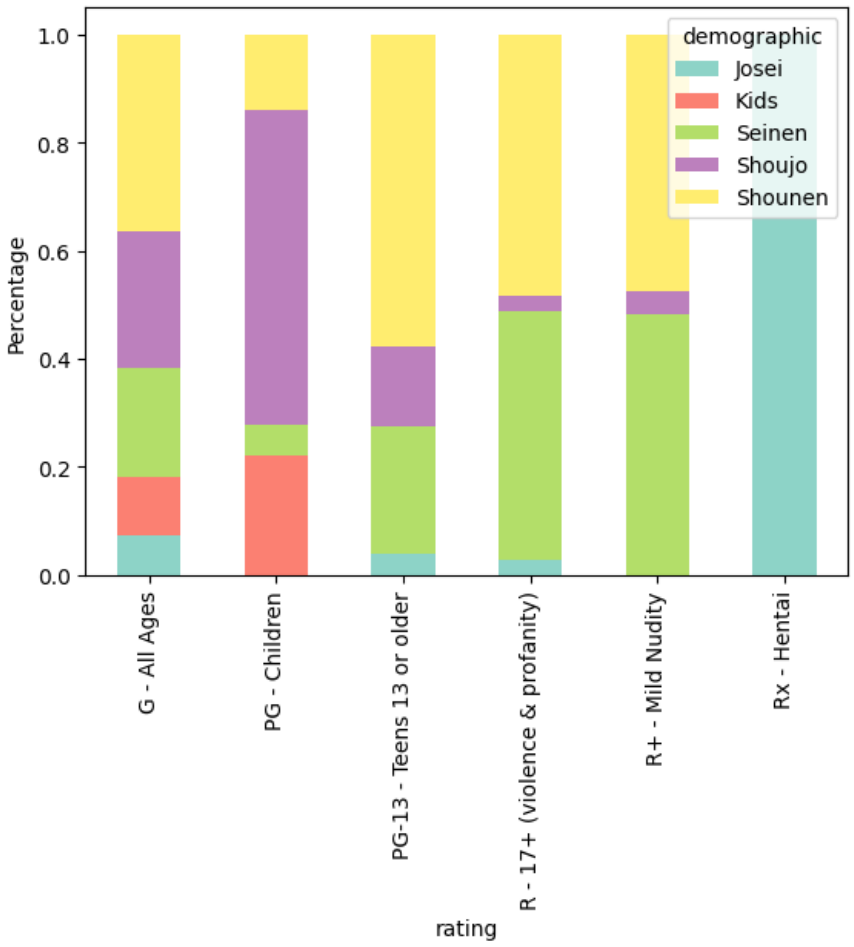

Рисунок 2.1.8 — Соотношение произведений с конкретной демографией к возрастному ограничению

Из Рисунка 2.1.8 можно сделать вывод, что демография Shoujo преобладает в возрастном рейтинге «PG», а Shounen чаще всего получает рейтинг «PG-13». Seinen чаще получает рейтинг «R+» и горазда реже «PG».

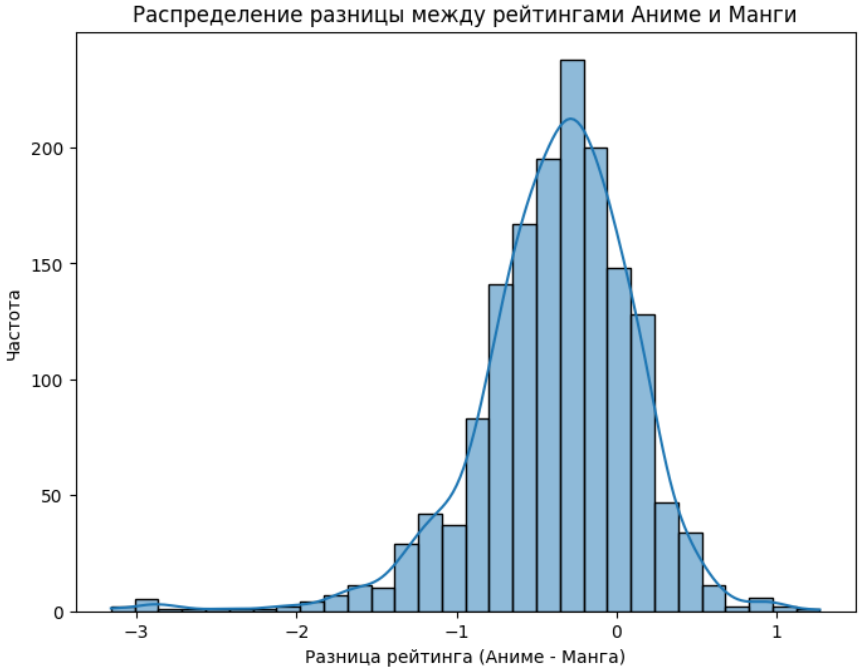

Рисунок 2.1.9 — Распределение разницы между рейтингами аниме и манги

Из этого графика видно, что рейтинг аниме чаще всего ниже рейтинга манги примерно на 0.8. В целом именно рейтинг аниме ниже рейтинга манги, а не наоборот. Поэтому конечная модель прогноза чаще работает в пессимистических предположениях, то есть для манги с рейтингом 8 она будет предсказывать рейтинг аниме 7.5, хотя действительный рейтинг может быть больше, чем 8. Подобное следует расценивать как исключение, нежели тенденцию.

Б. Важность признаков

I. Тепловая карта

Тепловая карта — это графическое представление данных, в котором значения в точках набора данных представлены цветами. Это помогает визуально выделить области с высокой или низкой концентрацией, что делает их анализ интуитивным и понятным.

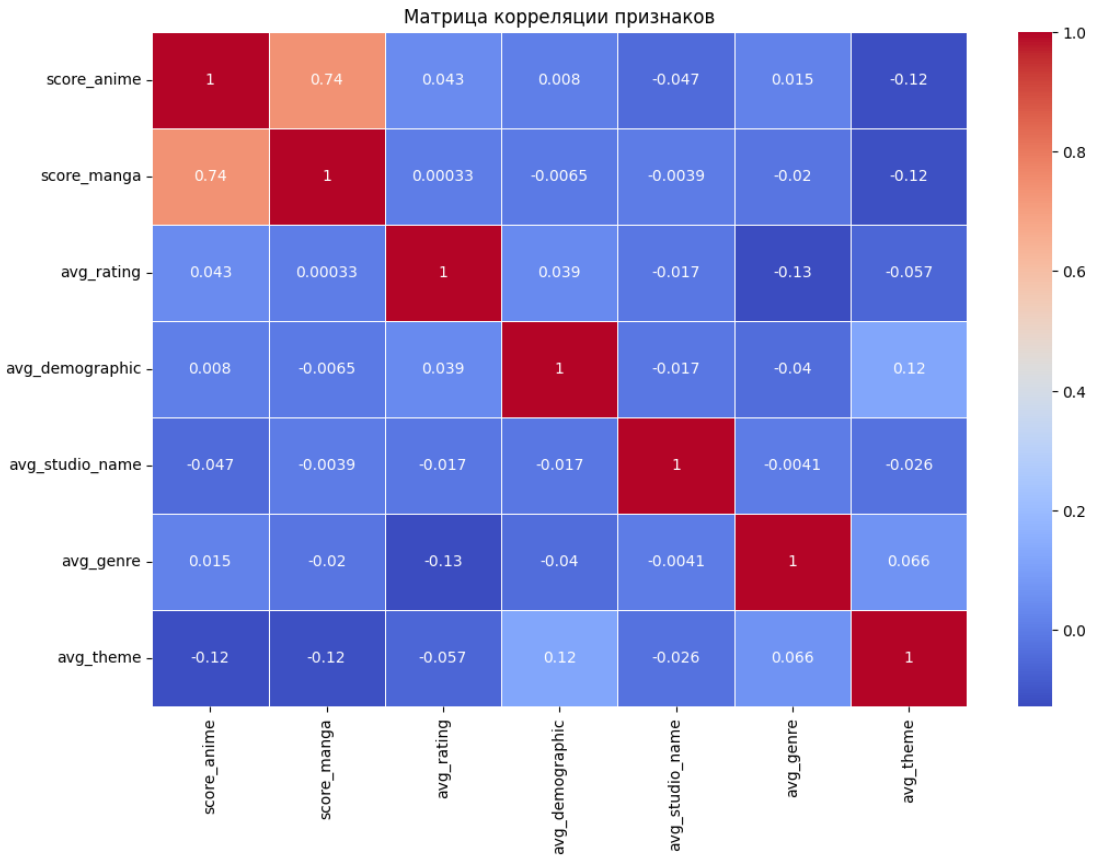

Рисунок 2.2.1 — Матрица корреляции признаков

Из Рисунка 2.2.1 видно, что больше всего на рейтинг аниме влияет рейтинг манги, затем - возрастной рейтинг, жанр и демография.

II. ANOVA

Дисперсионный анализ (ANOVA) — это статистический метод, который используется для сравнения средних значений двух или более выборок. Он позволяет определить, различаются ли средние значения между группами, или же различия случайны.

ANOVA является мощным инструментом, который может использоваться в статистическом анализе для оценки влияния исследуемого фактора на зависимую переменную. Это помогает установить, является ли фактор значимым, и позволяет идентифицировать взаимодействие между переменными.

Существует несколько видов дисперсионного анализа: однофакторный, двухфакторный и многофакторный. В данной работе используется однофакторный анализ – это метод статистического анализа данных, который используется для определения наличия статистически значимых различий между двумя или более группами по одной независимой переменной.

Чем больше значение F-статистики, тем более вероятно, что изменение, связанное с независимой переменной, является реальным, а не случайным, то есть данный параметр имеет существенное влияние на независимую переменную - рейтинг аниме.

Значение p статистики F показывает, насколько вероятно, что значение F, рассчитанное на основе теста, имело бы место, если бы нулевая гипотеза об отсутствии различий между средними группами была верной.

```
ANOVA for studio_name:
F-statistic: 4.5319831827743196, p-value: 9.98598144174669e-57
ANOVA for genre:
F-statistic: 6.401087482773943, p-value: 7.888828889448977e-15
ANOVA for theme:
F-statistic: 5.406216420081224, p-value: 7.077445806785957e-28
ANOVA for rating:
F-statistic: 30.523268002068196, p-value: 1.0945809299726733e-29
ANOVA for demographic:
F-statistic: 7.8957727192111, p-value: 2.6763059593631914e-06
```

Рисунок 2.2.2 — Важность признаков по ANOVA

Из полученных данных, представленных на Рисунке 2.2.2, видно, что наиболее важные категориальные признаки это возрастной рейтинг, демография и жанр.

III. Random Forest

Рисунок 2.2.3 — Важность признаков с помощью Random Forest

Random Forest вычисляет важность признаков на основе уменьшения «загрязненности» (часто измеряемой показателями неопределенности, такими как загрязнения Джини (Gini impurity) или энтропия), которое приносит признак при его использовании для разделения в процессе построения дерева.

IV. XGBoost

Преимущество использования повышения градиента (XGBoost) заключается в том, что после построения усиленных деревьев относительно просто получить оценки важности для каждого атрибута.

Как правило, важность дает оценку, которая показывает, насколько полезной или ценной была каждая функция при построении усиленных деревьев решений в модели. Чем больше атрибут используется для принятия ключевых решений с помощью деревьев решений, тем выше его относительная важность. Эта важность рассчитывается явно для каждого атрибута в наборе данных, что позволяет ранжировать атрибуты и сравнивать их друг с другом.

Важность рассчитывается для одного дерева решений по величине, на которую каждая точка разделения атрибута улучшает показатель производительности, взвешенной по количеству наблюдений, за которые отвечает узел. Затем важности функций усредняются по всем деревьям решений в модели.

```
Важность признаков на основе XGBoost:
           Feature  Importance_XGB
0      score_manga        0.441286
3       avg_rating        0.122929
4        avg_genre        0.120627
5  avg_demographic        0.110748
2        avg_theme        0.109584
1  avg_studio_name        0.094825
```

Рисунок 2.2.4 — Важность признаков с помощью XGBoost

	ANOVA	Random Forest	XGBoost	Correlation		Result
manga score	1	1	1	1		1
genre	4	2	3	5		3
studio name	6	6	6	3		6
rating	2	3	2	4		2
theme	5	4	5	2		4
demographic	3	5	4	6		5

Рисунок 2.2.5 — Сравнительная таблица

Таким образом, можем вывести наиболее значимые параметры, влияющие на итоговый рейтинг аниме, как указано на Рисунке 2.2.5. Цифры в ячейках от 1 до 6 - это значимость параметра по соответствующему методу. В столбце «Result» всем параметрам указана результирующая важность, рассчитанная как среднее арифметическое. Наиболее важные параметры это рейтинг манги, возрастной рейтинг и жанр.

2.3 Сравнение моделей глубокого обучения

Модели сравнивались по качеству прогнозирования параметров из класса манги: дисперсии и коэффициентов линейной регрессии. Оценка качества вычислялась по формуле (1).

Сравнение с нейронными сетями			
	CNN	RNN	MLP (исследуемая модель)
R^2	0.783	0.401	0.999

Рисунок 2.3.1 — Сравнительная таблица эффективности нейронных сетей с предложенной моделью

Сравнение с методами машинного обучения								
	Random Forest	SVM	K-Neighbors	Lasso	Ridge	Gradient Boost	XGBoost	MLP (исследуемая модель)
R^2	0.903	-0.083	0.893	0.997	0.997	0.947	0.905	0.999

Рисунок 2.3.2 — Сравнительная таблица эффективности методов машинного обучения с предложенной моделью

Параметр	Значение
Количество нейронов на входном слое	4
Количество нейронов на выходном слое	1
Количество нейронов на скрытом слое	8
Максимальное количество итераций	10000
Максимальная ошибка (max error)	0.001
Коэффициент скорости обучения (learning rate)	0.08
Тип алгоритма обучения	Back propagation
Функция активации	Sigmoid function

Рисунок 2.3.3 — Параметры для предложенной модели

Модель с параметрами, описанными на Рисунке 2.3.3, показывает хорошие результаты по сравнению с другими нейронными сетями (Рисунок 2.3.1) и методами машинного обучения (Рисунок 2.3.2).

MLP показали возможность находить приближенные решения для чрезвычайно сложных задач. В частности, они являются универсальным аппроксиматором функций, поэтому с успехом используются в построении регрессионных моделей. Была выбрана именно эта архитектура, так как она простая в реализации, легко интерпретируемая и, самое важное, показывает стабильно хорошие результаты за небольшое время выполнения. Рассмотрим исследуемую модель более подробно.

В предложенной модели используется один скрытый слой. Проблемы, требующие более одного скрытого слоя, встречаются редко. Для многих практических задач нет смысла использовать более одного скрытого слоя. Один слой может аппроксимировать любую функцию, содержащую непрерывное отображение одного конечного пространства в другое. Решение о количестве скрытых слоев нейронов — это лишь малая часть проблемы. Мы также должны определить, сколько нейронов будет в каждом из скрытых слоев. Необходимо тщательно учитывать как количество скрытых слоев, так и количество нейронов в каждом из этих скрытых слоев.

Использование слишком малого количества нейронов в скрытых слоях приведет к недообучению. Недообучение происходит, когда в скрытых слоях слишком мало нейронов для адекватного обнаружения зависимостей в сложном наборе данных.

Напротив, использование слишком большого количества нейронов в скрытых слоях может привести к ряду проблем. Во-первых, слишком много нейронов в скрытых слоях может привести к переобучению. Переобучение происходит, когда нейронная сеть имеет настолько большую мощность обработки информации, что ограниченного количества информации,

содержащейся в обучающем наборе, недостаточно для обучения всех нейронов в скрытых слоях. Вторая проблема может возникнуть, даже если обучающих данных достаточно. Непомерно большое количество нейронов в скрытых слоях может увеличить время обучения сети. Количество времени обучения может увеличиться до такой степени, что нейронную сеть невозможно будет адекватно обучить.

Очевидно, что необходимо достичь некоего компромисса между слишком большим и слишком малым количеством нейронов в скрытых слоях. Поэтому после ряда исследований было выбрано 8 нейронов в скрытом слое.

В качестве алгоритма обучения используется обратное распространение ошибки (backpropagation). Он позволяет определить, как изменять веса связей между нейронами на каждом слое сети, чтобы минимизировать ошибку предсказаний.

Принцип работы алгоритма основан на методе градиентного спуска. В начале обучения пропускается входной вектор через нейронную сеть, и полученный выход сравнивается с ожидаемым выходом. Ошибка между этими значениями вычисляется и используется для определения градиентов функции потерь по весам и смещениям сети.

В качестве функции активации используется сигмоидальная функция. Она преобразует входное значение в диапазоне от отрицательной бесконечности до положительной бесконечности в значение от 0 до 1. Эта функция подходит для задачи прогнозирования рейтинга, так как она никогда не может быть отрицательной и не может быть больше десяти. В контексте данной задаче полученные данные рейтингов, например 7.5 и 8.2, делятся на 10 до подачи в нейронную сеть. То есть сеть работает с числами вида 0.75 и 0.82. В результате работы модели получаем число не больше единицы и умножаем его на десять, чтобы получить предсказанное значение рейтинга.

Глава 3. Реализация

3.1 Математическая модель

Для прогноза рейтинга аниме будут использоваться следующие понятия: метод наименьших квадратов, дисперсию и нейронную сеть MLP.

На рисунках ниже представлены этапы работы модели в двух идейно различающихся классах: для обработки данных у литературного первоисточника и для данных от других параметров, таких как жанр, студия, тема и так далее.

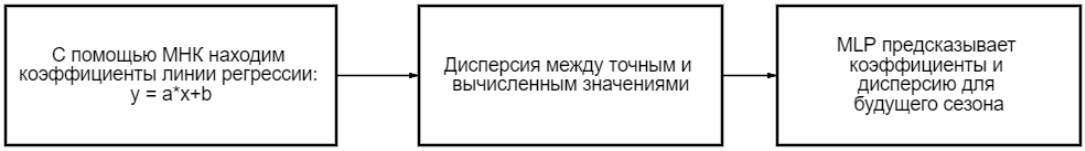

Рисунок 3.1.1 — Этапы предсказания для литературного первоисточника

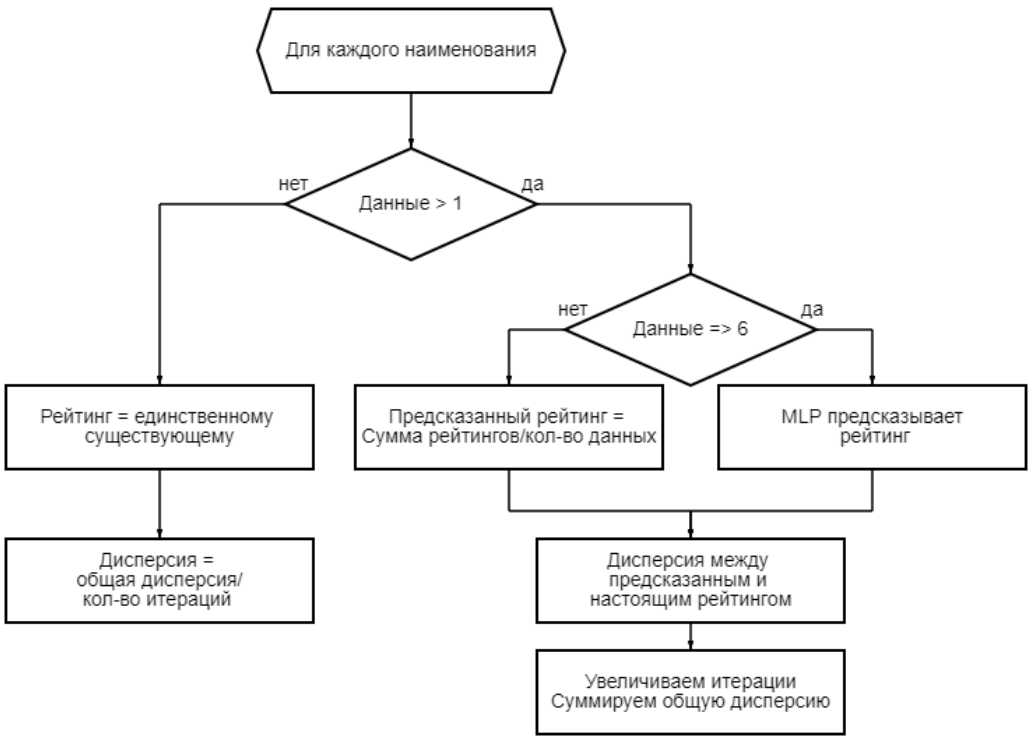

Рисунок 3.1.2 — Этапы предсказания для других параметров

На Рисунке 3.1.2 представлены основные этапы прогнозирования для таких параметров, как жанр, студия, тема, возрастное ограничение и демография. Внутри каждого параметра есть свои наименовании, например, в параметре жанр будут такие наименования как комедия, ужасы, драма и так далее. У каждого наименования есть свой набор рейтингов, предположим, у жанра комедии 6 рейтингов, то есть известных данных. В таком случае, ожидаемый рейтинг для этого жанра в следующем сезоне предсказывается с помощью нейронной сети. Теперь предположим, что у жанра ужасы всего 4 известных рейтинга, в таком случае ожидаемый рейтинг для будущего сезона находится как сумма известных 4 рейтингов, деленная на размер набора, то есть на 4. И пусть в жанре драма есть только один известный рейтинг, тогда ожидаемый будет совпадать с ним.

Самый важный параметр для конечного прогноза является рейтинг манги, этапы получения которого описаны на Рисунке 3.1.1. Поэтому подробнее остановимся на том, как именно получается этот рейтинг.

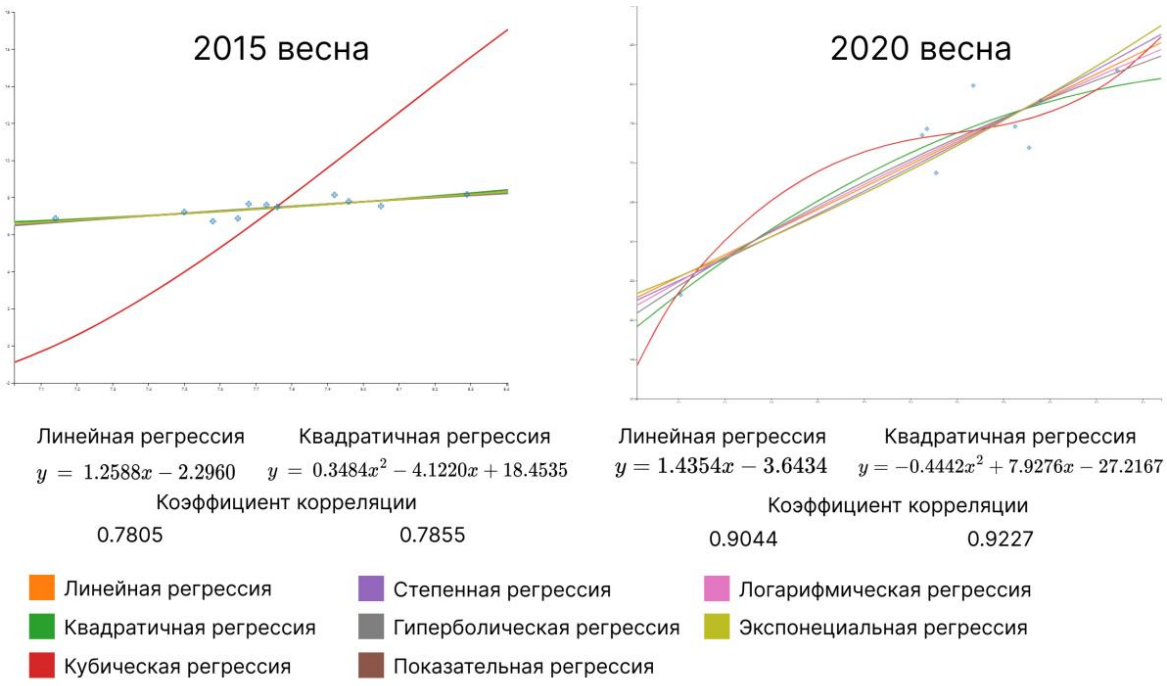

Рисунок 3.1.3 — Виды аппроксимирующих функций

На Рисунке 3.1.3 изображены различные аппроксимирующие функции, построенные по исследуемому набору данных. Нужно подобрать такую аппроксимирующую функцию $y = f(x)$, график которой проходит как можно ближе к исходным точкам. Но таких классов функций слишком много, поэтому будем искать достаточно простую и с приемлемой дисперсией. Будем это делать с помощью метода наименьших квадратов $\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - f(x_i))^2$ (МНК): ищем такую функцию, чтобы сумма квадратов отклонений была как можно меньше. Проанализировав графики по исходным данным на Рисунке 3.1.3, было выявлено, что точки имеют тенденцию располагаться на прямой, то есть на линии регрессии, – $y = f(x) = ax + b$. Теперь в предположении, что рейтинги аниме и манги линейно зависят, мы рассмотрим линейную регрессию, которая описана выше, где x - рейтинг манги, y - рейтинг аниме. Коэффициенты a и b, называемые также параметрами модели, определяются таким образом, чтобы сумма квадратов отклонений точек, соответствующих реальным наблюдениям данных, от линии регрессии была бы минимальной. Коэффициенты обычно оцениваются методом наименьших квадратов. Теперь стоит задача найти a и b, чтобы сумма квадратов отклонений, то есть $\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - (ax_i + b))^2$ была наименьшей.

Задача свелась к нахождению минимума функции двух переменных.

$$\frac{\partial F}{\partial a} = \left(\sum_{i=1}^{n} \left(y_i - (ax_i + b)\right)^2\right)'_a = \sum_{i=1}^{n} \left[2\left(y_i - (ax_i + b)\right) * \left(y_i - (ax_i + b)\right)'_a\right]$$

$$= 2\sum_{i=1}^{n} \left[(y_i - ax_i - b) * \left(0 - (x_i + 0)\right)\right]$$

$$= 2\sum_{i=1}^{n} \left[(y_i - ax_i - b) * (-x_i)\right] = 2\sum_{i=1}^{n} \left(ax_i^2 + bx_i - x_i y_i\right)$$

$$\frac{\partial F}{\partial b} = (\sum_{i=1}^{n}\big(y_i - (ax_i + b)\big)^2)`_b = \sum_{i=1}^{n}\Big[2\big(y_i - (ax_i + b)\big) * \big(y_i - (ax_i + b)\big)`_b\Big]$$

$$= 2\sum_{i=1}^{n}\big[(y_i - ax_i - b) * \big(0 - (0 + 1)\big)\big] = 2\sum_{i=1}^{n}(ax_i + b - y_i)$$

Составим стандартную систему:

$$\begin{cases}\frac{\partial F}{\partial a} = 0\\ \frac{\partial F}{\partial b} = 0\end{cases} \rightarrow \begin{cases}2\sum_{i=1}^{n}\big(ax_i^2 + bx_i - x_i y_i\big) = 0\\ 2\sum_{i=1}^{n}(ax_i + b - y_i) = 0\end{cases}$$

Сокращаем каждое уравнение на 2 и, кроме того, расписываем суммы:

$$\begin{cases}a\sum_{i=1}^{n}x_i^2 + b\sum_{i=1}^{n}x_i - \sum_{i=1}^{n}x_i y_i = 0\\ a\sum_{i=1}^{n}x_i + \sum_{i=1}^{n}b - \sum_{i=1}^{n}y_i = 0\end{cases} \rightarrow \begin{cases}a\sum_{i=1}^{n}x_i^2 + b\sum_{i=1}^{n}x_i = \sum_{i=1}^{n}x_i y_i\\ a\sum_{i=1}^{n}x_i + bn = \sum_{i=1}^{n}y_i\end{cases}$$

Координаты исходных данных (x_i , y_i) известны, поэтому можно найти неизвестные a и b. Составляем систему двух линейных уравнений с двумя неизвестными, решаем и получаем искомое. Таким образом, функция $y = f(x) = a^*x + b^*$ наилучшим образом приближает исследуемые данные.

Для вычисления веса параметров в прогнозе будем применять дисперсию: $D = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$ (3)

Для проверки достоверности модели был вычислен коэффициент корреляции: $r_{xy} = \frac{\dot{x}\dot{y} - \dot{x} * \dot{y}}{\sigma_x \sigma_y}$ (4)

где $\dot{x}\dot{y}$ - среднее значение произведения признаков,

$\dot{x}$, $\dot{y}$ - средние значения признаков,

σ_x, σ_y - стандартные отклонения признаков и находятся по формуле (5):

$$\sigma_x = \sqrt{\frac{\sum x_i^2}{n} - \dot{x}^2}, \sigma_y = \sqrt{\frac{\sum y_i^2}{n} - \dot{y}^2} \quad (5)$$

Коэффициент корреляции равен 0.79, что по шкале Чедокка означает сильную корреляционную зависимость между исследуемыми параметрами, то есть между рейтингом манги и рейтингом аниме. Этот коэффициент используется для измерения силы и направления линейной связи между двумя переменными, причем он устойчив к выбросам.

Все предсказанные параметры и дисперсии нужны для конечной формулы прогноза. Конечная формула выглядит как сумма значений параметров жанр, студия и манга, а также дополнительных параметров: темы, возрастного рейтинга и демографии, поделенная на сумму общего веса.

Рассмотрим формулы для каждого параметра более подробно.

$$Manga = (score * a + b) * |1 - D_{manga}|,$$

где a и b предсказываются нейронной сетью, на основании предыдущих значений коэффициентов в аппроксимирующей линейной функции; score - введенный рейтинг манги, D_{manga} - дисперсия манги, вычисленная по формуле (3).

$$Studio = score * |1 - D_{studio}|,$$

где score - заранее известное значение, вычисленное с помощью нейронной сети; D_{studio} - дисперсия студии из формулы (3).

$$Genre = score * |1 - D_{genre}|,$$

где score - заранее известное значение, вычисленное с помощью нейронной сети; D_{genre} - дисперсия студии, вычисленная по формуле (3). При этом указывать можно более одного жанра, если такие есть.

Дополнительные параметры, которые не обязательно указывать в конечной формуле, но желательно, так как они улучшают качество прогноза, имеют ту же структуру, что Studio и Genre:

$$Theme = score * |1 - D_{theme}|$$

$$Rating = score * |1 - D_{rating}|$$

$$Demographic = score * |1 - D_{demographic}|$$

Результирующая формула без учета дополнительных параметров:

$$Res = \frac{(Manga+Studio+\sum Genre)}{|1-D_{manga}|+|1-D_{studio}|+\sum|1-D_{genre}|} \quad (6)$$

С учетом дополнительных параметров:

$$\frac{(Manga + Studio + \sum Genre + Theme + Rating + Demographic)}{|1 - D_{manga}| + |1 - D_{studio}| + \sum|1 - D_{genre}| + |1 - D_{theme}| + |1 - D_{rating}| + |1 - D_{demographic}|}$$

В предложенной модели также утверждается, что если у какого-либо параметра дисперсия равна нулю или более 1, то вес этого параметра будем считать за 0.

Формула (6) и ее расширенная версия стабильно показывают точность более 50%, но для лучших показателей ее можно усовершенствовать. А именно, в зависимости от рейтинга манги можно умножать посчитанный параметр Manga на константу. Константы подбираются эмпирически. Границы интервалов рейтингов манги также можно настраивать, но это возможно только после аналитики по крайней мере 4 сезонов, которая позволит выявить некоторые закономерности. Рекомендуется делить интервалы по крайней мере на три класса: меньше 7.0, до 8.0 и от 8.0. Коэффициенты и разбиение интервалов, использующиеся в текущей версии работы представлены на Рисунке 3.1.4.

Интервал	Константа
[0, 6.4)	0.57
[6.4, 6.85)	0.725
[6.85, 7.15)	0.775
[7.15, 7.5)	0.8825
[7.75, 7.85)	1.6
[7.85, 7.95)	1.225
[8.40, 8.55)	1.285
[8.57, 10]	1.3

Рисунок 3.1.4 — Коэффициенты для формулы (6)

2.2 Приложения для прогнозирования рейтинга

Для реализации архитектуры MLP используется библиотека Neuroph, разработанная специально для работы с нейронными сетями [29], базовые концепты которой представлены на Рисунке 3.2.1.

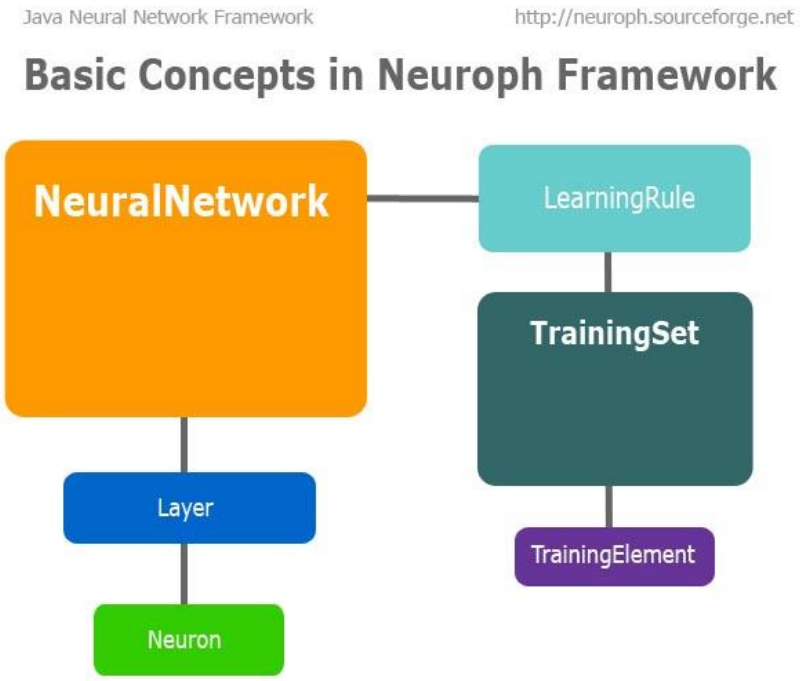

Рисунок 3.2.1 — Базовый концепт нейронной сети

Как упоминалось в параграфе 1 главы 2, для получения json-объектов с внешнего ресурса использовался пакет «java.net.http», а для обработки получившихся объектов использовался пакет «org.json.simple» [30].

Основной функционал приложения можно разделить на три класса: создание датасета, оценка качества модели, а также формирование и отправление json-объекта, содержащего спрогнозированный значения.

Для создания датасета используется функция createDataset(), которая с помощью внешних функций создает набор csv файлов за установленный промежуток времени. Файлы имеют названия вида «год»+«сезон», например, «2020winter.csv». Основные внешние функции реализованы в классе GetInfo.

Рассмотрим каждую из этих функций в классе GetInfo.

1. Функция getAnimeListbySeason(int year, String season, int page) по запросу к внешнему ресурсу возвращает ответ, в котором содержится json-объект.
2. Функция parseAnime(String generaljson, int year, String season) обрабатывает json-объект, извлекает из него необходимые параметры, создает и записывает полученную информацию в csv файл.
3. getMangaIdbyAnimeId(Long id) позволяет получать все связанные произведения с этим аниме, например продолжения, альтернативные версии истории, адаптации и так далее.
4. Функция parseRelation (String generaljson) достает из общего списка связанных произведений только адаптацию и возвращает ее id.
5. С помощью getMangabyId(String id) получаем json, содержащий всю информацию о первоисточнике.
6. Из parseManga (String generaljson) получаем рейтинг манги, которую адаптировали.

Сравнение результатов модели, а именно вычисление RMSE по формуле (2), по которому можно оценить качество предсказания, осуществляется в функции check_rmse(int target_year, int target_season). Для этого используются известные данные, а именно - берется уже существующий файл из датасета. Например, «2022winter.csv» и для него вычисляется предсказание, в предположении данных известных до осень 2021 включительно. Для этого используется функционал класса FinalFormula:

1. Функция create_json(int start, int end) формирует четыре json, по одному на каждый сезон, содержащий спрогнозированные данные до текущего момента. Например, если start совпадает с end и равен 2021, то получим: прогноз до весны 2021, то есть модель обучалась на датасете до зимы 2020 включительно; прогноз до лета 2021, то есть обучение было до весны 2021 и так далее. Для вычисления прогнозов используются классы ProcessingManga, ProcessingGenre, ProcessingStudio, ProcessingTheme, ProcessingRating и ProcessingDemographic. Более подробно о них будет ниже.

2. Функция formul(double manga_score, String tg_genre, String tg_studio, String tg_theme, String tg_rating, String tg_demographic, int year, int season) получает на вход известные параметры из csv таблиц. Из сформированных функцией описанной выше json-объектов достаются параметры, которые надо поставить в формулу (6) или в её расширенную версию. Все данные подставляются в эту формулу и получаем предсказание.

Разберем основную часть программного обеспечения - вычисление предсказаний. Этапы предсказаний описаны на Рисунках 3.1.1 и 3.1.2.

Рассмотрим класс ProcessingManga, описанного на Рисунке 3.1.1, в котором обрабатываются все данные, связанные с первоисточником.

1. Из getMangaData(int year, String season) получаем рейтинги аниме и манги из соответствующего по году и сезону csv файла.

2. Функция relationScores(int year_end, int season, boolean final_forecasting) записывает в csv файл предсказанные коэффициенты a и b из линии регрессии, а также найденную дисперсию. Параметр final_forecasting - создает json, использующийся для проверки модели, если он равен false, или создает json, который передается на сервер и данные из которого используются в конечной формуле.

3. trainNeuralNetwork(int k, ArrayList<MangaCoeff> finalC) - предсказание с помощью нейронной сети, где k отвечает за то, какой именно параметр будет предсказан. k=0 - коэффициент a, k=1 - коэффициент b и k=2 - дисперсия. finalC это объект собственного класса MangaCoeff с такими полями как сезон, коэффициент a, коэффициент b и дисперсия.

4. Функция squareDispersion(ArrayList<Double>xA, ArrayList<Double> yA, double a, double b) реализует дисперсию, указанную в формуле (3). xA - вектор, содержащий рейтинги манги, yA - вектор, содержащий рейтинги аниме.

5. Функция least_square(ArrayList<Double> xArr, ArrayList<Double> yArr) реализует метод наименьших квадратов, который подробно описан в предыдущем параграфе. Также внутри определено вычисление коэффициента корреляции по формуле (4).

6. determinant(double[] xArr, double[] yArr) - нахождение определителя, используемого в МНК, то есть в функции least_square.

Классы ProcessingGenre, ProcessingStudio, ProcessingTheme, ProcessingRating и ProcessingDemographic имеют одинаковын структуры и методы, отличающиеся только названиями и входными данными. Рассмотрим методы на конкретном примере ProcessingGenre.

1. Создается собственный класс GenreInfo с такими полями, как рейтинг, название жанра и сезон.

2. В getGenreData(int year, String season) заполняется класс GenreInfo, где из csv файлов достаются рейтинг аниме и его жанр.

3. Метод call(int year_end, int season, boolean final_forecasting) вызывает функции, которые вычисляют и прогнозируют значения, а затем записываются в json-объект. Параметр final_forecasting - создает json, использующийся для проверки модели, если равен false, или иначе json, который передается на сервер и данные из которого используются в конечной формуле.

4. trainNeuralNetwork(ArrayList<Double> genrebyScore) - предсказывает рейтинг конкретного жанра, например комедии, на основе рейтингов известных аниме в этом жанре.

5. Уже известная функция squareDispersion(double a, ArrayList<Double> xArr) вычисляет дисперсию по формуле (3) между спрогнозированным рейтингом и предыдущими рейтингами.

Отдельно стоит упомянуть функцию upload_file(), которая отправляет сформированный json с предсказаниями на общедоступный сервер. Сама функция находится в классе Main.

Таким образом, мы рассмотрели весь функционал приложения и узнали, каким образом совершаются предсказания. Приложение обладает не только функционалом, относящимся конкретно к прогнозированию, но и обладает возможностью создавать собственные наборы данных, а также позволяет быстро проверить точность модели.

2.3 Реализация Android приложения

Для удобства пользования программы для прогнозирования было принято решение реализовать приложение на базе ОС Android. Основными факторами, повлиявшими на данное решение, являлась высокая доступность средств разработки и использование того же языка программирования, на котором написан основной функционал. В качестве среды разработки была использована Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, предлагающая встроенный SDK (Software Development Kit – комплект разработки программного обеспечения). Минимальная версия android, для работы приложение - API Level: 26, Android Oreo от 2017 года [31].

В реализации приложения были использованы такие библиотеки, как «okhttp3» [32], который предоставляет простой, легкий в использовании API для выполнения HTTP-запросов, и «org.json.simple» для обработки получившихся ответов.

Данное приложение построено на использование фрагментов, а не activity. Фрагмент представляет кусочек визуального интерфейса приложения, который может использоваться повторно и многократно. Фрагмент существует в контексте activity и имеет свой жизненный цикл, вне activity обособленно он существовать не может. Каждая activity может иметь несколько фрагментов. В данном приложении действует только одно activity - Main.

Смысл приложения заключается в сборе и обработке входных параметров. Затем они подставляются в формулу (6) и в результате мы получаем ожидаемый рейтинг аниме по входным параметрам.

Рассмотрим основные фрагменты приложения:

1. PredictFrag нужен для ввода и запоминания рейтинга манги, а также для вызова класса ReadData, который получает json с сервера, парсит его и обновляет данные во всех фрагментах.

2. GenreFrag в паре с адаптером GenreAdapter позволяет найти по поиску и выбрать один или несколько жанров, одновременно запоминая название, дисперсию и рейтинг выбранных параметров.

3. StudioFrag вместе с адаптером StudioAdapter дает возможность найти по поиску студию и выбрать её, одновременно запоминая название, дисперсию и её рейтинг.

4. AdditionalParametrs нужен для выбора дополнительных параметров: темы, возрастного ограничения и демографии. Можно ничего не выбирать, или выбрать любую комбинацию этих параметров. Дисперсия и рейтинг для выбранных параметров запоминаются.

5. ResultFrag используется для конечного расчета формулы и её вывода.

Стоит отметить еще несколько классов: собственные классы StufioOrGenre и AddParam, содержащий такие поля, как название, дисперсия и рейтинг, и класс ReadData реализующий обновление данных.

Рассмотрим функционал приложения и точность предсказания на конкретном примере. Входные данные (на момент обращения 23.04.2024) [33] для аниме с id 55866: студия - Ajia-do, жанр - Romance, рейтинг манги - 8.49, тема - Adult Cast, возрастное ограничение - PG-13 - Teens 13 or older, демография - Shoujo.

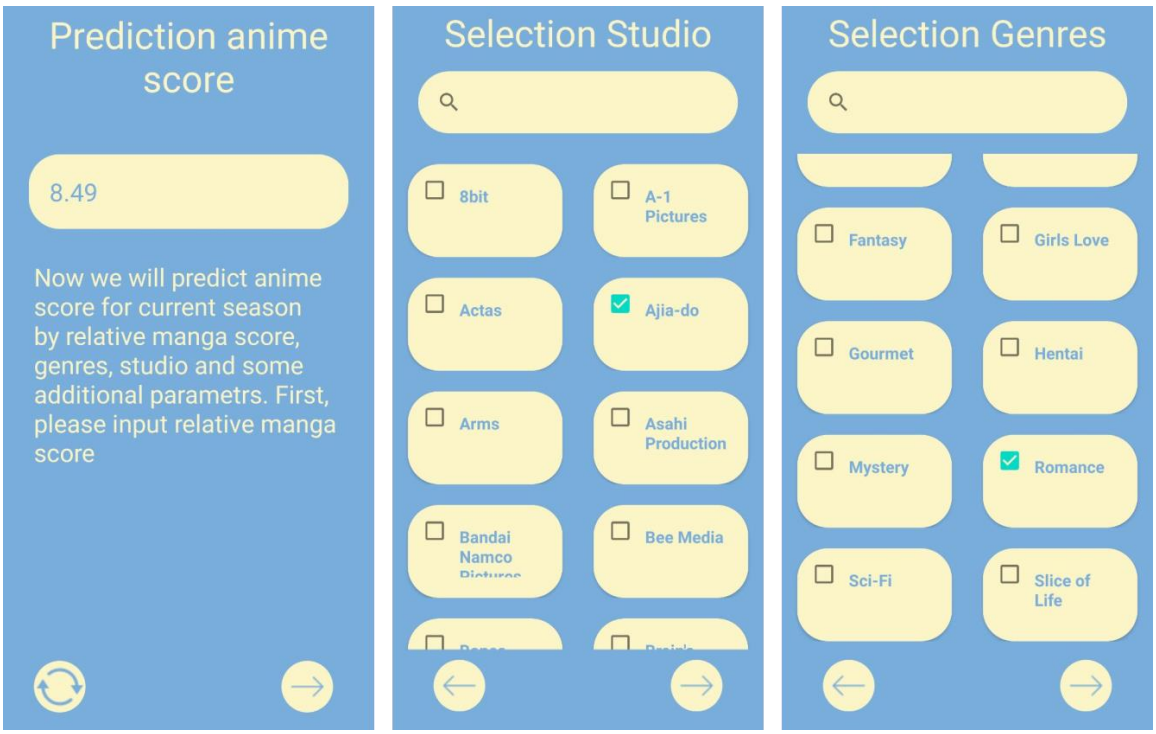

Рисунок 3.2.1 — Ввод основных параметров. Фрагменты слева направо: PredictFrag, GenreFrag, StudioFrag

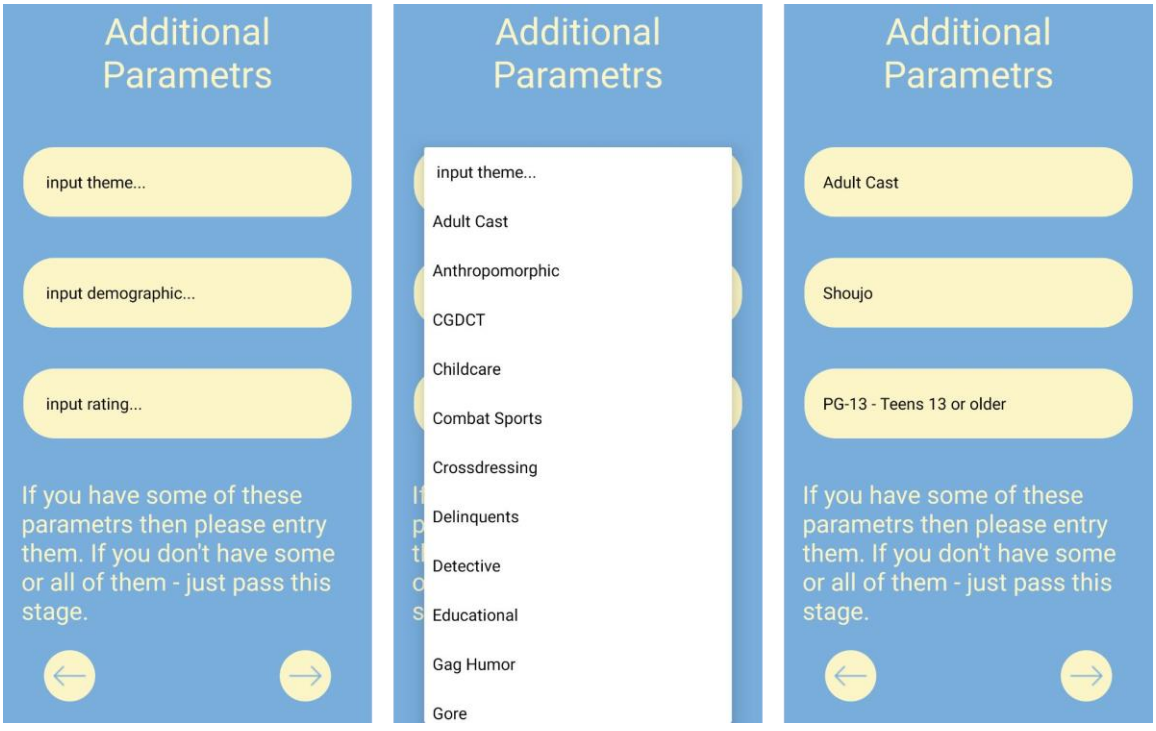

Рисунок 3.2.2 — Ввод дополнительных параметров. Фрагмент AdditionalParametrs

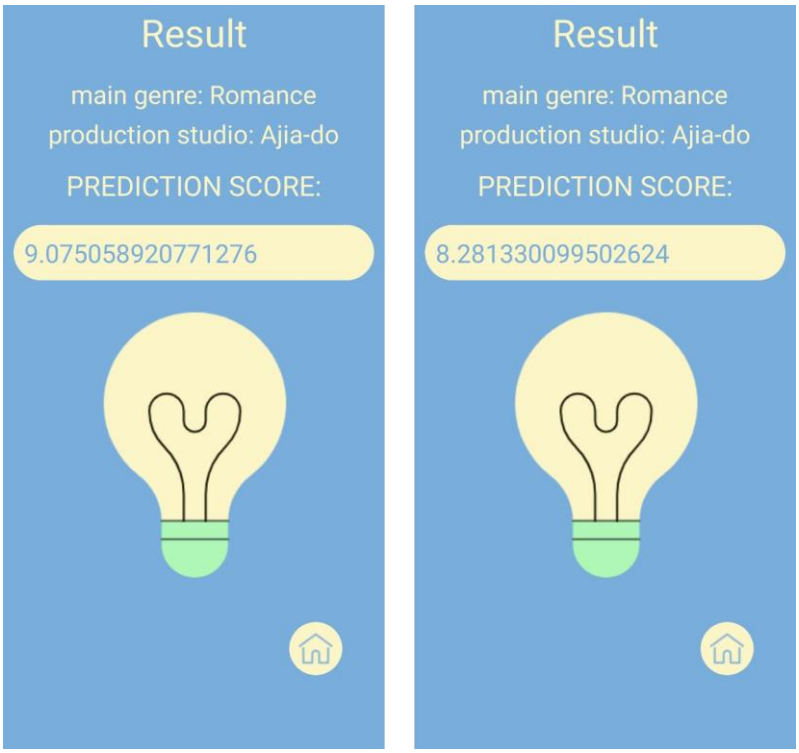

Рисунок 3.2.3 — Результаты - слева без указания дополнительных параметров, справа – с дополнительными параметрами. Фрагмент ResultFrag

Текущий рейтинг рассматриваемого произведения 8.27. Спрогнозированное значение без учета дополнительных параметров – 9.07, а с учетом 8.29. Рассмотрим еще один пример, аниме с id 53421 [34] и такими входными параметрами: студия - SILVER LINK, жанр - Comedy, рейтинг манги - 7.1, тема - Romantic Subtext, возрастное ограничение - PG-13 - Teens 13 or older, демография - Shounen, получился предсказанный рейтинг 6.68 без учета дополнительных параметров и 6.77 с их учетом, тогда как реальный рейтинг на момент обращения 23.03.2024 равен 7.03. На большой выборке становится ясно, что для более точного прогноза стоит учитывать дополнительные параметры.

Таким образом, приложение состоит из 5 фрагментов: ввод рейтинга манги, выбор жанра или несколько жанров, выбор студии, ввод дополнительных параметров и вывод полученного прогноза. Всё приложение построено интуитивным для пользователя и сопровождается уточняющими надписями и подсказками.

Выводы

В результате проделанной работы было создано приложение на языке программирования Java, которое создает наборы данных и реализует математическую модель для предсказания рейтингов.

Также было создано мобильное приложение на базе ОС Android, осуществляющее взаимодействие с пользователем и вывод результирующего прогноза рейтинга.

В ходе работы были исследованы различные подходы глубокого обучения для предсказания параметров и проанализированы ряд статей. В результате сравнения была выбрана архитектура MLP, которая удовлетворяет не только в точности, но и в простоте реализации и времени обучения.

Было проведено исследование данных для выявлений закономерностей и зависимостей между параметрами. В результате были выявлены наиболее значимые параметры, а также достигнуты некоторые представления о влияние параметров друг на друга.

Был разработан и реализован алгоритм, позволяющий получать данные о произведениях за настраиваемые интервалы времени. Также была реализована математическая модель, которая предсказывает значения параметров с помощью архитектуры MLP, метода наименьших квадратов и дисперсии. Android приложение выступает в качестве графического интерфейса для отображения результирующих данных пользователю.

RMSE модели на выборке за 2023 год составляет 0.3. Для спрогнозированных рейтингов за зиму 2024 RMSE = 0.4. Тонкая настройка констант и интервалов в формуле (6) может сильно снизить RMSE, рекомендуется их настраивать хотя бы раз в сезон. RMSE математической

модели без учета констант и разделения на интервалы все еще ниже 0.5, если рассматривать выборки за год.

Ознакомиться с кодом для программного обеспечения, android приложением и набором данных можно на Github [35].

Заключение

В данной выпускной квалификационной работе были достигнуты следующие результаты:

• Сделан обзор предметной области

• Сделан обзор и сравнение различных подходов глубокого обучения

• Разработана программа для создания набора данных

• Проведен анализ данных и важности признаков

• Разработана и реализована математическая модель

• Разработано android приложение для прогнозирования рейтинга по введенным пользователем параметрам

В математической модели использовалась нейронная сеть и методы математической статистики. Модель показывает хорошие результаты. Предложенная формула для прогноза может быть расширена на большее количество параметров. В функции, отвечающей за создание набора данных, также можно настраивать временные интервалы для сбора данных и указывать виды параметров. Реализовано интуитивное понятное android приложение, с помощью которого можно легко узнать ожидаемый рейтинг произведения по введенным данным.

Список литературы

1. Report 2022 Summary // URL: https://aja.gr.jp/english/japan-anime-data

2. Carter L. Marketing anime to a global audience: A paratextual analysis of promotional materials from Spirited Away //East Asian Journal of Popular Culture. – 2018. – Т. 4. – №. 1. – С. 47-59

3. Mihara R. A Coming of Age in the Anthropological Study of Anime?: Introductory Thoughts Envisioning the Business Anthropology of Japanese Animation //Journal of Business Anthropology. – 2020. – Т. 9. – №. 1. – С. 88-110

4. AlSulaim S. M., Qamar A. M. Prediction of Anime Series' Success using Sentiment Analysis and Deep Learning //2021 International Conference of Women in Data Science at Taif University (WiDSTaif). – IEEE, 2021. – С. 1-6

5. Setiawan N. et al. Time Series Model to Predict Future Popular Animes Genres in 2025 //E3S Web of Conferences. – EDP Sciences, 2023. – Т. 388. – С. 02002

6. Armenta-Segura J., Sidorov G. Anime Success Prediction Based on Synopsis Using Traditional Classifiers //Proceedings of Congreso Mexicano de Inteligencia Artificial, COMIA. – 2023

7. Fu X. Multiple Linear Regression Analysis of the Animation Score //Highlights in Science, Engineering and Technology. – 2023. – Т. 49. – С. 183-188

8. Manav Agarwal, Shreya Venugopal, Rishab Kashyap. Movie Success Prediction and Performance Comparison using Various Statistical Approaches / // International Journal of Artificial Intelligence and Applications (IJAIA), Vol.13, No.1, January 2022

9. Gandasari R. A. et al. Predicting Over the Top Services Movies and Shows Success Using Machine Learning //2023 International Conference on Information Management and Technology (ICIMTech). – IEEE, 2023. – С. 89-94

10. e Souza T. L. D., Nishijima M., Pires R. Revisiting predictions of movie economic success: random Forest applied to profits //Multimedia tools and applications. – 2023. – Т. 82. – №. 25. – С. 38397-38420

11. Зимин И. В., Мироненко А. О., Агаев Ш. САМОАДАПТИРУЕМАЯ НЕЙРОСЕТЕВАЯ СИСТЕМА ОЦЕНКИ И ПРОГНОЗИРОВАНИЯ МИРОВЫХ КАССОВЫХ СБОРОВ ФИЛЬМА //Интеллектуальные системы в науке и технике. Искусственный интеллект в решении актуальных социальных и экономических проблем XXI века. – 2020. – С. 569-581

12. Поселенцева Д. Ю., Миков Р. О. НЕЙРОСЕТЕВОЕ ПРОГНОЗИРОВАНИЕ МИРОВОГО РЕЙТИНГА ФИЛЬМА И ЗАВИСИМОСТЬ ЭТОГО РЕЙТИНГА ОТ ПОПУЛЯРНОСТИ КИНОЛЕНТЫ В СОЦИАЛЬНЫХ СЕТЯХ //Интеллектуальные системы в науке и технике. Искусственный интеллект в решении актуальных социальных и экономических проблем XXI века. – 2020. – С. 582-587

13. Yu H., Fu M. DbRMP: Predicting Douban Rating of Movies with high-dimensional Features by Comprehensive Machine Learning Algorithms //2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). – IEEE, 2022. – С. 540-544

14. Abidi S. M. R. et al. Popularity prediction of movies: from statistical modeling to machine learning techniques //Multimedia Tools and Applications. – 2020. – Т. 79. – С. 35583-35617

15. Qin M. et al. MAMRP: Multi-modal Data Aware Movie Rating Prediction //International Conference on Advanced Data Mining and Applications. – Cham : Springer Nature Switzerland, 2023. – С. 660-675

16. N. Cristianini and E. Ricci. Support Vector Machines // MA: Springer US, pp. 928–932, 2008

17. Breiman L. Random forests //Machine learning. – 2001– T. 45. – C.5-32

18. Natekin A., Knoll A. Gradient boosting machines, a tutorial //Frontiers in neurorobotics. – 2013. – T. 7. – C. 21

19. Chen T., Guestrin C. Xgboost: A scalable tree boosting system //Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. – 2016. – C. 785-794

20. Zhou Y., Zhang L., Yi Z. Predicting movie box-office revenues using deep neural networks //Neural Computing and Applications. – 2019. – T. 31. – C. 1855-1865

21. Abarja R. A., Wibowo A. Movie rating prediction using convolutional neural network based on historical values //Int. J. – 2020. – T. 8. – C. 2156-2164

22. Chen Y., Guo L., Zhang C. Score prediction model based on neural network //Optical Memory and Neural Networks. – 2020. – T. 29. – C. 37-43

23. Darapaneni N. et al. Movie success prediction using ml //2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). – IEEE, 2020. – C. 0869-0874

24. Zhang L., Luo J., Yang S. Forecasting box office revenue of movies with BP neural network //Expert Systems with Applications. – 2009. – T. 36. – №. 3. – C. 6580-6587

25. Sharda R., Delen D. Predicting box-office success of motion pictures with neural networks //Expert Systems with Applications. – 2006. – T. 30. – №. 2. – C. 243-254.

26. Quader N. et al. A machine learning approach to predict movie box-office success //2017 20th International Conference of Computer and Information Technology (ICCIT). – IEEE, 2017. – С. 1-7.

27. jikan.moe // URL: https://jikan.moe/ (дата обращения: 23.03.24)

28. Package java.net.http // URL: https://docs.oracle.com/en%2Fjava%2Fjavase%2F11%2Fdocs%2Fapi%2F%2Fjava.net.http/java/net/http/package-summary.html

29. Neuroph // URL: https://neuroph.sourceforge.net/

30. Package json.simple // URL: https://code.google.com/archive/p/json-simple/

31. Android Oreo // URL: https://developer.android.com/about/versions/oreo/android-8.0

32. Package okhttp3 // URL: https://square.github.io/okhttp

33. Yubisaki to Renren // URL: https://myanimelist.net/anime/55866/Yubisaki_to_Renren (дата обращения: 23.03.24)

34. Dosanko Gal wa Namara Menkoi // URL: https://myanimelist.net/anime/53421/Dosanko_Gal_wa_Namara_Menkoi (дата обращения: 23.03.24)

35. Github: NNanime // URL: https://github.com/topperal/NNanime (дата обращения: 23.03.24)