МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южный федеральный университет»
Институт высоких технологий и пьезотехники

Кафедра прикладной информатики и инноватики

Направление: 09.03.03 "Прикладная информатика"

Индивидуальный проект

" Сбор, предобработка и анализ эффективности раннего доступа для компьютерных игр"

Выполнили студенты 3 курса

Мозговой М.А. _____
подпись

Елинский Д.В. _____
подпись

Ростов-на-Дону, 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Эффективность раннего доступа для компьютерных игр является важной темой как для разработчиков, так и для игроков. Ранний доступ позволяет разработчикам выпускать незавершенные игры, чтобы собрать обратную связь от игроков и улучшить продукт до официального релиза. Вот несколько ключевых аспектов, которые подчеркивают важность раннего доступа:

1. Финансовая поддержка

Ранний доступ предоставляет разработчикам возможность получить финансовую поддержку на ранних стадиях разработки. Это особенно важно для независимых разработчиков, у которых может не быть других источников финансирования.

2. Обратная связь от игроков

Игроки, участвующие в раннем доступе, могут предоставлять ценную обратную связь, помогая выявлять баги, улучшать геймплей и балансировку. Это позволяет разработчикам внести необходимые изменения до официального релиза.

3. Формирование сообщества

Ранний доступ способствует формированию и укреплению сообщества вокруг игры. Игроки, участвующие в процессе разработки, чувствуют себя частью проекта, что может увеличить их лояльность и готовность рекомендовать игру другим.

4. Маркетинговый эффект

Ранний доступ может служить маркетинговым инструментом. Игра, доступная на ранних стадиях, может привлечь внимание СМИ и блогеров, что способствует повышению осведомленности о проекте.

5. Проверка концепции

Для разработчиков ранний доступ является возможностью проверить жизнеспособность и интерес к их игре. Если игра получает положительные отзывы и значительное количество продаж, это может служить индикатором успешности будущего релиза.

Цель: проанализировать эффективность раннего доступа для компьютерных игр основываясь на данных Датасета.

.

Гипотеза

“Ранний положительно влияет на успех компьютерной игры в долгосрочной перспективе, способствуя улучшению качества конечного продукта и формированию активного сообщества игроков. Для этого необходимо сравнить показатели игр без раннего доступа и игр прошедших через ранний доступ.”

Описание датасета

Датасет об отзывах пользователей к играм онлайн-сервиса цифрового распространения компьютерных игр и программ “Steam” - 100 Million+ Steam Reviews представляет собой таблицу с 24 столбцами и с 498k отзывами.
Столбцы датасета:

1. Recommendationid (id отзыва)
2. Appid (id игры)
3. Game (название игры)
4. author_steamid (id пользователя)
5. author_num_games_owned (кол. игр во владении пользователя)
6. author_num_reviews (количество отзывов пользователя)
7. author_playtime_forever (общее время в игре)
8. author_playtime_last_two_weeks (время в игре за последние 2 недели)

9. author_playtime_at_review (время в игре на момент написания отзыва)
10. author_last_played (последнее время появления)
11. language (язык)
12. review (отзыв)
13. timestamp_created (дата создания отзыва)
14. timestamp_updated (дата, когда отзыв был отредактирован последний раз)
15. voted_up (положительный или отрицательный отзыв)
16. votes_up (количество пользователей оценивших полезность этого отзыва)
17. votes_funny (количество пользователей оценивших этот отзыв забавным)
18. weighted_vote_score (оценка полезности отзыва)
19. comment_count (количество комментариев под отзывом)
20. steam_purchase (есть ли эта игра у пользователя в библиотеке)
21. received_for_free (игра получена бесплатно)
22. written_during_early_access (был ли отзыв написан во время раннего доступа)
23. hidden_in_steam_china (доступно ли в Китае)
24. steam_china_location (находится ли в Китае)

Ход работы

Предобработка датасета:

В датасете на момент установки находилось 56789 строк с ошибочными значениями и 40 строк с нулевыми значениями, от которых необходимо было избавиться.

```
steamCountIncorrectVotesDF = spark.sql("""SELECT count(*)
                             FROM steam
                             WHERE voted_up NOT IN (0,1)
                              """)
steamCountIncorrectVotesDF.printSchema()
steamCountIncorrectVotesDF.show()
```

```
root
 |-- count(1): long (nullable = false)
```

```
[Stage 275:>
```

```
+--------+
|count(1)|
+--------+
|   56789|
+--------+
```

```
steamNullGameDF = spark.sql("""SELECT count(*)
                               FROM steam
                               WHERE game IS NULL
                                """)
steamNullGameDF.printSchema()
steamNullGameDF.show()
```

```
root
 |-- count(1): long (nullable = false)
```

```
[Stage 277:>
```

```
+--------+
|count(1)|
+--------+
|      40|
+--------+
```

Следующий этап предобработки заключался в выборке необходимых нам столбцов.
Для последующего анализа мы использовали столбцы:

- Recommendationid
- Appid
- Game
- voted_up
- written_during_early_access

Новый датасет содержет в себе 441058 строк

```
steamCountDF = spark.sql("""SELECT count(*)
                            FROM steam
                             """)
steamCountDF.printSchema()
steamCountDF.show()
```

```
root
 |-- count(1): long (nullable = false)
```

```
[Stage 284:=======================================>
```

```
+--------+
|count(1)|
+--------+
|  441058|
+--------+
```

Проверка популярности:
Для начала мы решили проверить популярность такого решения, как выпускать свою игру в раннем доступе среди разработчиков.

Из 24935 игр, 21617 были выпущены без раннего доступа.

```
steamCountGamesDF = spark.sql("""SELECT COUNT(DISTINCT game)
                                 FROM steam
                              """)
steamCountGamesDF.printSchema()
steamCountGamesDF.show()
```

```
root
 |-- count(DISTINCT game): long (nullable = false)
```

```
[Stage 287:================================================>     (180 + 6) / 200]
```

```
+--------------------+
|count(DISTINCT game)|
+--------------------+
|               24935|
+--------------------+
```

```
steamCountGamesWithoutEADF = spark.sql("""SELECT COUNT(DISTINCT game)
                                  FROM steam
                                  WHERE appid NOT IN (
                                     SELECT DISTINCT appid
                                     FROM steam
                                     WHERE written_during_early_access = 1)
                              """)
steamCountGamesWithoutEADF.printSchema()
steamCountGamesWithoutEADF.show()
```

```
root
 |-- count(DISTINCT game): long (nullable = false)
```

```
[Stage 291:====================>                                    (2 + 4) / 6]
```

```
+--------------------+
|count(DISTINCT game)|
+--------------------+
|               21617|
+--------------------+
```

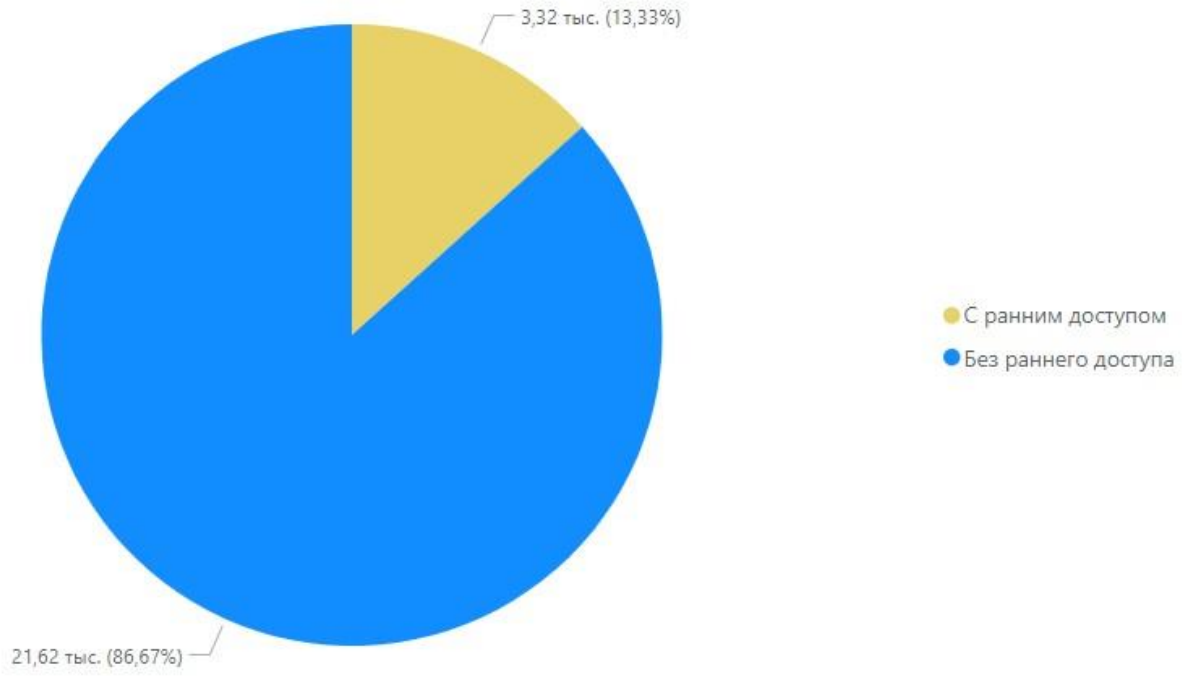
Остальные выпускались в раннем доступе

```
steamCountGamesWithEADF = spark.sql("""SELECT COUNT(DISTINCT game)
                                       FROM steam
                                       WHERE appid IN (
                                          SELECT DISTINCT appid
                                          FROM steam
                                          WHERE written_during_early_access = 1)
                                    """)
steamCountGamesWithEADF.printSchema()
steamCountGamesWithEADF.show()
```

```
root
 |-- count(DISTINCT game): long (nullable = false)
```

```
[Stage 297:=====================================================> (193 + 7) / 200]
```

```
+--------------------+
|count(DISTINCT game)|
+--------------------+
|                3324|
+--------------------+
```



Необходимо было оценить влияние такого фактора как ранний доступ на оценки пользователей.

```
steamAVGRatesWithEADF = spark.sql("""SELECT AVG(voted_up)
                                     FROM steam
                                     WHERE appid IN (
                                        SELECT DISTINCT appid
                                        FROM steam
                                        WHERE written_during_early_access = 1)
                                  """)
steamAVGRatesWithEADF.printSchema()
steamAVGRatesWithEADF.show()
```

```
root
 |-- avg(voted_up): double (nullable = true)
```

```
[Stage 305:=====================================================>  (191 + 6) / 
```

```
+------------------+
|     avg(voted_up)|
+------------------+
|0.8707621182404035|
+------------------+
```

```
steamAVGRatesWithoutEADF = spark.sql("""SELECT AVG(voted_up)
                                        FROM steam
                                        WHERE appid NOT IN (
                                           SELECT DISTINCT appid
                                           FROM steam
                                           WHERE written_during_early_access = 1)
                                     """)
steamAVGRatesWithoutEADF.printSchema()
steamAVGRatesWithoutEADF.show()
```

```
root
 |-- avg(voted_up): double (nullable = true)
```

```
[Stage 301:>                                                        (0 + 6) / 6]
```

```
+-----------------+
|    avg(voted_up)|
+-----------------+
|0.854798520194884|
+-----------------+
```

Из полученных данных можно увидеть, что игры с ранним доступом имеют более высокую среднюю оценку нежели игры без раннего доступа.

Оценка влияния:
Необходимо было проверить имеют ли игры с ранним доступом положительную динамику в долгосрочной перспективе. Для этого мы высчитали средние оценки у игр во время раннего доступа и после:

```
steamAVGRatesWithEABeforeDF = spark.sql("""SELECT AVG(voted_up)
                                    FROM steam
                                    WHERE appid IN (
                                       SELECT DISTINCT appid
                                       FROM steam
                                       WHERE written_during_early_access = 1)
                                    AND written_during_early_access = 1
                                """)
steamAVGRatesWithEABeforeDF.printSchema()
steamAVGRatesWithEABeforeDF.show()
```

```
root
 |-- avg(voted_up): double (nullable = true)
```

```
[Stage 309:================================================>    (184 + 6)
```

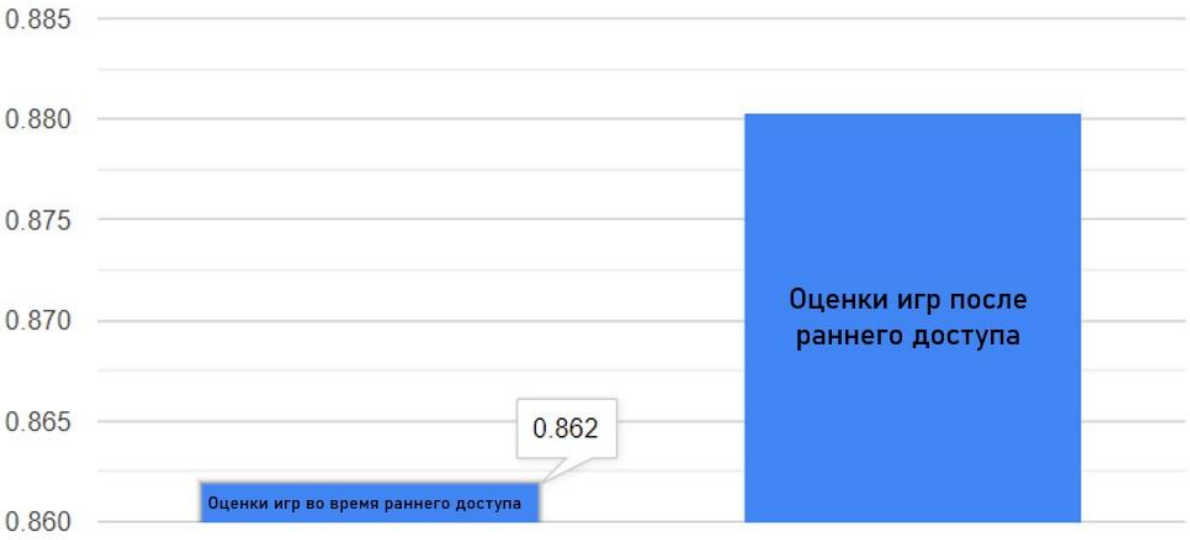
```
+------------------+
|     avg(voted_up)|
+------------------+
|0.8619422370167978|
+------------------+
```

```
steamAVGRatesWithEAAfterDF = spark.sql("""SELECT AVG(voted_up)
                                    FROM steam
                                    WHERE appid IN (
                                       SELECT DISTINCT appid
                                       FROM steam
                                       WHERE written_during_early_access = 1)
                                    AND written_during_early_access = 0
                                """)
steamAVGRatesWithEAAfterDF.printSchema()
steamAVGRatesWithEAAfterDF.show()
```

```
root
 |-- avg(voted_up): double (nullable = true)
```

```
[Stage 312:>                                                        (0 + 6)
```

```
+------------------+
|     avg(voted_up)|
+------------------+
|0.8803194262004546|
+------------------+
```



Из полученных результатов видно, что у игр с выходом из раннего доступа средняя оценка поднималась.

Повторное сравнение:
Для актуальности результатов мы решили сравнить оценки игр без раннего доступа и игры после раннего доступа, так как аналитики при анализе оценок опираются на текущие показатели, а не на те, которые были написаны во время раннего доступа.

```
steamAVGRatesWithoutEADF = spark.sql("""SELECT AVG(voted_up)
                                    FROM steam
                                    WHERE appid NOT IN (
                                       SELECT DISTINCT appid
                                       FROM steam
                                       WHERE written_during_early_access = 1)
                                """)
steamAVGRatesWithoutEADF.printSchema()
steamAVGRatesWithoutEADF.show()
```

```
root
 |-- avg(voted_up): double (nullable = true)
```

```
[Stage 301:>                                                        (0 + 6) / 6]
```

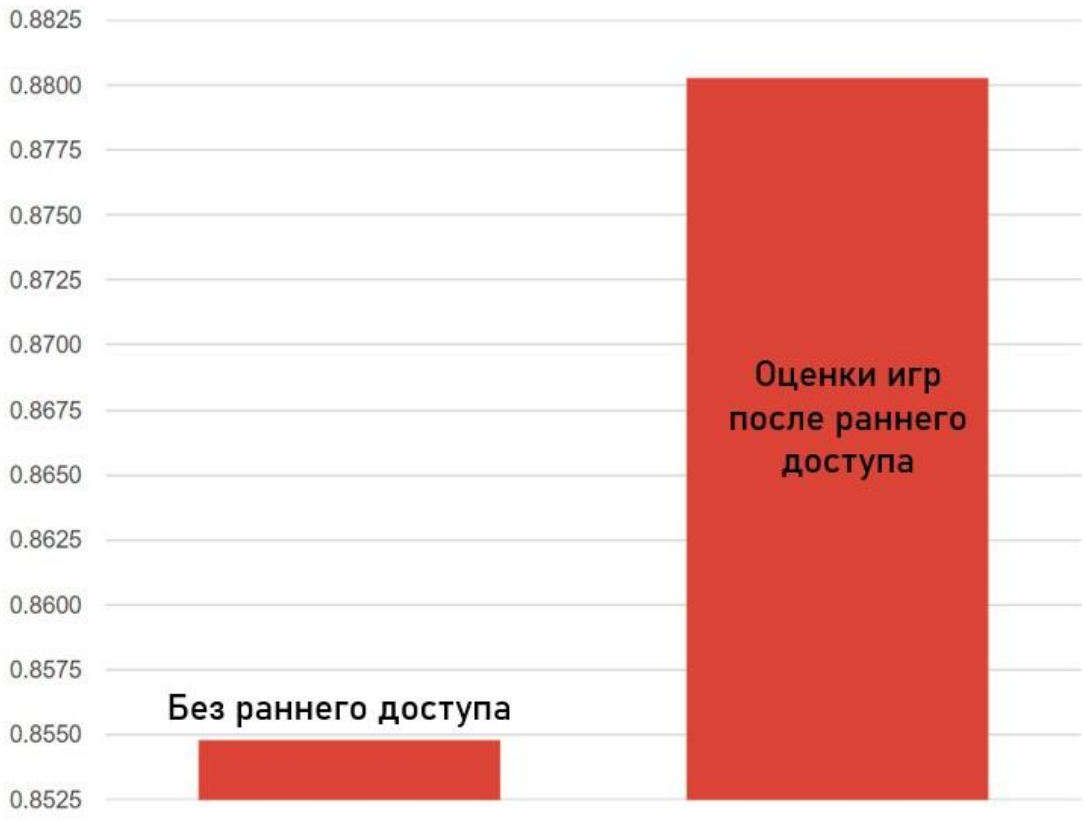
```
+-----------------+
|    avg(voted_up)|
+-----------------+
|0.854798520194884|
+-----------------+
```

```
steamAVGRatesWithEAAfterDF = spark.sql("""SELECT AVG(voted_up)
                                     FROM steam
                                     WHERE appid IN (
                                        SELECT DISTINCT appid
                                        FROM steam
                                        WHERE written_during_early_access = 1)
                                     AND written_during_early_access = 0
                                  """)
steamAVGRatesWithEAAfterDF.printSchema()
steamAVGRatesWithEAAfterDF.show()
```

```
root
 |-- avg(voted_up): double (nullable = true)
```

```
[Stage 312:>                                                        (0 + 6)
```

```
+------------------+
|     avg(voted_up)|
+------------------+
|0.8803194262004546|
+------------------+
```



Из полученного сравнения видно, что оценка игр после раннего доступа превосходит над оценкой игр без раннего доступа.

Попытка ограничить выборку:
Было выдвинуто предложение уменьшить разницу в количестве игр с ранним доступом и без для получения более точных данных.
Было выдвинуто предположение, что более популярные игры дадут более качественные результаты.
Для этого мы нашли среднее количество отзывов для игр с ранним доступом и без него.

```
steamAvgCountRatesWithoutEADF = spark.sql("""SELECT avg(count_reviews)
                              FROM steam_without_ea
                               """)
steamAvgCountRatesWithoutEADF.printSchema()
steamAvgCountRatesWithoutEADF.show()
```

```
root
 |-- avg(count_reviews): double (nullable = true)

+------------------+
|avg(count_reviews)|
+------------------+
|15.780450571309617|
+------------------+
```

```
steamAvgCountRatesWithEADF = spark.sql("""SELECT avg(count_reviews)
                              FROM steam_with_ea
                               """)
steamAvgCountRatesWithEADF.printSchema()
steamAvgCountRatesWithEADF.show()
```

```
root
 |-- avg(count_reviews): double (nullable = true)

+------------------+
|avg(count_reviews)|
+------------------+
|30.063778580024067|
+------------------+
```

Для дальнейших исследовваний было принято решение ограничить количество игр до игр с минимальным количеством озывов = 30.

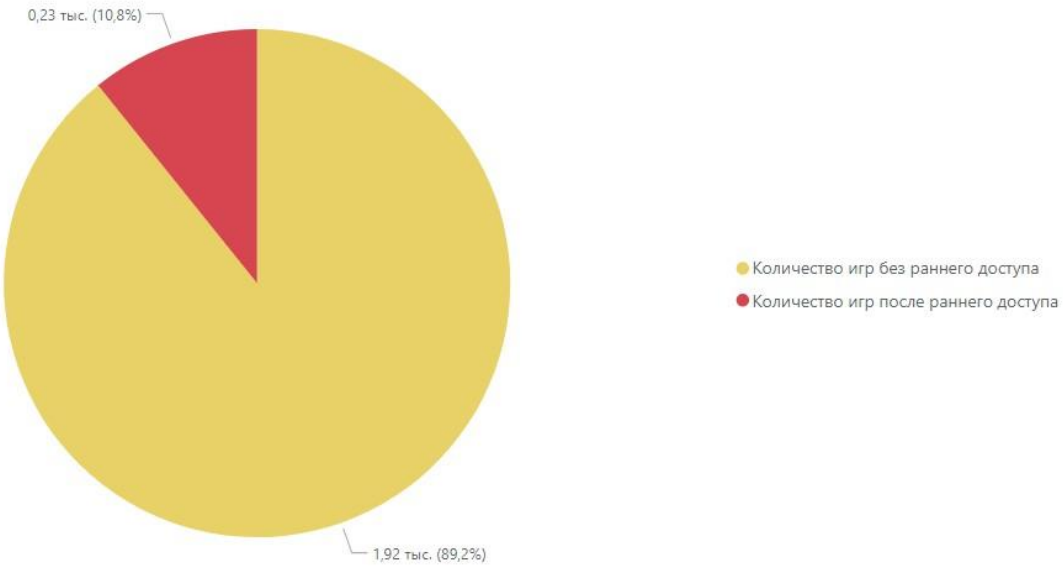
Повторное преобразование датасета:

```
steamCountWithEAAfterMore30DF = spark.sql("""SELECT count(*)
                            FROM steam_with_ea_after
                            WHERE count_reviews > 30
                             """)
steamCountWithEAAfterMore30DF.printSchema()
steamCountWithEAAfterMore30DF.show()
```

```
root
 |-- count(1): long (nullable = false)

+--------+
|count(1)|
+--------+
|     232|
+--------+
```

```
steamCountWithoutEAMore30DF = spark.sql("""SELECT count(*)
                            FROM steam_without_ea
                            WHERE count_reviews > 30
                             """)
steamCountWithoutEAMore30DF.printSchema()
steamCountWithoutEAMore30DF.show()
```

```
root
 |-- count(1): long (nullable = false)

+--------+
|count(1)|
+--------+
|    1917|
+--------+
```



Теперь разница в выборке была не настолько велика, но оставалась довольно большой.

Повторное сравнение:

```
steamAvgRatingWithEAAfterMore30DF = spark.sql("""SELECT avg(avg_rating)
                            FROM steam_with_ea_after
                            WHERE count_reviews > 30
                             """)
steamAvgRatingWithEAAfterMore30DF.printSchema()
steamAvgRatingWithEAAfterMore30DF.show()
```

```
root
 |-- avg(avg_rating): double (nullable = true)

+------------------+
|   avg(avg_rating)|
+------------------+
|0.8701199984976935|
+------------------+
```

```
steamAvgRatingWithoutEAMore30DF = spark.sql("""SELECT avg(avg_rating)
                            FROM steam_without_ea
                            WHERE count_reviews > 30
                             """)
steamAvgRatingWithoutEAMore30DF.printSchema()
steamAvgRatingWithoutEAMore30DF.show()
```

```
root
 |-- avg(avg_rating): double (nullable = true)

+------------------+
|   avg(avg_rating)|
+------------------+
|0.8541110053684383|
+------------------+
```

После повторного сравнения полученных датасетов результаты показывали, что оценки игр после раннего доступа были выше, чем у игр без раннего доступа.

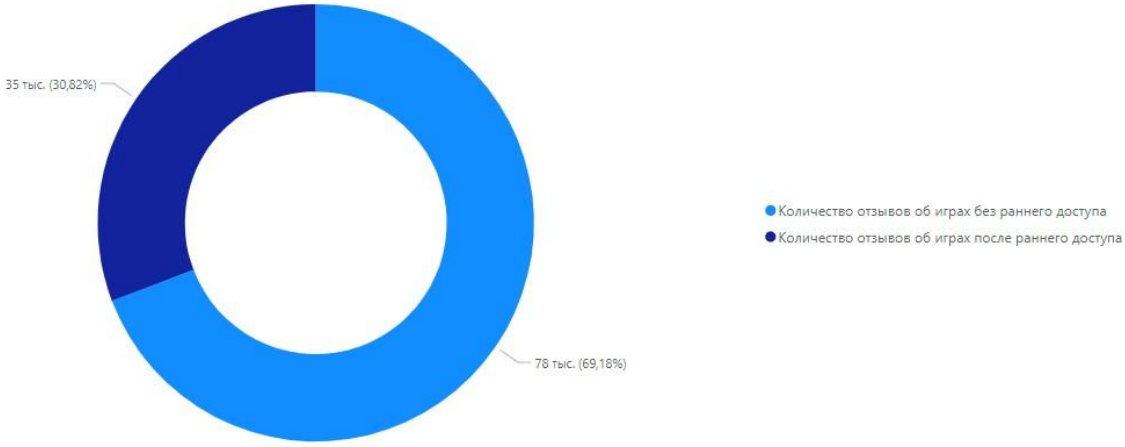
Последняя выборка:
Было принято решение для получение максимально достоверных результатов провести сравнение двухсот игр максимальным количеством отзывов.

```
steamSessionGroupGamesWithEAAfterTop100DF = spark.sql("""SELECT *
                            FROM steam_with_ea_after
                            LIMIT 100
                             """)
steamSessionGroupGamesWithEAAfterTop100DF.printSchema()
steamSessionGroupGamesWithEAAfterTop100DF.show()

steamSessionGroupGamesWithEAAfterTop100DF.coalesce(1).write.options
    .csv("file:/home/student/Data/steam_project/steam_with_early_ac
```

```
root
 |-- game: string (nullable = true)
 |-- count_reviews: integer (nullable = true)
 |-- avg_rating: double (nullable = true)

+--------------------+-------------+--------------------+
|                game|count_reviews|          avg_rating|
+--------------------+-------------+--------------------+
| PUBG: BATTLEGROUNDS|         1946|0.45477903391572455|
|    Wallpaper Engine|         1422| 0.9711673699015471|
|          The Forest|         1413| 0.9922151450813871|
|                Rust|         1171| 0.8684884713919727|
|          Subnautica|         1004|  0.999003984063745|
|            RimWorld|          942| 0.9989384288747346|
|ARK: Survival Evo...|          937|  0.855923159018143|
|Don't Starve Toge...|          924| 0.9989177489177489|
|              Mirror|          816|                1.0|
|                DayZ|          808| 0.7673267326732673|
|              Kenshi|          733| 0.9959072305593452|
|Divinity: Origina...|          710| 0.9985915492957746|
|       The Long Dark|          613| 0.9755301794453507|
|               Squad|          607|  0.914332784184514|
|            Unturned|          551|  0.956442831215971|
|  Deep Rock Galactic|          546| 0.9945054945054945|
|    Darkest Dungeon®|          528| 0.9715909090909091|
|               Hades|          506| 0.9960474308300395|
|      Hunt: Showdown|          502| 0.8924302788844621|
|          Green Hell|          456| 0.9736842105263158|
+--------------------+-------------+--------------------+
only showing top 20 rows
```

```
steamSessionGroupGamesWithoutEATop100DF = spark.sql("""SELECT *
                            FROM steam_without_ea
                            LIMIT 100
                             """)
steamSessionGroupGamesWithoutEATop100DF.printSchema()
steamSessionGroupGamesWithoutEATop100DF.show()

steamSessionGroupGamesWithoutEATop100DF.coalesce(1).write.optio
    .csv("file:/home/student/Data/steam_project/steam_without_e
```

```
root
 |-- game: string (nullable = true)
 |-- count_reviews: integer (nullable = true)
 |-- avg_rating: double (nullable = true)

+--------------------+-------------+------------------+
|                game|count_reviews|        avg_rating|
+--------------------+-------------+------------------+
|    Counter-Strike 2|         2754|0.8591140159767611|
|      Stardew Valley|         2078|0.9980750721847931|
|             Terraria|         1800|0.9938888888888889|
|Tom Clancy's Rain...|         1755|0.9002849002849003|
|  Grand Theft Auto V|         1754|0.6995438996579247|
|The Witcher 3: Wi...|         1664|0.9927884615384616|
|Red Dead Redempti...|         1414|0.9356435643564357|
|         Garry's Mod|         1382|0.9971056439942113|
|       Left 4 Dead 2|         1243|               1.0|
|Euro Truck Simula...|         1215|0.9975308641975309|
|             PAYDAY 2|         1185|0.9181434599156119|
|             Warframe|         1174|0.9011925042589438|
|      Cyberpunk 2077|         1165|0.8523605150214593|
|The Elder Scrolls...|         1149|0.9103568320278503|
|        No Man's Sky|         1114|  0.86983842010772|
|    Dead by Daylight|         1091|0.8331805682859762|
|           Undertale|         1069|0.9962581852198317|
|Sekiro™: Shadows ...|         1064|0.9830827067669173|
|Sid Meier's Civil...|         1061|0.7445805843543827|
|Doki Doki Literat...|         1017|0.9970501474926253|
+--------------------+-------------+------------------+
only showing top 20 rows
```



Сравнение топ 100:

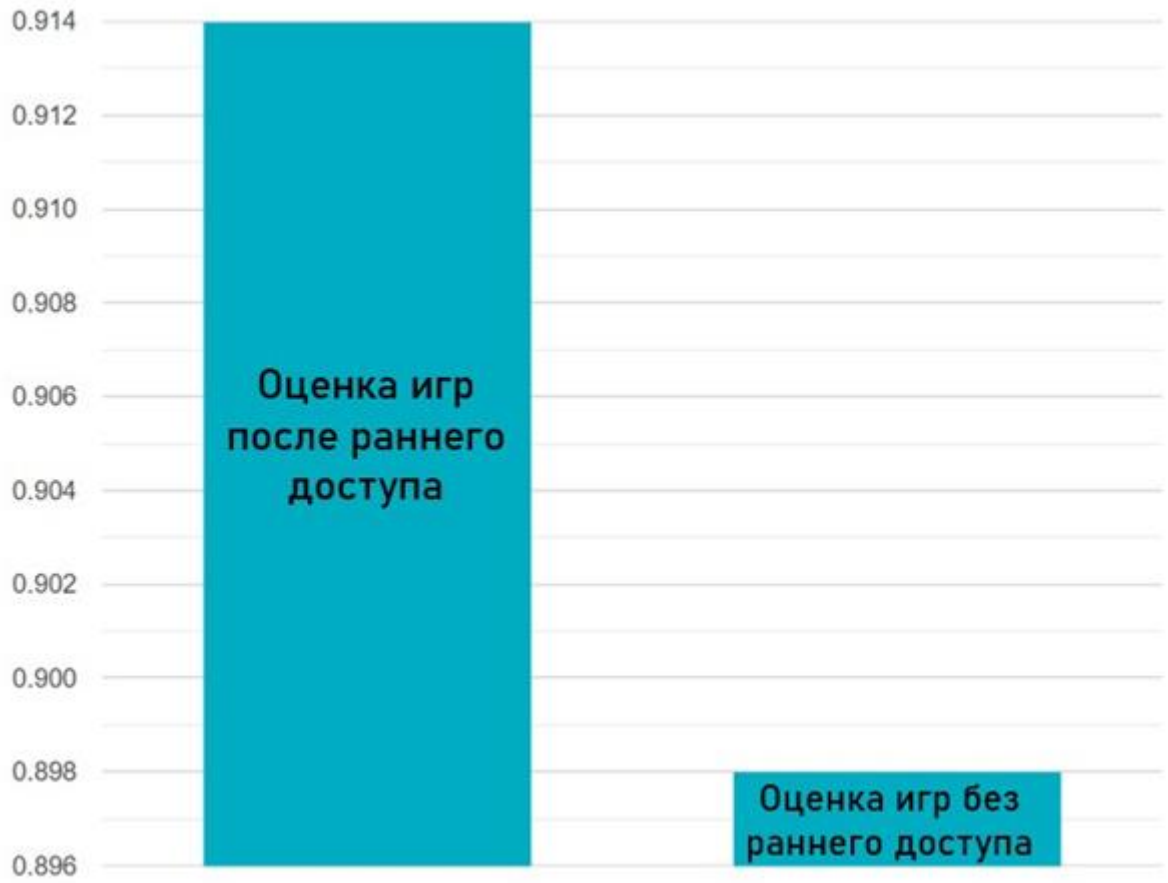
Сравниваем средние оценки полученных датасетов:

```
steamAvgRatingWithEAAfterTop100DF = spark.sql("""SELECT avg(avg_rating)
                            FROM steam_with_ea_after100
                             """)
steamAvgRatingWithEAAfterTop100DF.printSchema()
steamAvgRatingWithEAAfterTop100DF.show()
```

```
root
 |-- avg(avg_rating): double (nullable = true)

+------------------+
|   avg(avg_rating)|
+------------------+
|0.9093949622389083|
+------------------+
```

```
steamAvgRatingWithoutEATop100DF = spark.sql("""SELECT avg(avg_rating)
                            FROM steam_without_ea100
                             """)
steamAvgRatingWithoutEATop100DF.printSchema()
steamAvgRatingWithoutEATop100DF.show()
```

```
root
 |-- avg(avg_rating): double (nullable = true)

+----------------+
| avg(avg_rating)|
+----------------+
|0.89668220162413|
+----------------+
```



Сравнение вновь нам показало, игры, прошедшие через ранний доступ, имеют чуть более высокие оценки, по сравнению с играми без раннего доступа.

Заключение

Ранний доступ является мощным инструментом для разработчиков игр, позволяющим собрать финансовую поддержку, получить важную обратную связь и сформировать сообщество.
Однако успех раннего доступа зависит от способности разработчиков эффективно управлять процессом разработки и взаимодействия с сообществом.
Во всех проведенных ранее исследованиях игры с ранним доступом показали более высокое одобрение геймерской аудитории. Из этого следует вывод, что ранний доступ, точно оказал не последние влияние на общие показатели симпатии, и гипотеза заявленная ранее было доказана.