ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет об исследовательском проекте на тему:

Методы машинного обучения в предсказании рейтинга видеоигр

Выполнил студент:

группы БПМИ238, 2 курса Копнев Максим Михайлович

Принял руководитель проекта:

Алиев Мишан Хаммад оглы
Эксперт, Лаборатория теоретических основ моделей искусственного интеллекта
ФКН, НИУ ВШЭ

Москва 2025

Содержание

Аннотация

Проект посвящен изучению и применению методов машинного обучения в предсказании рейтинга видеоигр на основе датасетов, содержащих общую информацию о них, а также отзывов игроков. В рамках исследования рассмотрены такие типы моделей обучения с учителем, как линейная регрессия, решающие деревья и градиентный бустинг. Для каждой модели проведен сравнительный анализ качества предсказания на двух датасетах: с общими данными и с добавлением данных о тональности отзывов игроков. Для проверки тональности используются готовые методы Python-модуля NLTK. Результатом работы является вывод о сложности задачи предсказания рейтинга видеоигр и приведены возможные варианты улучшения модели в будущем.

Ключевые слова

Машинное обучение, обучение с учителем, регрессия, видеоигры, предсказание рейтинга, отзывы, решающие деревья, градиентный бустинг.

1 Введение

Многие годы мир видеоигр стремительно развивается, и всё больше людей интересуются этой сферой. Обычно перед покупкой люди анализируют различные данные об игре, такие как жанр, студия-разработчик, доступная платформа и тому подобные. Кроме того, о большинстве игр можно прочитать отзывы пользователей на платформе Steam [2], где они выражают своё независимое мнение. Вся эта информация помогает пользователю понять характеристики и качество игры, чтобы принять решение о её покупке.

1.1 Описание предметной области

В основе исследования лежат методы машинного обучения, которое тесно связано с такими областями, как математический анализ, линейная алгебра и математическая статистика [5]. Методы машинного обучения используются во множестве задач. К самым базовым из них относятся классификация, сопоставление объекту класса из определенного множества, и регрессия - сопоставление объекту числового значения, чаще всего вещественного числа. [3]. Конкретными примерами задачи регрессии служат оценка стоимости квартиры по её характеристикам (удаленность от метро, количество комнат) или предсказание доходности предприятия, которое планируется построить. Видеоигры также представляют собой объект, имеющий различные характеристики. Качество видеоигры с точки зрения математики принято измерять либо количеством её продаж, либо рейтингом. Каждая из метрик имеет свои преимущества и имеет свою интерпретацию качества игры, но данное исследование предполагает предсказание рейтинга, то есть решение задачи регрессии.

1.2 Постановка задачи

Задачей нашего исследования является построение эффективной модели машинного обучения, которая поможет пользователю определить рейтинг игры на основании различных данных о ней. Готовые датасеты для обучения моделей взяты с сайта `kaggle.com` [1]. Первый датасет содержит различную информацию об играх, в том числе студию-разработчика и платформу. Второй датасет сопоставляет каждой игре отзывы пользователей на платформе Steam [2] о ней. До начала работы датасеты приводятся в порядок - выкидываются лишние столбцы и удаляются строки с пропусками данных. Затем идет обработка категориальных признаков в первом датасете, чтобы они остались интерпретируемы. Для второго датасета определяется тональность отзывов и каждой игре сопоставляется средняя величина.

Далее датасеты разбиваются на обучающую и тестовые выборки и используются для обучения моделей, таких как линейная регрессия, решающие деревья и градиентный бустинг. При этом обучение проводится отдельно на данных, содержащих столбец тональности отзывов, и данных без него. Затем проводится оценка эффективности каждой из моделей посредством различных метрик и графического изображения предсказаний.

1.3 Цели и гипотезы

Основной целью исследования является разработка и анализ моделей машинного обучения для прогнозирования рейтингов видеоигр на основе данных в открытом доступе, а также оценка факторов, влияющих на качество предсказаний. В исследовании выдвинуты и проверены следующие гипотезы:

- Модели градиентного бустинга покажут более высокую точность прогнозирования рейтинга игр по сравнению с линейными моделями.
- Наиболее значимым фактором в предсказании рейтинга игр являются отзывы пользователей и платформа, на которой выпущена игра.
- Величина студии-разработчика не обязательно находится в прямой зависимости с рейтингом выпускаемых ею игр.

1.4 Актуальность и значимость

В настоящее время на рынке видеоигр ежемесячно появляется множество различных проектов. В интересах пользователей выбрать игру, стоящую их времени и денег. Модель, предсказывающая рейтинг видеоигр может помочь сделать выбор в пользу более качественного проекта. Добавление в модель информации об отзывах других игроков (в случае только вышедшего проекта таковыми могут являться участники бета-тестирования или игроки в раннем доступе), может сделать предсказание более точным. Кроме того, такая модель несёт в себе ценность для области машинного обучения и анализа данных, так как позволяет позволяет математически обосновать те или иные зависимости и сделать вывод об эффективности используемых методов при решении задачи предсказания рейтинга видеоигр.

2 Обзор литературы

2.1 Используемая математика

Основы линейной алгебры исчерпывающе рассмотрены в классическом учебнике А. И. Кострикина [6], что включает в себя операции с векторами и матрицами, теорию о евклидовых пространствах и линейных операторах. Также для понимания алгоритмов машинного обучения нужно ознакомиться с некоторыми главами учебника по математическому анализу Г. М. Фихтенгольца [8].

В переведённом с английского учебнике [7] содержится общая информация по математике для машинного обучения, поэтому его можно использовать в качестве краткого пособия. Кроме того, некоторый теоретический минимум рассмотрен в онлайн учебнике [3].

2.2 Машинное обучение

2.2.1 Основы

В книге [5] описываются теоретические основы машинного обучения, формально описывается понятие модели и рассматриваются решаемые им проблемы. Помимо этого в ней рассматриваются сами методы машинного обучения и способы валидации качества моделей.

2.2.2 Линейная регрессия

Задача регрессии переформулируется как поиск отображения, наилучшим образом (с точки зрения некоей метрики) сопоставляет объект, а точнее его признаки, и некоторое значение из $\mathbb{R}^n$ [3]. Модель линейной регрессии в таком случае возникает тогда, когда мы принимаем, что данное отображение принадлежит семейству линейных функций вида:

$$y = w_1 x_1 + \cdots + w_D x_D + b,$$

где y - целевая переменная, x_i - числовые значения признаков, w_i - веса признаков, b - сдвиг (который можно реализовать через добавление тождественно равного 1 признака).

Пусть $X \in \mathbb{R}^{N \times D}$ - матрица признаков (в i-й строке находятся значения признаков i-го объекта), $y \in \mathbb{R}^N$ - вектор значений целевой переменной. Тогда задача линейной регрессии состоит в нахождениии вектора $w = (w_1, \ldots, w_D, b) \in \mathbb{R}^D$, минимизирующего среднеквадратичную ошибку:

$$MSE = \frac{1}{N} ||y - Xw||_2^2 \to min$$

Эта задача имеет как аналитическое, так и приближенное числовое решение. Аналитическое решение имеет вид:

$$w = (X^T X)^{-1} X^T y$$

Из [6] известно, что у этого решения есть несколько проблем. В частности, нахождение обратной матрицы не всегда возможно. Кроме того, такое решение численно неустойчиво, поэтому зачастую применяется алгоритм градиентного спуска [3, 5].

2.2.3 Решающие деревья

Другая используемая в работе модель - решающие деревья. Модель представляет из себя бинарное дерево, в каждой вершине которого находится либо условие на вектор признаков (обычно сравнение с пороговым значением одного из признаков), либо, если вершина листовая, предсказанное значение целевой переменной. Соответственно, предсказание осуществляется проходом от корня к листу дерева.

Построение решающего дерева для набора данных является NP-полной задачей, поэтому обычно используется жадный алгоритм построения дерева от корня к листам. Необходимо определить критерий остановки (когда вершина объявляется листовой), критерий ветвления (функция, оценивающая предложенное разбиение) и критерий определения ответа для листа. В качестве критерия остановки обычно служит глубина дерева, максимальное число учитываемых признаков и некоторые другие.

Для критерия ветвления необходим выбор функции потерь, в случае задачи регрессии это либо среднеквадратическое отклонение (MSE), либо средняя абсолютная ошибка (MAE). Далее мы ищем константу c, минимизирующую выбранную функцию, причем оптимальное значение константы называют информативностью (impurity). Соответственно, чтобы принять решение о разделении, остается оценить разность информативности в случае, если оставить вершину листом, и в случае разбиения. Подробное описание формулы предоставлено в учебнике ШАД [3].

2.2.4 Градиентный бустинг

Градиентный бустинг - один из самых эффективных способов объединения нескольких моделей в одну для улучшения качества предсказания. В стандартной реализации градиентный бустинг работает над решающими деревьями. Сначала строится решающее дерево $T_1(x)$ для исходных данных. Затем оценивается разность предсказаний построенного дерева $s_1 = y - T_1(x)$. Так как само по себе одно решающее дерево предскажет данные неточно,

строится ещё одно решающее дерево $T_2(x)$, которое обучается предсказывать ту самую разность s_1. Полученная величина $T_1(x) + T_2(X)$ и будет текущим предсказанием модели. Этот процесс можно повторять много раз, постепенно приближая значение целевой переменной. Естественно, велик риск переобучения, поэтому нужно подобрать верные параметры каждого решающего дерева, а также их количество и темп обучения (learning rate) - коэффициент, на который мы будем умножать каждое последующее предсказание в сумме, чтобы снизить вклад каждого следующего дерева в композицию. [3]

2.3 Анализ тональности

Анализ тональности текста - один из подразделов обработки естественного языка (NLP) - позволяет поставить в соответствие тексту метку тональности - некоторое число, определяющее соотношение отрицательных и положительных оттенков в нём [9]. Одним из методов выявления эмоциональной окраски текста является лингвистический анализ, требующий составления словаря с оценкой тональности слов, словосочетаний и символов. Метод заключается обработке текста с помощью словаря и определения общей его тональности путем усреднения суммы оценок. При этом можно получить отдельные степени тональности для положительного, отрицательного и нейтрального типов речи. Данный метод хорошо подходит для использования в анализе коротких текстов, к примеру, постов в социальных сетях или отзывов, так как они редко содержат лексику из предметных областей, что позволяет достаточно эффективно использовать один словарь для анализа.

2.4 Обзор других работ в области предсказания рейтингов видеоигр

В статье [4] проведено схожее с нашим исследование. Основной его целью было создание модели для предсказания глобальных продаж видеоигр, основываясь на общей информации о них, и поиска комбинаций характеристик игр, максимизирующих продажи. Здесь также были использованы различные модели и проведено их сравнение относительно различных метрик, в том числе среднеквадратического отклонения (RMSE). Основное отличие данного исследования от нашего в том, что мы используем общие данные о видеоиграх для предсказания их рейтинга, тогда как автор [4] использует данные о рейтинге наряду с другими для обучения модели, что облегчает предсказание, так как между рейтингом игры и её продажами существует близкая к прямой зависимость. При этом мы используем не только общую информацию о видеоиграх, но и тональность отзывов игроков, чтобы улучшить качество предсказания.

3 Подготовка данных

Для хранения и обработки данных используются методы библиотеки pandas, позволяющие быстро и эффективно выполнять сложные операции с таблицами.

3.1 Обработка общего датасета

Для применения описанных в обзоре литературы методов машинного обучения требовалось правильно обработать данные. Сначала из взятых с kaggle.com [1] датасетов были удалены не используемые в моделях признаки и строки, содержащие нулевые значения (None, NaN, пустые строки). Для предсказания были выбраны следующие признаки: год выпуска, жанры, платформа и студия-разработчик. Так как все признаки можно считать категориальными (в случае с годом это оказалось более эффективно), их необходимо было дополнительно обработать. Все столбцы признаков были проверены на уникальные значения. В случае с платформами и годами выпуска таковых оказалось 22 и 28 соответственно, поэтому решено было использовать One-Hot Encoding (OHE), который каждому уникальному значению сопоставляет новый бинарный признак [3]. OHE реализован в библиотеке pandas через метод get_dummies.

Количество уникальных студий-разработчиков оказалось равно 4376. Для принятия решения о выборе метода обработки была построена диаграмма зависимости количества выпущенных игр от количества разработчиков в одинарном логарифмическом масштабе по вертикальной оси (см. Рисунок 3.1).

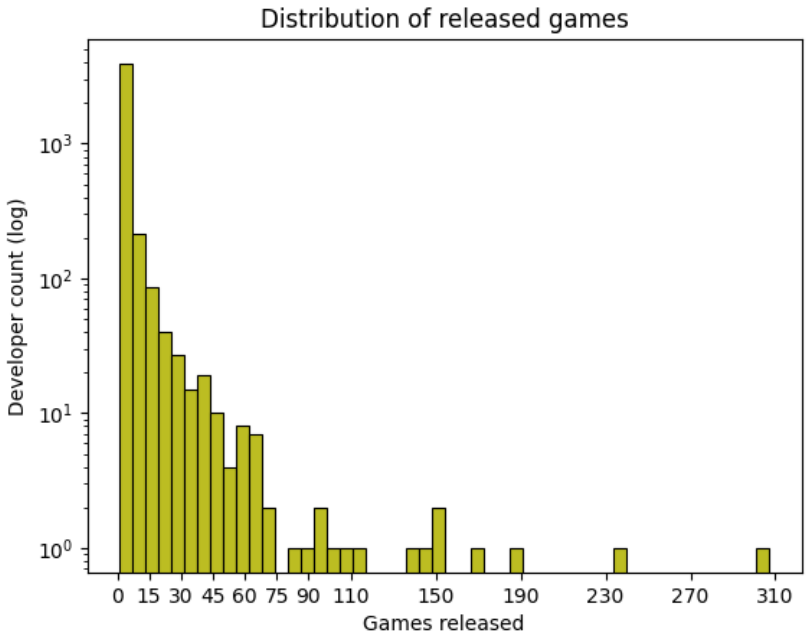

Рисунок 3.1 – Распределение выпущенных игр в зависимости от количества разработчиков

Из диаграммы видно, что большинство разработчиков выпустили меньше 75 игр, а остальных студий около 15. Было решено разделить большую часть разработчиков на 11 групп по количеству выпущенных игр, что позволило получить лучшую интрепретируемость этого признака: его значение определяет размер и популярность студии-разработчика. Итого было получено 26 различных значений в столбце разработчиков, к которым уже был применен OHE.

Столбец жанров содержит список жанров для каждой игры. Он был дополнительно обработан с целью приведения этого списка к виду строки с разделителем. Оказалось, что уникальных значений жанров всего 170, поэтому также был применен OHE, что позволило интерпретировать каждый жанр в качестве отдельного бинарного признака.

3.2 Обработка отзывов

В датасете с отзывами нас интересовали лишь столбы с названием игры и самим текстом отзыва. Для анализа тональности каждого поста использовался SentimentIntensityAnalyzer из библиотеки NLTK, реализующий лингвистический метод определения эмоциональной окраски на основе словаря VADER. Каждому отзыву был сопоставлен параметр compound, отвечающий за совокупную оценку тональности и лежащий в вещественном диапазоне $[-1, 1]$. Затем тональность была усреднена по каждой игре.

3.3 Разбиение данных на выборки

Для обучения моделей данные должны быть разделены на обучающую и тестовую выборки. Для улучшения качества предсказаний был использован параметр stratify функции train_test_split из модуля sklearn.model_selection, который позволяет сделать одинаковое распределение данных в обучающей и тестовой выборках. Также была проведена стандартизация данных с помощью sklearn.preprocessing.StandardScaler.

В итоге были получены обработанные датасеты и обучающая и тестовая выборки, на которых можно обучать модели.

4 Предсказания без отзывов

Сначала модели были обучены на датасете без столбца тональности отзывов.

4.1 Линейная регрессия

Обучение линейной модели было сделано с помощью класса LinearRegression модуля sklearn.linea После были посчитаны средняя абсолютная ошибка в процентах (MAPE), корень из среднеквадратического отклонения (RMSE) и коэффициент детерминации (R^2) для тестовой выборки. Примерные значения ошибок вышли следующие:

$$MAPE \approx 17.7\%, \quad RMSE \approx 1.259, \quad R^2 \approx 0.164$$

Отсюда можно сделать вывод о плохом качестве модели: она ошибается в средем на 17.7%. Низкий коэффициент детерминации отражает высокую долю не объясняемой моделью дисперсии в дисперсии целевой переменной. Для лучшего понимания качества предсказаний была построена диаграмма сравнения реальных значений рейтинга с предсказанными на тестовой выборке (см. Рисунок 4.1). Здесь координаты точки по горизонтальной оси отражают реальное значение целевой переменной, а координаты по вертикальной оси - предсказанные моделью значения. Красная линия отражает идеальную модель (реальность = прогноз), поэтому чем ближе точка к красной линии, тем выше качество предсказания для конкретного объекта.

Из диаграммы видно, что для большинства объектов модель предсказывает значение в диапазоне $[5, 8]$. Интуитивно можно предположить. что рейтинг игр чаще всего оказывается в этом диапазоне значений. Для подтверждения гипотезы было решено проверить распределение значений целевой переменной на тестовой и тренировочной выборках (см. Рисунок 4.2).

Гипотеза подтвердилась: данные оказались сильно смещены в сторону диапазона $[6, 9]$, из-за чего линейная модель имеет плохую обобщающую способность: то есть сильно ошибается при предсказании на играх с низким рейтингом. Были проверены различные способы улучшения распределения данных, в том числе удаление части данных в данном диапазоне. Однако такие методы привели лишь к увеличению ошибок модели, из чего был сделан вывод о том, что для решения задачи лучше подойдут другие алгоритмы, которые позволят выявить нелинейные зависимости в данных и будут более устойчивы к их распределению.

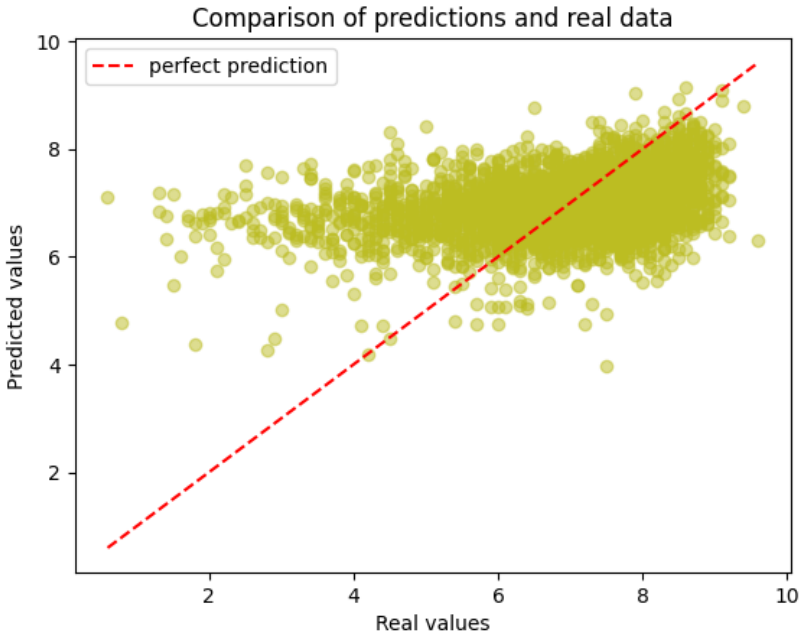

Рисунок 4.1 – Сравнение реальных и предсказанных значений для линейной модели без отзывов. Зеленые точки - положение объекта в координатах (реальный рейтинг, предсказанный рейтинг), красная линия - идеальное предсказание.

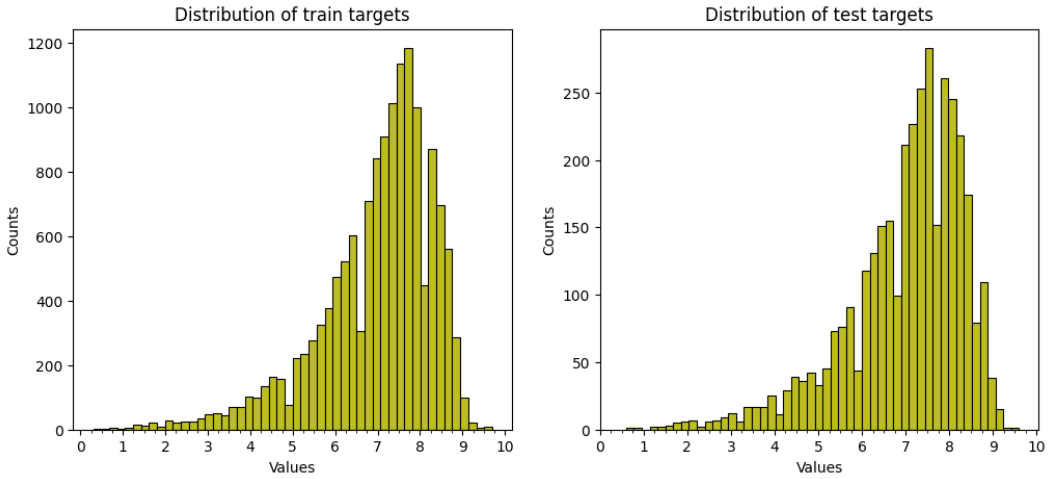

Рисунок 4.2 – Распределение значений целевой переменной на выборках

4.2 Решающие деревья

В качестве первой нелинейной модели было выбрано решающее дерево. Для его обучения требовалось найти оптимальную максимальную глубину дерева для наших данных. Для этого был использован GridSearchCV из sklearn.model_selection, откуда было получено значение 7. Решающее дерево было обучено с помощью класса DecisionTreeRegressor модуля sklearn.tree. Для модели были получены следующие значения ошибок на тестовой выборке:

$$MAPE \approx 18.4\%, \quad RMSE \approx 1.305, \quad R^2 \approx 0.058$$

Эти значения логичны: одно решающее дерево не может точно уловить сложные зависимости, которые присутствуют в наших данных, что можно заключить из результатов для линейной модели, а, следовательно, и корректно обучиться на используемом многомерном наборе данных. Для консистентности исследования была также построена диаграмма сравнения (см. Рисунок 4.3).

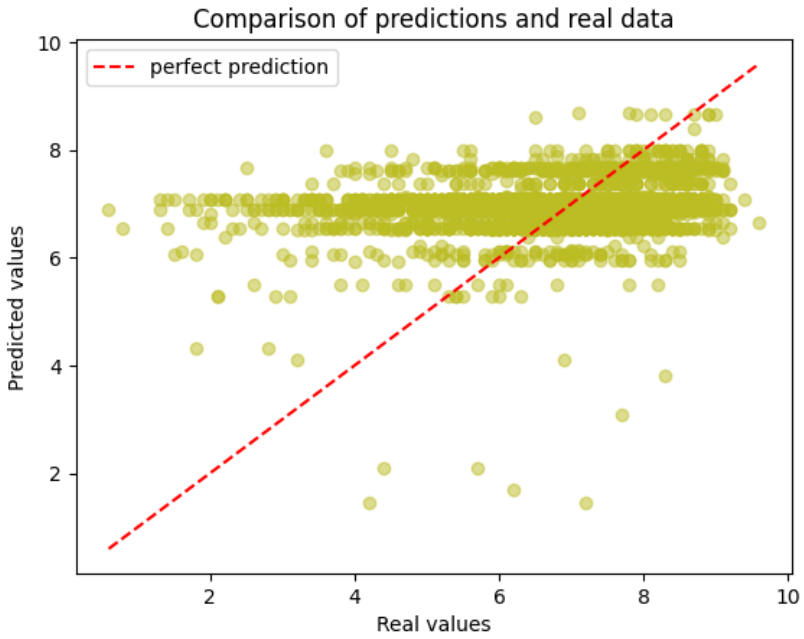

Рисунок 4.3 – Сравнение реальных и предсказанных значений для одиночного решающего дерева без отзывов. Зеленые точки - положение объекта в координатах (реальный рейтинг, предсказанный рейтинг), красная линия - идеальное предсказание.

4.3 Градиентный бустинг

В качестве продвинутого метода объединения нескольких решающих деревьев был выбран градиентный бустинг, как один из самых продвинутых видов моделей. Для его обучения использовался класс GradientBoostingRegressor из модуля skl earn.ensemble. Подбор гиперпараметров был произведен локально ввиду ограниченных вычислительных скоростей Google Colab[1]. Ввиду довольно высокого числа признаков параметр max_features был выбран как двоичный логарифм, то есть при поиске разбиения в деревьях мы используем лишь логарифм от общего числа признаков. Количество обучаемых решающих деревьев - 1000, темп обучения - 0.025, максимальная глубина каждого из них - 11. В итоге ошибки модели

[1]Google Colab - облачный сервис для программирования на Python, зачастую используемый в машинном обучении и анализе данных

градиентного бустинга на тестовых данных:

$$MAPE \approx 16.6\%, \quad RMSE \approx 1.209, \quad R^2 \approx 0.192$$

Заметно улучшение качества предсказаний, при этом модель всё равно осталась недостаточно точной, что можно заключить из диаграммы на Рисунке 4.4.

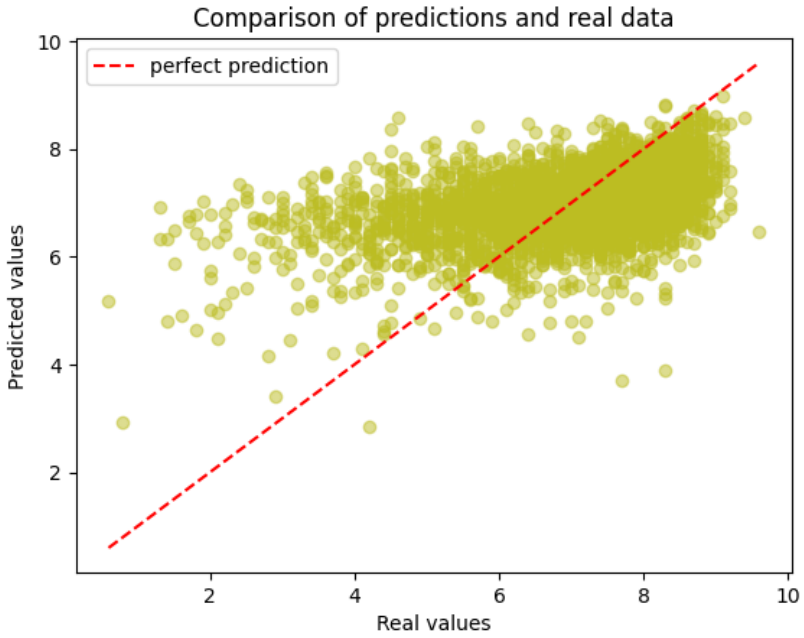

Рисунок 4.4 – Сравнение реальных и предсказанных значений для градиентного бустинга на решающих деревьях без отзывов. Зеленые точки - положение объекта в координатах (реальный рейтинг, предсказанный рейтинг), красная линия - идеальное предсказание.

4.4 Выводы

Задача предсказания рейтинга игры без использования данных об отзывах пользователей оказалась довольно сложной. Это может быть обусловлено как неравномерным распределением данных, так и отсутствием хорошо интерпретируемых числовых признаков. Следующим шагом необходимо проверить гипотезу о том, что тональность отзывов может улучшить качество предсказаний рассмотренных моделей.

5 Предсказания с отзывами

5.1 Обучение моделей

Теперь для предсказания был использован датасет, содержащий столбец усредненной тональности отзывов об игре. Те же виды моделей были обучены на дополненных данных с уже найденными параметрами. Новые ошибки линейной модели составили:

$$MAPE \approx 15.8\%, \quad RMSE \approx 1.144, \quad R^2 \approx 0.279$$

Было получено существенное уменьшение MAPE и RMSE и увеличение коэффициента детерминации. Из этого уже можно заключить, что тональность отзыва находится в близкой к прямой зависимости с целевой переменной, из-за чего линейная модель лучше справляется с поставленной задачей. На Рисунке 5.1 приведена новая сравнительная диаграмма.

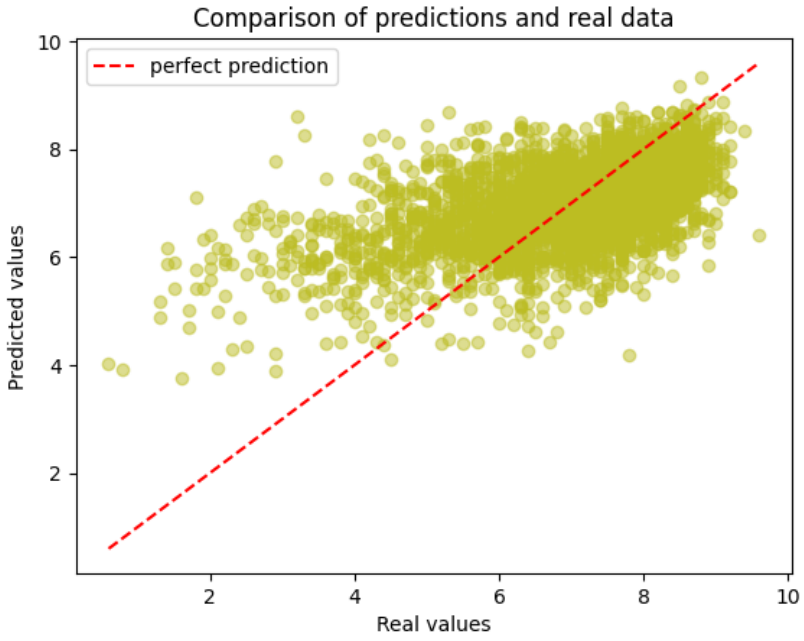

Рисунок 5.1 – Сравнение реальных и предсказанных значений для линейной модели с отзывами. Зеленые точки - положение объекта в координатах (реальный рейтинг, предсказанный рейтинг), красная линия - идеальное предсказание.

Аналогично было обучено решающее дерево глубины 7, полученные ошибки:

$$MAPE \approx 15.7\%, \quad RMSE \approx 1.161, \quad R^2 \approx 0.257$$

Одиночное дерево решений с использованием тональности отзывов дало гораздо более точные предсказания относительно уровней ошибок, однако из диаграммы на Рисунке 5.2 следует очень низкая обобщающая способность модели.

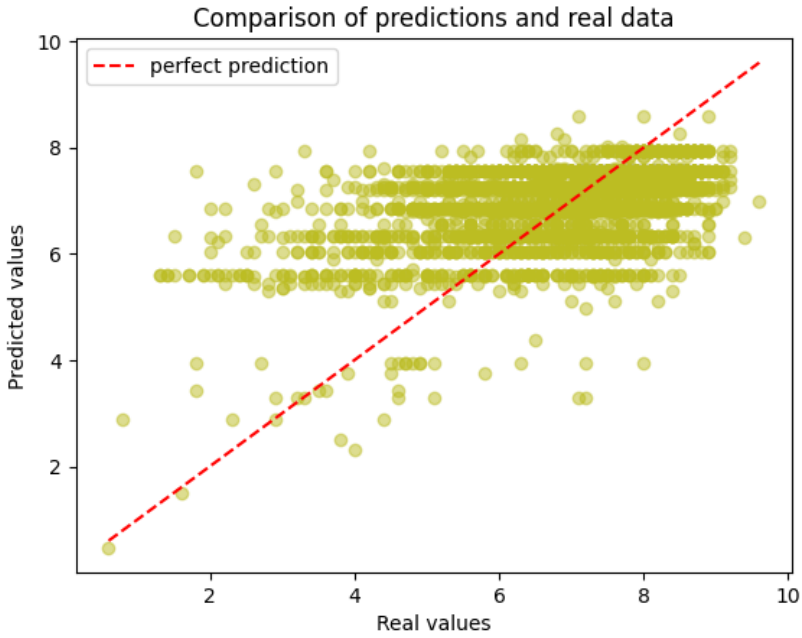

Рисунок 5.2 – Сравнение реальных и предсказанных значений для одиночного решающего дерева с отзывами. Зеленые точки - положение объекта в координатах (реальный рейтинг, предсказанный рейтинг), красная линия - идеальное предсказание.

Для модели градиентного бустинга, обученной с теми же гиперпараметрами, были получены следующие метрики:

$$MAPE \approx 14.2\%, \quad RMSE \approx 1.062, \quad R^2 \approx 0.378$$

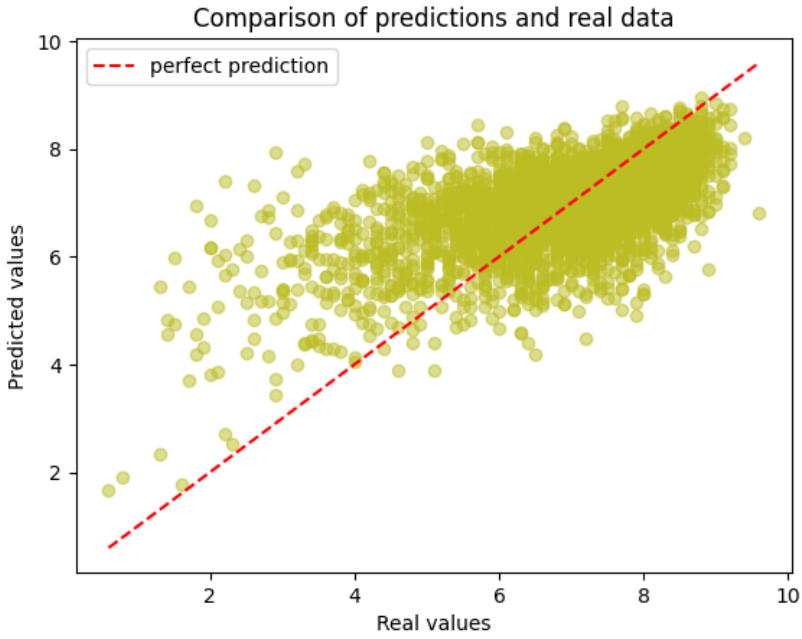

Рисунок 5.3 – Сравнение реальных и предсказанных значений для градиентного бустинга с отзывами. Зеленые точки - положение объекта в координатах (реальный рейтинг, предсказанный рейтинг), красная линия - идеальное предсказание.

Из R^2 можно заключить, что модель приблизилась к среднему уровню качества предсказаний (соответствующему $R^2 \approx 0.5$), что, по сравнению с изначальными результатами, можно считать прогрессом. Аналогично предыдущим моделям была построена диаграмма (см. Рисунок 5.3).

5.2 Выводы

Если рассматривать полученные для линейной регрессии и градиентного бустинга диаграммы в динамике, то можно заметить, что облако точек постепенно поворачивается, подстраиваясь под линию идеального предсказания, из чего следует, что добавление данных об отзывах и использование градиентного бустинга позволили значительно уменьшить последствия неравномерного распределения данных. Кроме того, модели с использованием тональности показали лучшие результаты с точки зрения метрик, чем модели, обученные на общей информации, что наглядно видно из Таблицы 5.1.

	Предсказания без отзывов			Предсказания с отзывами		
	MAPE	RMSE	R^2-score	MAPE	RMSE	R^2-score
Линейная регрессия	17.7%	1.259	0.164	15.8%	1.144	0.279
Решающее дерево	18.4%	1.305	0.058	15.7%	1.161	0.257
Градиентный бустинг	16.6%	1.209	0.192	**14.2%**	**1.062**	**0.378**

Таблица 5.1 – Сравнение метрик для обученных моделей. Жирным выделены лучшие результаты.

Из этого можно сделать вывод о высокой важности тональности отзывов. Это было проверено построением столбчатой диаграммы важности признаков (см. Рисунок 5.4).

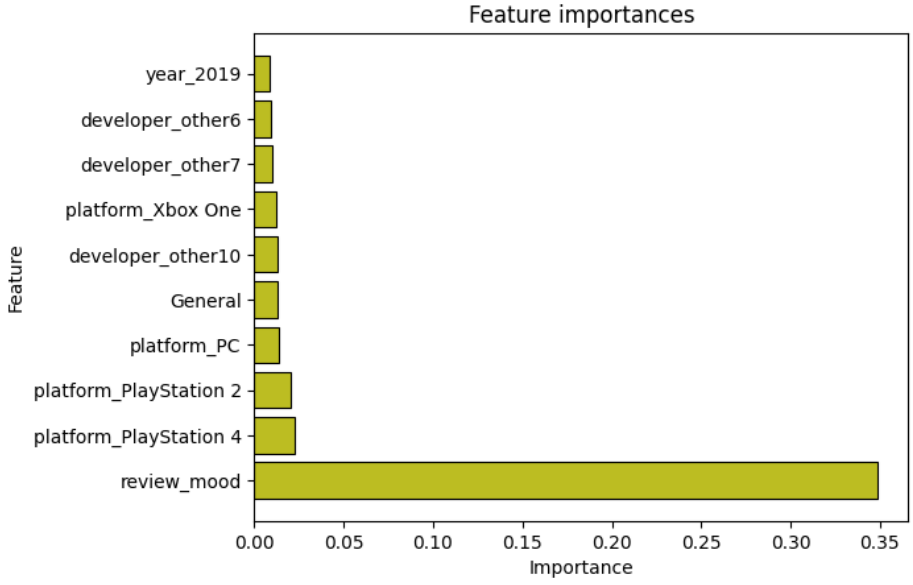

Рисунок 5.4 – Диаграмма важности признаков

Важность признака review_mood, то есть средней тональности отзывов, в разы превысила важность остальных признаков. В первую очередь это объясняется тем, что остальные признаки категориальны и были обработаны с помощью OHE, поэтому важность каждого отдельного из них достаточно низкая. При этом можно заметить, что платформы PlayStation4 и Playstation2 также часто использовались для предсказания, что в контексте решающих деревьев значит, что разбиение дерева по этим признакам чаще улучшало impurity, чем по другим. Эта информация соотносится с реальностью, так как PlayStation4 и Playstation2 считаются одними из лучших консолей, при этом PC также имеет сравнительно высокую важность, что обусловлено популярностью персональных компьютеров в гейминге. Также довольно информативен тот факт, что группа developer_other10, соответствующая разработчикам, выпустившим ровно 1 игру (так называемые инди-студии[2]), также попала в десятку самых важных признаков, что подтверждает гипотезу об отсутствии явной прямой зависимости между величиной студии и рейтингом игры.

[2]Инди-разработчики (англ. indie, от. англ. independent) - независимые от крупных издателей студии-разработчики, чаще всего выпускающие небольшие бюджетные игры. В игровом сообществе инди-разработчики считаются надеждой индустрии ввиду увеличения частоты появления провальных с точки зрения качества игр от крупных студий.

6 Заключение

Положительным итогом исследования является модель градиентного бустинга, обученная в том числе на данных о тональности отзывов, и способная с приемлемой точностью предсказывать рейтинг видеоигр. Таким образом, из трёх рассмотренных методов машинного обучения градиентный бустинг лучше других подошел для поставленной задачи. Линейная регрессия не сработала, так как наблюдаемая зависимость между характеристиками игр и их рейтингом оказалась далека от линейной, а одиночное решающее дерево не смогло выработать достаточную обобщающую способность для корректного предсказания.

Кроме того, на основе ошибок и построенных диаграмм можно сделать вывод о наблюдаемой положительной корреляции между рейтингов игры и средней тональностью отзывов о ней. Это соотносится с интуитивным пониманием зависимости, но этот вывод показывает, что выбранный метод лингвистического анализа применим в решении задачи предсказания рейтинга. При этом использование тональности отзывов можно считать не совсем оправданным, так как не для любой игры могут иметься такие отзывы, в том числе потому, что игра ещё не вышла. В таком случае отзывы могут быть заменены на комментарии под анонсами игры в социальных сетях или другую подобную информацию.

Основными проблемами в проведенном исследовании стали:

- **Неравномерное распределение данных** - чаще всего в столбце рейтинга встречались значения из диапазона $[6, 9]$, что соответствует реальности, но критически влияет на обобщающую способность обучаемых моделей
- **Отсутствие числовых признаков** - все данные основного датасета были категориальными. В открытом доступе не было найдено набора данных, содержащего наряду с используемыми признаками некоторые дополнительные (к примеру, цену). Это объясняется в том числе сложностью видеоигр как объекта для предсказания: игры выпускаются на разных платформах разными разработчиками, из-за чего сбор обширных данных о них довольно затруднителен.
- **Использование тональности отзывов** - как было доказано, тональность отзывов сильно коррелирует с целевой переменной, что уменьшает ценность предсказаний модели, так как любой человек сам может прочитать отзывы и принять решение о покупке.

Таким образом, ещё одним итогом исследования можно считать определение применимости методов машинного обучения в задаче предсказания рейтинга видеоигр. Методы машин-

ного обучения могут быть применены для этой задачи, но, если считать неизменными используемые для обучения данные, довольно посредственно справляются с ней. Это обсуловлено в том числе тем, что видеоигры являются интерактивным искусством, а следовательно плохо поддаются строгому математическому описанию ввиду крайне высокой сложности зависимостей их рейтинга от множества характеристик, не все из которых возможно измерить и добавить в датасет для обучения модели. Данное исследование предполагает продолжение с работой в следующих направлениях:

- **Сбор данных** - первоначально для улучшения результатов необходимо провести ручной парсинг различных данных об играх на всевозможных платформах. Кроме того, необходимо использовать продвинутые методы конструирования признаков (feature engineering) для увеличения их интерпретируемости.
- **Использование продвинутого NLP** - использование в предсказании тональности специализированных текстов (отзывов) можно заменить парсингом комментариев и постов пользователей в социальных сетях. Это сделает зависимость с целевой переменной менее явной и, в случае положительных результатов, увеличит научную и практическую ценность модели. Кроме того, можно рассмотреть идею обработки кратких описаний игр с помощью методов NLP, что вполне может привести к неожиданным результатам.
- **Использование иных методов предсказания** - вывод о сложности задачи предсказания рейтинга видеоигр приводит нас к тому, что для решения этой задачи можно попробовать использовать более сложные методы, такие как глубинное обучение. Однако в данном случае придется столкнуться с проблемой нехватки данных.

Список литературы

[1] Kaggle: Machine learning and data science community. URL: https://kaggle.com. (дата обр. 1.04.2025).

[2] Steam store. https://store.steampowered.com/, 2003. (дата обр. 1.04.2025).

[3] Учебник по машинному обучению. Яндекс Образование. https://education.yandex.ru/handbook/ml, 2021. (дата обр. 1.04.2025).

[4] Hannah Igboke. Prediction and classification of video games. https://medium.com/@HannahIgboke/prediction-and-classification-of-video-games-c8bafd86f5e1, 2024. (дата обр. 1.04.2025).

[5] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

[6] Кострикин А.И. *Введение в алгебру в 3 томах.* МЦНМО, 2012.

[7] Он Ч. Дайзенрот М., Фейзал А. *Математика в машинном обучении.* Питер, 2024.

[8] Фихтенгольц Г. М. *Основы математического анализа в 2 томах.* Наука, 1968.

[9] Самигулин Тимур Русланович и Джурабаев Анвар Эркин Угли. Анализ тональности текста методами машинного обучения. *Научный результат. Информационные технологии*, 2021.