Cross-Sectional Study of 57 Thousand Elephant Learning Students

(Published July 4, 2022; Updated Oct 17, 2022) Dr. Aditya Nagrath PhD

Abstract

We examine the effectiveness of the Elephant Learning Platform for teaching mathematics. From 2017-2022, Elephant Learning has had over 140,000 students come through their proprietary math learning system. We analyzed the data within the database to get a large sample of students to determine effectiveness.

The purpose of this study is to evaluate the Elephant Learning Guarantee of a student learning 1 year of mathematics over the course of 3 months using the system 30 minutes per week. In this study, we examine all the student data from January 2017 to June 30 2022.

Methodologies

Students were provided with access to Elephant Learning's mathematics system via their parents. The students then used Elephant Learning naturally over different periods of time, for varying time durations and length of trial. Trials were administered by the parent or were self driven by the student. Coaching was provided for the parents to help students overcome any struggles.

Coaching consists of teacher training videos and coaching around mathematics anxiety. Teacher training videos are presented at a subject level based on the student's Elephant Age™ and open subjects.

Students older than five years of age start within our placement exam to determine their initial level of understanding. Students younger than five are placed based on age. Elephant Learning's system asks parents upon entry to confirm the placement exam for each student, allowing the parent to adjust the given exam based on their knowledge of the student's understanding.

The default settings conduct a placement based on the student's age and is intended to start approximately 2 years behind the student's current level in order to catch up and identify gaps. Students are presented puzzles that show proficiency in common core and state standards for conceptual mathematics via a number of subjects (for example: "Addition and Subtraction To 20"). If a student demonstrates understanding in the subject matter, more challenging subject matter is presented also within examination mode. Otherwise, the subject matter is assigned to the student.

Once all subjects are assigned, the system goes into learning mode. Elephant Learning's proprietary system presents puzzles in an optimal order for facilitating student conceptual understanding. At that time, an initial Elephant Age ™ is assigned to the student and progress is measured from there.

The Elephant Age™ represents, approximately, the average age that a student outside of the system is doing the same mathematics as a student within our system. The Elephant Age™ is the average of the ages assigned to five recently passed milestones with the largest ages assigned to them. The milestones are associated with line items within the standards (common core) and the ages assigned to the milestones were calculated based on those standards. As students use the system and milestones are completed, the Elephant Age™ is updated live and a graph is displayed for the student.

Data Collection Methodology

We are analyzing the history table in Elephant Learning. Upon answering a question within the Elephant Learning system, a history record is created within the database. The history record contains:

- exam: a flag indicating whether the history item was created during the placement exam
- event_type: an internal indicator of the type of the history item (pass/fail/milestone pass/subject pass)
- elephant_age: the current elephant age of the student at the time of passing the question. If passing a question increased the student's Elephant Age™, it would not be recognized until the next history item of the same type.
- duration: The amount of time a student spent on a particular question in seconds
- play_age: The number of weeks the student has been in the system at the time of this history item.

We examined the history table in a mongo database that was a backup of the live data. There were 69,097,889 records within the analysis. The following aggregate pipeline was determined as optimal for this study.

```
// Pipeline
[
    // Stage 1
    {
        $match: {
            exam: false,
            event_type: {$in:[-1,1]},
            elephant_age: {$gt: 0},
            play_age: {$lte: 20},
        }
    },
    // Stage 2
```

```
    {
      $group: {
        _id: "$studentId",
        time: { $sum: "$duration"},
        minea: {$min: "$elephant_age"},
        maxea: {$max: "$elephant_age"},
        maxplayage: {$max: "$play_age"}
      }
    },
    {
      $match: {
        time: {$gte: 3600},
        maxplayage: {$gt: 5}
      }
    },

    // Stage 4
    {
      $project: {
        time: 1,
        minea: 1,
        maxea: 1,
        maxplayage: 1,
        avg_time: {$divide: [{$divide: ["$time", "$maxplayage"]}, 60]},
        ea_age_change: {$divide: [{$subtract: ["$maxea", "$minea"]},12]}
      }
    },

    // Stage 5
    {
      $group: {
        _id: null,
        st_count: {$sum: 1},
        average_weeks: {$avg: "$maxplayage"},
        average_time: {$avg: "$avg_time"},
        average_ea_gain: {$avg: "$ea_age_change"}
      }
    }
  ]
```

Stage 1: The match operator is a filter on the table.

- We filter the history table for items that are not calculated within exam mode.
- The Elephant Learning system calculates the Elephant Age™ live for each student as they participate in the individual activities including during the placement exam. For this

query, we only wish to look at pass and fail activities so that we can get an accurate sum of the time duration and so we only want event_ types of 1 and -1.
- Upon examining the data, there were data points we had noticed that were between the placement exam exit and the starting elephant age where the student had not passed any milestones during the placement exam. In this case, the elephant_age in the history item was set to 0. We filter these records out so that we may calculate the minimum elephant_age and maximum elephant_age for each student in upcoming pipeline items.
- play_age filter: We only want to look at the first 20 weeks of play. Many students stay with Elephant Learning, and many students may leave and come back a year later, skewing the average play_age up once calculated. The play_age is calculated based on the creation timestamp of the student.

Stage 2: All of the data points from stage 1 are fed into a group by which groups by the studentId, allowing us to now aggregate the data per student. We have the database calculate:
- time: a sum of all the durations for the surviving history records per student. This is the total time played within the first 20 weeks.
- minea: the minimum elephant_age value per student on the surviving history records.
- maxea: the maximum elephant_age value per student on the surviving history records.
- maxplayage: the maximum play_age of the surviving history records. This is roughly the number of weeks of play.

Stage 3: Usage filter. We do not want to examine students that did not use the system for the minimum usage requirements. In this match stage, we are looking for a total play of 1 hour, and at least 5 weeks of play. This filters out students that have not played at least 5 weeks, and have not put in at least 1 hour of play within that time.

Stage 4: This projection allows us to calculate the difference between the maxea and minea. It also allows us to calculate the number of minutes (averaged over play_age weeks) the student used the system per week.

Stage 5: Of the resulting records, and based on the previous projection we calculate the number of students, the average play_age or number of weeks the students were in the system, the average time duration students played per week, and the average difference in elephant_age from start to finish.

Mongo passes the results from each stage to the next stage, transforming the data using the operations provided above. The resulting data would be an analysis of the 67 million records within the database.

At the end of Stage 4 in the above pipeline, we have a row for each student that has used the system at least 1 hour (3600 seconds) in total and been within the system for at least 5 weeks. Because week 1 is play_age 0, a student would have had to have played on the sixth week or beyond to be included.

Stage 5 actually crunches the data to get the results, it goes over each row in Stage 4 (each student) and averages the number of weeks, the weekly average usage duration, the average improvement, and provides a count of the number of rows so that we understand how many students made it through the initial filter.

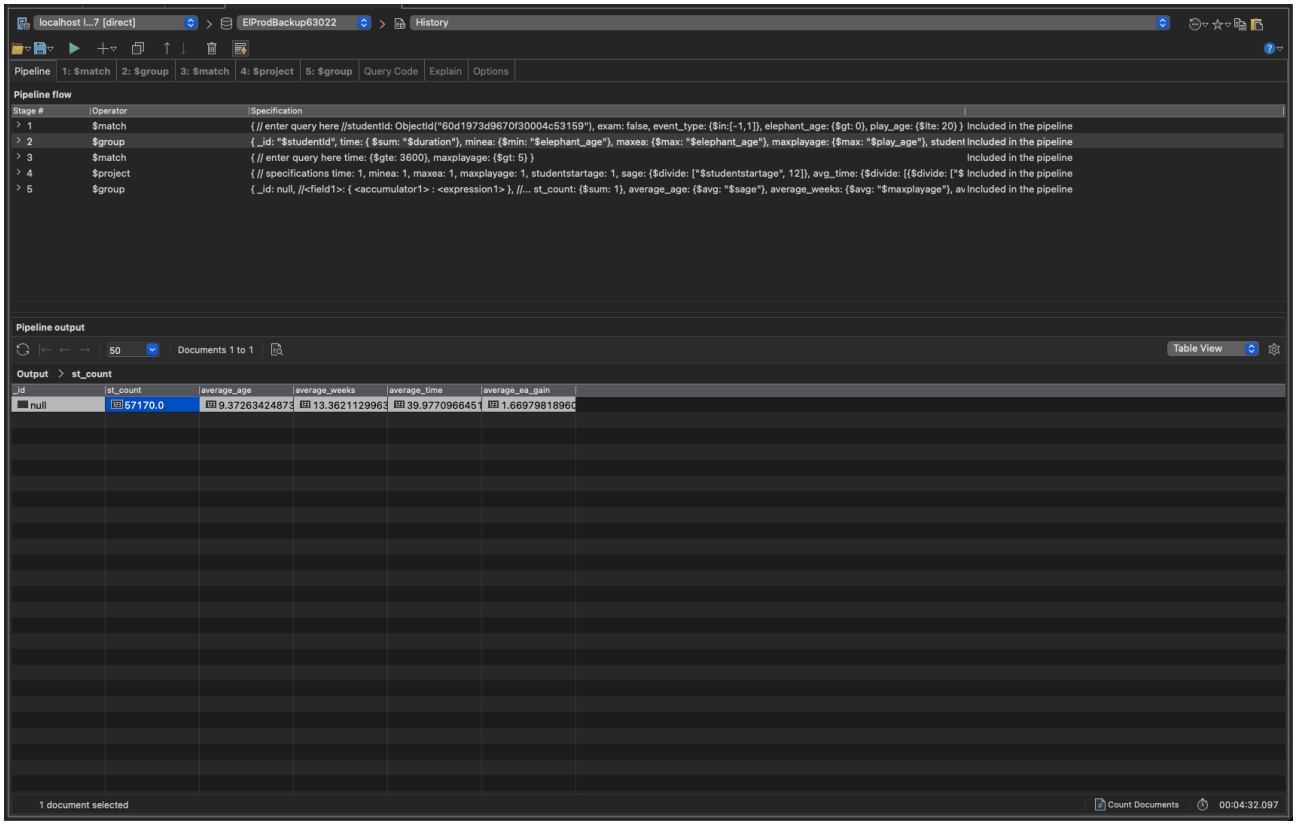
Analysis Considerations

Beyond non-usage there are several known challenges that students may have faced within the data. Prior to 2020, it was possible for a student to miss every question in a placement exam, indicating that the starting point of the placement exam should have been lowered. The student would be able to play, but generally does not progress as the initial setting was incorrect. We have manually recovered students via customer success representatives, but because we are only looking at the first 20 weeks of usage, their recovery would not be represented within the data. After 2019, we added capabilities to notify the parent when we detected student struggle in this manner so that they may manually take action. After 2020, our system was upgraded to detect many of these scenarios and automatically place the student in an earlier placement exam.

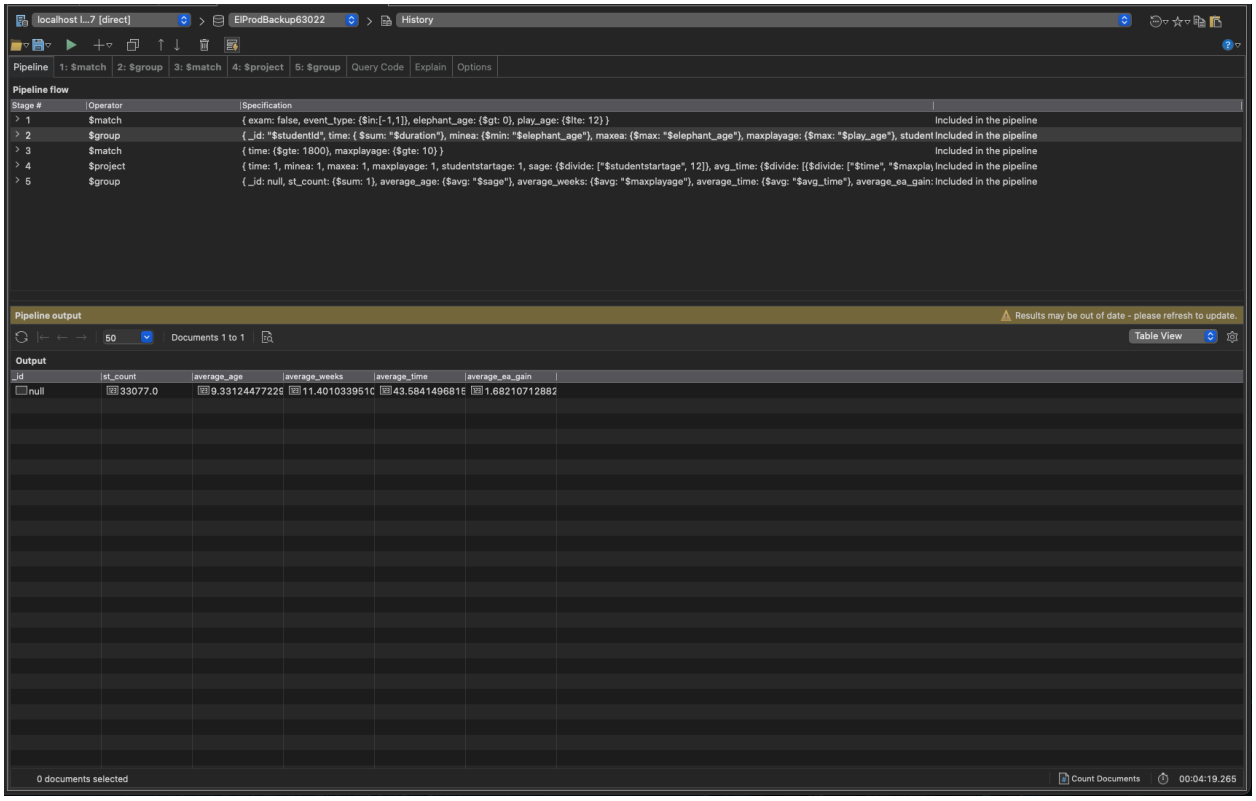
In 2017, our curriculum was mainly aimed at pre-k, kindergarten, and early elementary students. In 2018, we added fractions, decimals and percentages and in 2019 we added Algebra. In 2020, we upgraded our Algebra curriculum and in 2021 we have begun adding on top of the Algebra curriculum with courses such as personal finance. Because we are analyzing all the data collected since 2017, the average age of the student versus earlier studies should have increased. The incident of positive results also should also increase due to alterations to the system making it more effective.

The Results

57170 students learned, on average, 1.67 years of mathematics. Average time in the system was 13 weeks using the system just under 40 minutes per week. The average age of the student was 9.37 years of age. The curriculum covers counting through Algebra and we have accepted students of all ages.



We then decided to narrow down on the first 3 months of usage, so we changed the initial filter to only look at the first 12 weeks of data, and asked for only students that played for at least 10 weeks.



As a result, we narrowed it to 33,077 students, the average age about the same at 9.3. Average weeks in the system are as expected at 11 weeks. The average time using the system is actually slightly higher at 43 minutes per week. The average gain was maintained at 1.68 years of mathematics learned.

Breakdown by Age

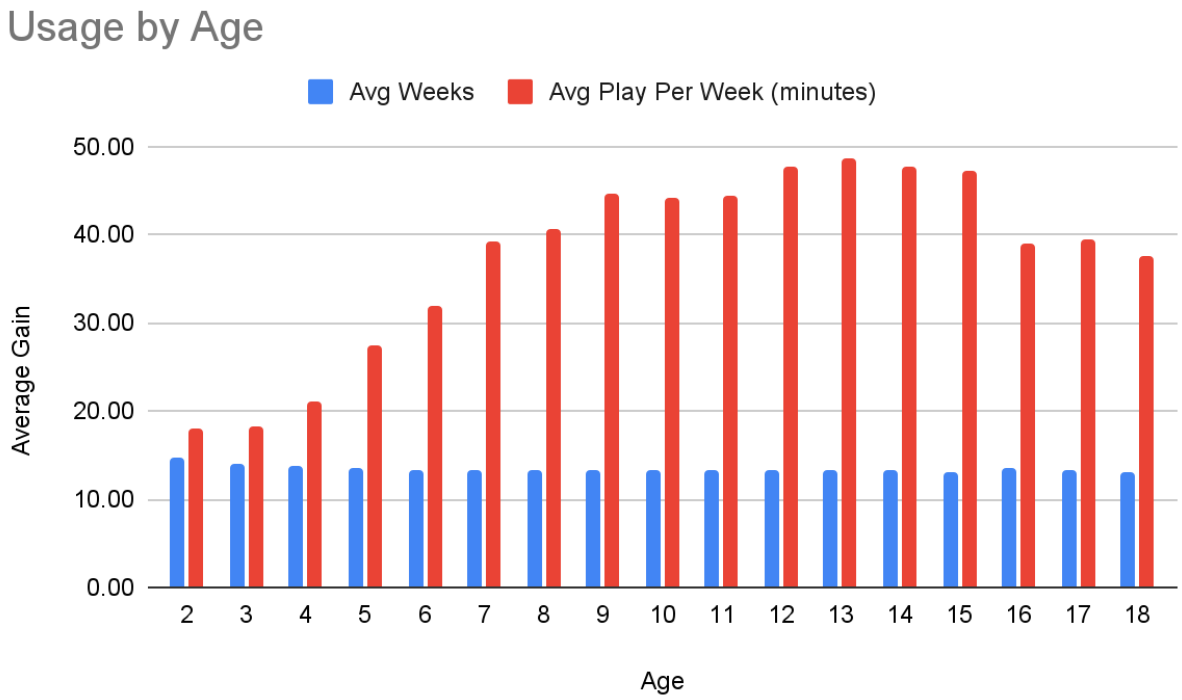
An alteration to the above query at stage 5 would allow us to breakdown the data by the age of the student.

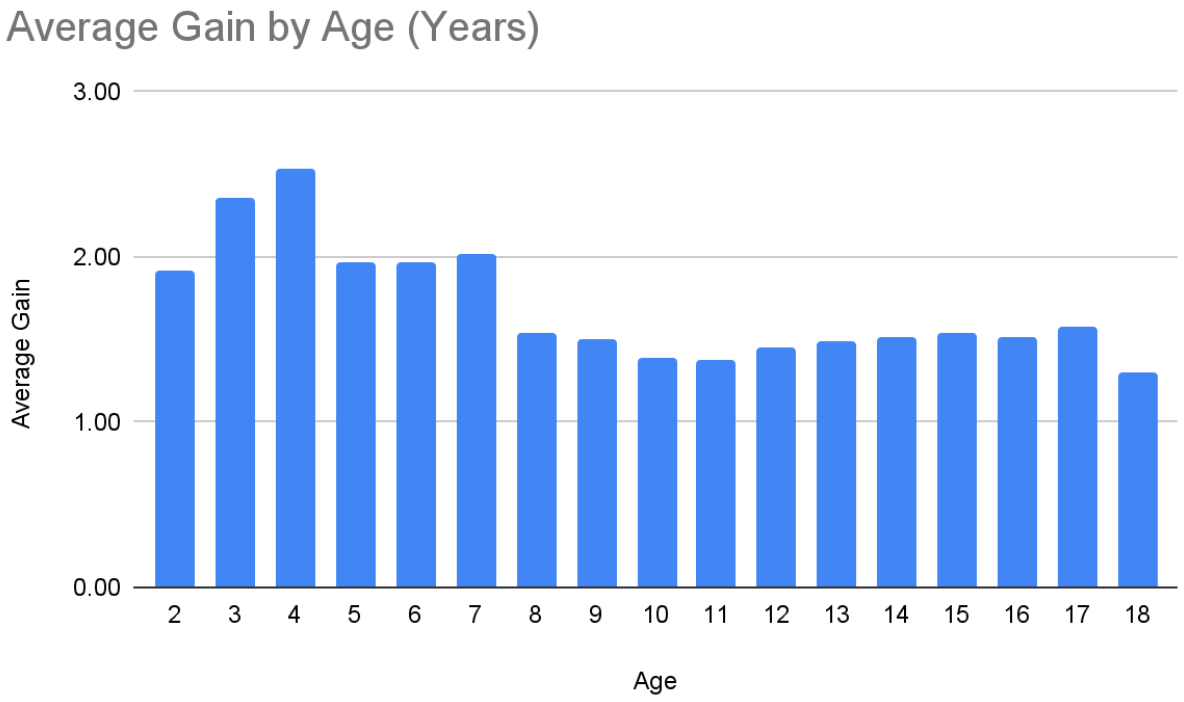
Stage 5:

```
$group: {
        _id: {$toInt: "$sage"},
        st_count: {$sum: 1},
        average_age: {$avg: "$sage"},
        average_weeks: {$avg: "$maxplayage"},
        average_time: {$avg: "$avg_time"},
        average_ea_gain: {$avg: "$ea_age_change"}
    }
```

The difference between this stage 5 and the previous stage 5 happens on the _id field, which we ask it to group by the integer number of the starting age of the student (rounding down). So if the student was 7.9 years old, that student would be grouped into the 7 year old category.

The resulting two charts break down usage by age, and then the average gain of the students by age. We have truncated the ages by only those that contained more than 50 students. For example, it appears that 5 students started at age 0 (likely erroneously), so we have chosen to remove that data so that the displayed data only shows ages with more than 50 students. We have inserted a table with teh values so that the reader can see the number of students per each age level..





Age	Student Count	Average Age	Avg Weeks	Avg Play Per Week (minutes)	Average Gain (Years)
2	232	2.58	14.61	17.90	1.91
3	1139	3.53	13.93	18.20	2.36
4	2445	4.50	13.87	21.19	2.53
5	4126	5.45	13.51	27.56	1.97
6	5029	6.45	13.42	31.98	1.96
7	6244	7.44	13.28	39.12	2.02
8	6809	8.43	13.36	40.58	1.54
9	7051	9.43	13.26	44.72	1.50
10	6829	10.42	13.34	44.11	1.39
11	5849	11.41	13.29	44.36	1.37
12	4342	12.41	13.26	47.76	1.45
13	3048	13.41	13.30	48.76	1.49
14	1948	14.37	13.19	47.61	1.51
15	1081	15.38	13.03	47.21	1.53
16	630	16.41	13.53	39.01	1.52

17	185	17.34	13.37	39.51	1.58
18	61	18.35	12.98	37.48	1.31

Conclusions

In 2017, analyzing our initial 50 students, we had determined that students had learned an average of 1.5 years of mathematics over the course of 10 weeks (average), using the system 22.5 minutes per week. The average age of the student was 5 years of age and the curriculum at the time covered through multiplication and division.

An older average student population gives us a larger attention span allowing for more usage naturally. However, results have remained consistent in that the gain over approximately 3 months is over 1.5 years.

Elephant Learning was effective for students of all ages. Minimum gain was 1.31 years (average) for 18 years of age. Younger students tended to gain more. Children at younger age have less materials to master and avoid mathematics anxiety altogether. Oftentimes, young students' Elephant Age is ahead of their age. In this scenario, the student likely is unaware that the materials received are for older students.

Elephant Learning guarantees that students within the system will learn at least 1 year of mathematics over the course of 3 months when they use the system 10 minutes per day, 3 days per week. Given the above data and assuming a bell distribution, over 90% of students will achieve those results naturally. With human intervention either in the classroom, by the parent, or by one of our results counselors, we can drive success rates to 100% for students.