KLE Society's

KLE Technological University



**Exploratory Data Analysis**

**(22ECAC210)**

**Course Project Report on**

**“Predicting Student Performance from Game Play”**

*Submitted in partial fulfilment of the requirement for the award of*

**Degree of Bachelor of Engineering**

**in**

**Computer Science and Engineering**

*Submitted By*

**NEHA R SHANBHOG – 01FE21BCS138**

**AMRUTHA BEEDIKAR – 01FE21BCS198**

**SHRIHARI HAMPIHOLI – 01FE21BCS204**

**MAANASI SHASTRI – 01FE21BCS210**

*Submitted T0*

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING,**

KLE TECHNOLOGICAL UNIVERSITY

HUBLI – 580 031 (India).

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| CONTENT | PAGE NUMBER |
| 1] ABSTRACT | 3 |
| 2] INTRODUCTION | 4 |
| 3] PROBLEM STATEMENT | 5 |
| 4] DATASET DESCRIPTION | 5 |
| 5] DATA PRE-PROCESSING | 6 |
| 6] DATA EXPLORATION | 7 |
| 7] REFLECTING ON THE DATA PROCESSED | 10 |
| 8] A NEW JOURNEY TO STUDY THE DATA | 11 |
| 9] FEATURE ENGINEERING | 12 |
| 10] CORRELATION ANALYSIS | 14 |
| 11] MODEL TRAINING | 18 |
| 12] CONCLUSION | 22 |

**ABSTRACT**

The project "Predicting the Student Performance from Gameplay" focuses on enhancing game-based learning, an educational approach that fosters engagement and enjoyment in the learning process by integrating educational content within a gaming framework. The dataset provided contains extensive gameplay records from an educational game named **Jo Wilder and the Capitol case**, offering a valuable opportunity to apply data science and learning analytics principles to advance game-based learning practices.

The primary goal of the project is to develop knowledge tracing methods specifically tailored to educational games, enabling personalized support for individual students' learning journeys. While knowledge tracing methods have been extensively studied in online learning environments and intelligent tutoring systems, their application in educational games remains relatively unexplored, due to the lack of sufficient data-sets available.

The competition is hosted by Field Day Lab, a research lab dedicated to creating educational games that leverage contemporary research to support learning for diverse age groups. Their commitment to accessibility ensures that their games are freely available to all, fostering broader access to game-based learning platforms.

The successful implementation of this project has the potential to empower game developers with insights into how students learn through gameplay, leading to the improvement of educational games. Furthermore, the development of dashboards and analytic tools will aid educators in providing personalized support to students' learning experiences.

Overall, the project's outcomes have significant implications for advancing the integration of game-based learning in educational settings and fostering broader support for innovative and engaging learning platforms.

**INTRODUCTION**

Exploratory Data Analysis is a fundamental step in the data analysis process, essential for gaining insights, identifying patterns, and understanding the underlying structure of a dataset, and further making inferences that can have a huge impact on the society. In this project, we embark on a comprehensive EDA journey, focusing on the dataset titled "Predict the Student Performance from Gameplay."

The importance of EDA lies in its ability to **unveil hidden trends** and **relationships** within data, enabling data scientists to make **informed decisions** and **guide subsequent analyses** effectively. By delving into the dataset provided by Field Day Lab, a research lab committed to enhancing game-based learning, we aim to unravel the potential of game data in advancing educational practices.

Our case study's objectives encompass several key facets. Firstly, we seek to understand the dynamics of game-based learning, where educational content is intertwined with interactive gameplay, creating engaging learning experiences.

Secondly, by applying data science and learning analytics principles, we aspire to develop knowledge tracing methods specific to educational games. These methods can facilitate individualized support for students, tailoring learning experiences based on their unique learning paths.

We also build a model which is trained on the dataset given, and which can predict the correctness of a student’s response.

Ultimately, our EDA project holds the potential to bridge the gap between game-based learning and knowledge tracing, promoting the broader adoption of game-based learning platforms in education. By merging the allure of gaming with data-driven insights, we aim to contribute to the advancement of contemporary educational practices and enrich the learning experiences of students worldwide.

**PROBLEM STATEMENT**: Given a dataset encompassing the data of many students, we should analyze, draw inferences and construct a model Predicting the Student Performance in a Game, in real-time.

**DATASET DESCRIPTION & COLLECTION**

* **Description of the Data**
  + Link to the Dataset: [here](https://www.kaggle.com/competitions/predict-student-performance-from-game-play/overview)
  + Number of Attributes given in the *Training* Dataset: 20
  + Number of Rows given in the *Training* Dataset: 26296946
* **Attribute List**

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| session\_id | The ID of the session the event took place in |
| index | The index of the event for the session |
| elapsed\_time | How much time has passed (in milliseconds) between the start of the session and when the event was recorded |
| event\_name | The name of the event type |
| name | The event name (e.g., identifies whether a notebook\_click is opening or closing the notebook) |
| level | The level of the game where the event occurred (0 to 22) |
| page | The page number of the event (only for notebook-related events) |
| room\_coor\_x | The coordinates of the click in reference to the in-game room (only for click events) |
| room\_coor\_y | The coordinates of the click in reference to the in-game room (only for click events) |
| screen\_coor\_x | The coordinates of the click in reference to the player’s screen (only for click events) |
| screen\_coor\_y | The coordinates of the click in reference to the player’s screen (only for click events) |
| hover\_duration | How long (in milliseconds) the hover happened for (only for hover events) |
| text | The text the player sees during this event |
| fqid | The fully qualified ID of the event |
| room\_fqid | The fully qualified ID of the room where the event took place |
| text\_fqid | The fully qualified ID of the text |
| fullscreen | Whether the player is in fullscreen mode |
| hq | Whether the game is in high-quality |
| music | Whether the game music is on or off |
| level\_group | Which group of levels - and group of questions - this row belongs to (0-4, 5-12, 13-22) |

The source for the above data is from the link to the Competition posted above.

**Data Pre-Processing**

**Null values**  
There are enormous NULL values in our dataset. Consider the below table attached:

|  |  |
| --- | --- |
| **Column Name** | **Missing Values** |
| session\_id | 0 |
| Index | 0 |
| elapsed\_time | 0 |
| event\_name | 0 |
| Name | 0 |
| Level | 0 |
| Page | 25,732,402 |
| room\_coor\_x | 2,073,272 |
| room\_coor\_y | 2,073,272 |
| screen\_coor\_x | 2,073,272 |
| screen\_coor\_y | 2,073,272 |
| hover\_duration | 24,294,702 |
| Text | 16,679,807 |
| Fqid | 8,274,415 |
| room\_fqid | 0 |
| text\_fqid | 16,679,702 |
| Fullscreen | 0 |
| Hq | 0 |
| Music | 0 |
| level\_group | 0 |

* We have decided to fill in the NULL values with –1 to maintain uniformity throughout the whole dataset. The attributes with the NULL values specify an “event” that occurs during the game. Filling them with any viable quantity would not make sense because we will be disturbing and adding noise.   
  We will only work with the given data with NULL values because we believe that there is no additional requirement necessary, because that is how the gameplay occurs.
* There is no requirement of normalizing the data, as there is only ONE source for this data.

**DATA EXPLORATION**

* Initially, we started with looking out for “illogical” tuples that could be in the dataset. By this, what we mean is that, for any (**session\_id, index**) for a given **session\_id**, we SHOULD only have ONE pair that exists. This is because, having a duplicate would mean that there were 2 instances at the same time, which *does not* make any sense, logically speaking.  
    
  So, we looked out for such occurrences, and there were 142 sessions out of the **23562** total sessions.

**Reversed Index Phenomenon**

* As the general game progresses (for a given session\_id), the **index** should be **monotonically increasing**, which preserves the ordering nature of the events. But this property doesn't always hold true for all the records. This phenomenon is termed as the presence of “reversed\_indexing” in the dataset.  
  There are **258** sessions in total which follow the above phenomenon.  
    
  A pictorial representation is shown below for the above:

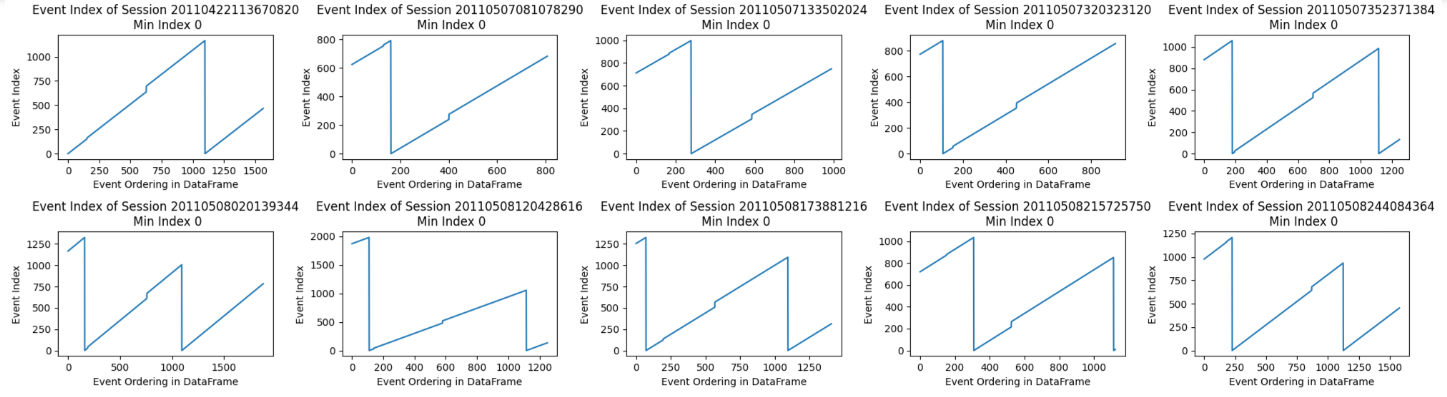


Fig 1: Reversed indexing in dataset

Only a handful of such sessions are shown above, but there are **258** of such sessions. The index *increases* up to a certain point, then *falls*, and then *increases again*. This is the “**reversed index**” phenomenon.

To address the above issue, we fitted a new column, and the results are as below in the **red line**. We have included the original “blue” indexing for comparison.

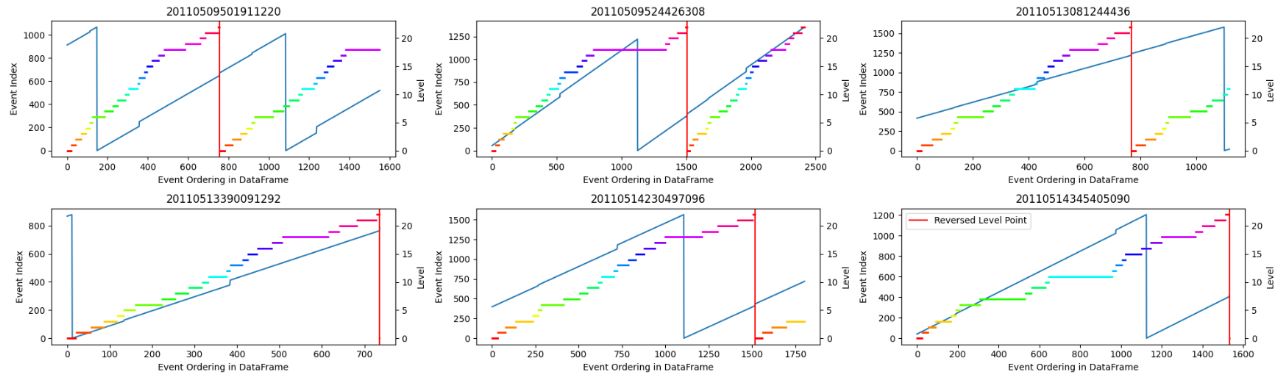


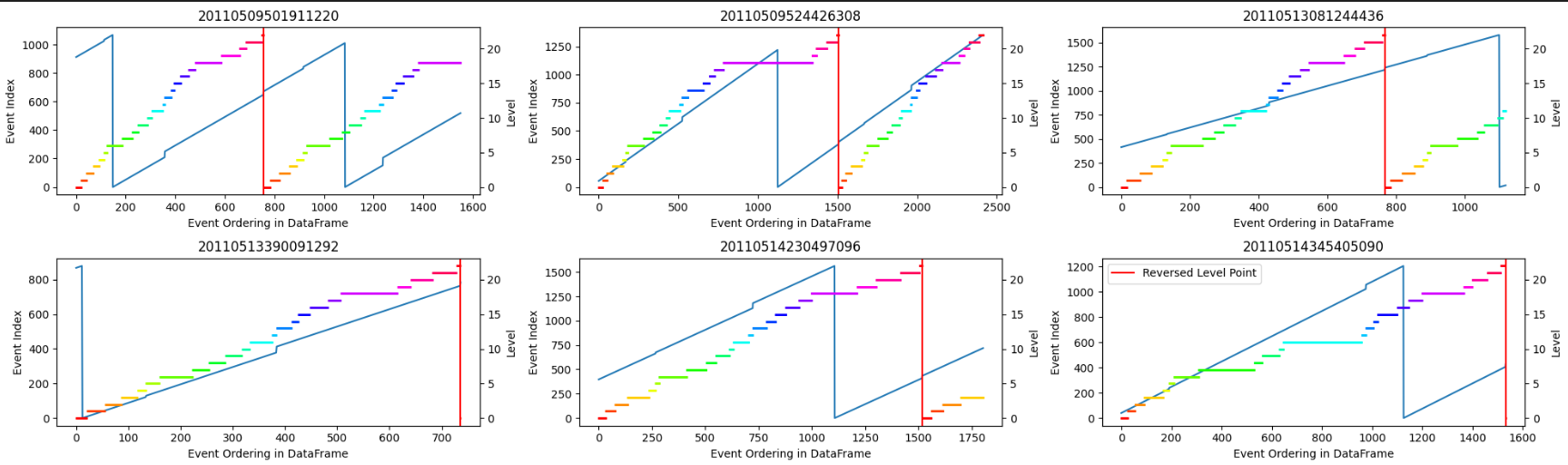
Fig 2: Solution for reversed indexing

The red line fitted is increasing linearly along with the game play of a student during a session. Although there are a few sudden jumps noticeable, it is much better than the original indexing.

**Reversed Level Phenomenon**

* There are instances where the level for a session\_id, increases for a certain time, falls suddenly, and then rises. This is known as the “**Reversed Level**” phenomenon.  
    
  To showcase what we mean, consider the below graph:

We can see this in the below figure. The horizontal-coloured lines indicate one of the 22 levels, and as the game progresses, the student’s level falls. This means that the student can play 2 games within the same session.



**Fig 3: Reversed level phenomenon**

**Analysing where the “index” jumps / drops abruptly**

* Nearly all of the jumped index phenomena occur right after checkpoint.   
  As we know that index indicates the ordering of events and jumped index also preserves the ordering nature.  
    
  The below pie chart proves the above statement:

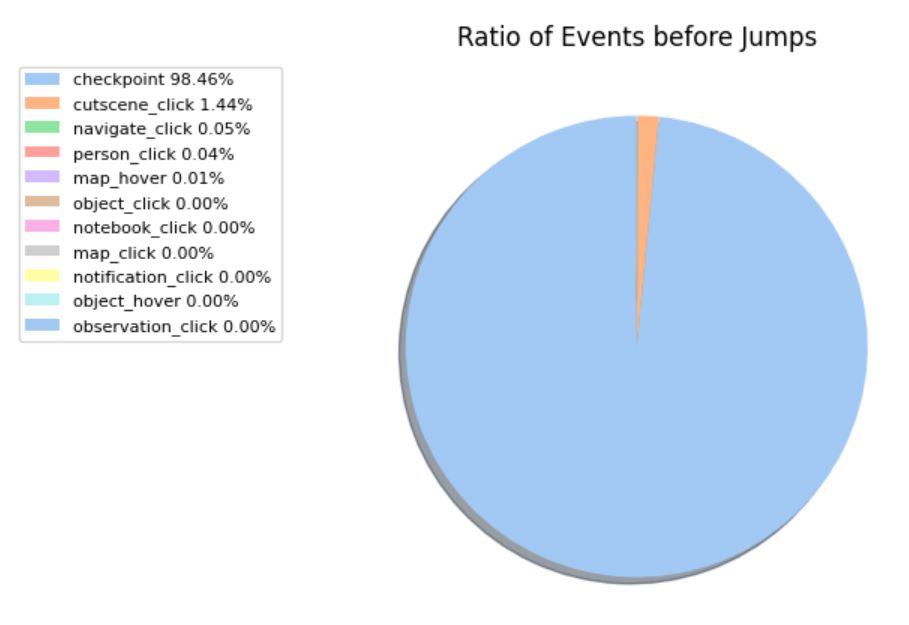


Fig 4: Jumped index phenomena

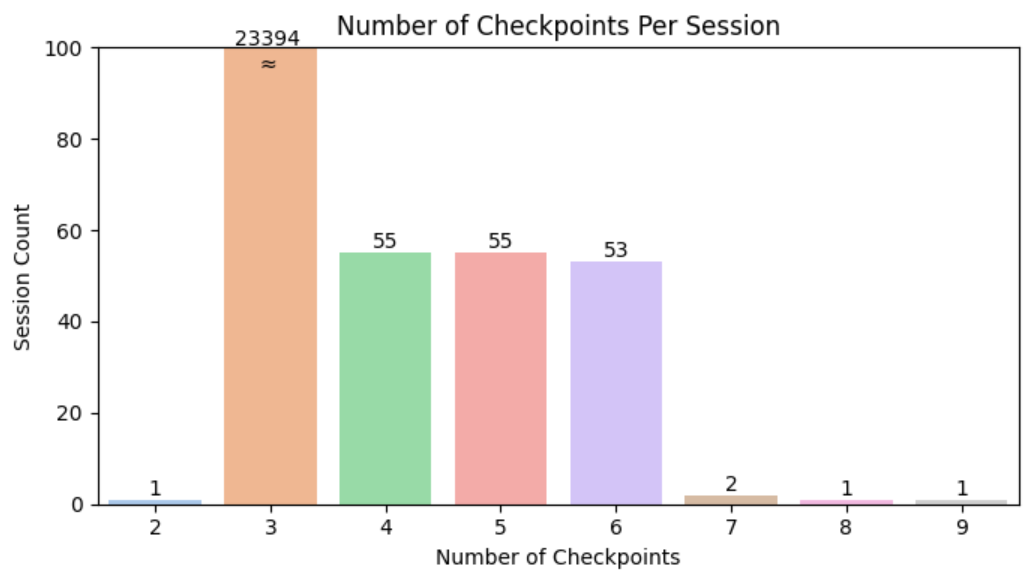
**Analysing Checkpoints**

According to the data description, each session should have 3 checkpoints.

But it DOES NOT hold true for all the sessions. The below histogram shows

that there are **168** sessions that DO NOT have 3 checkpoints, but are having

anywhere from [2, 9] except 3.



**Reflecting on the data processed**

After seeing the Exploratory Data Analysis that we have done till this point, especially the "**reversed index**", "**reversed level**" phenomenon, we thought of looking back as to how much value these analyses provided us moving forward with the model building.   
  
To explain what we mean to say, there were **258 sessions** with the **reversed index** phenomenon as we claimed and showed you before. BUT, in the whole dataset, we have **23562** sessions in total!

Also, there were only **6 instances** where the "**reversed level**" phenomenon was observed, but there were again, **23562** sessions in total.

So, we decided to think and then stop proceeding further in this direction. We further decided to try out a different approach, that we start from the next slide.

**A New Journey to Study the Data**

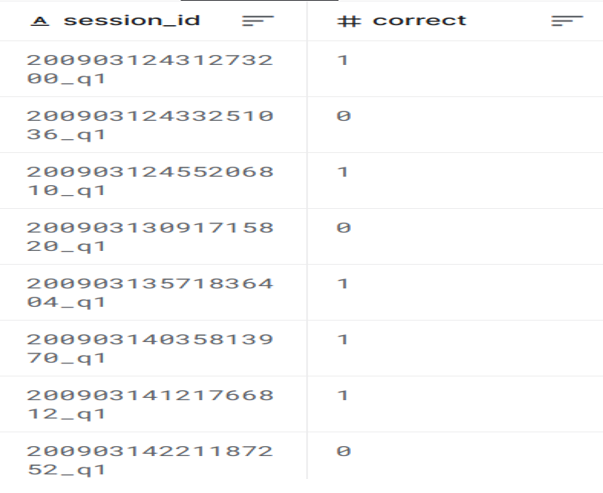
* We loaded the same "train" dataset, in a variable called "**df\_train**".
* We then loaded the labelled data given by the competition in the variable "**df\_labels**".
* The **labelled** **data** is of the form below:

Fig 5: Session ID and their answers

* Notice how the "**session\_id**" is basically a **concatenation** of the actual "**session\_id**" and the "**question\_number**"
* Now, to have a cleaner dataframe, we split the **session\_id** into the specific **session\_id**, and then make a new column called "**question\_number**" which allows us to easily proceed further.  
  Please refer below for the pictorial representation:  
  A screenshot of a computer

  Description automatically generated

Fig 6: Updates session\_id with their question number and answer

* We also reset the index of this data-frame to be indexed of the session\_id column for again, ease of further merging & processing.

**FEATURE ENGINEERING**

We now are going to show *how* we extracted the needful features that we thought would **allow for a better model training process**.

We split the attributes as "**CATS**", **"NUMS**" and "**EVENTS**", indicating "Categorical", "Numerical" and "Event Driven (textual)" attributes.



The plan was to find the unique number of each variable's instance for a specific **(session\_id, level\_group)**.

This process makes **31 features**, that are segregated, and uniquely identifiable based on the (s**ession\_id, level\_group**).

**After we do that, we have condensed the features to 31 as below:**

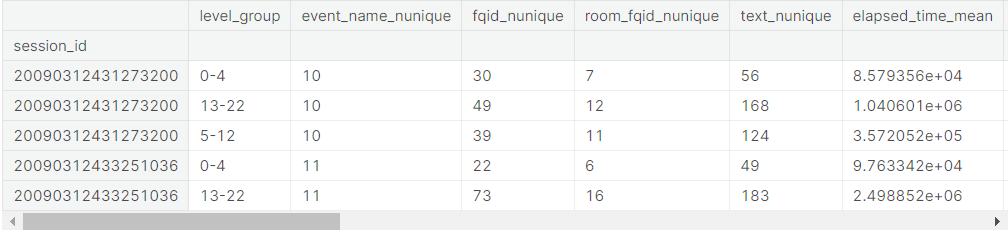


Fig 7: Session\_id with their level\_group

We now have **three** (session\_id, level\_group) pairs for each session\_id.

This helps us simplify things, in the sense that we can now train our model on the condition that:

* Questions 1 to 3 are asked during the (0-4) LG.
* Questions from 4 to 14 are asked during the (5-12) LG.
* Questions from 15 to 18 are asked during the (13-22) LG.

The above assumptions are not very robust in reasoning, but we had to assume this to be true, as we had given the game a try, we saw that this could work very well.

Now, we had finally reached a state where we could combine the labelled data and the analyzed data till now, to make a prediction model.

We now, **for every level\_group** (from the **df\_train\_featured**) and **question\_number** (from the **labelled\_df**) pair, ***merge*** the two datasets according to the previously defined rule on the questions being asked on the specific group level.

This allows us to associate each question with “**level\_group, session\_id**” and give the *correct answer*. We have saved this dataframe with the name “**df\_analysis**”.

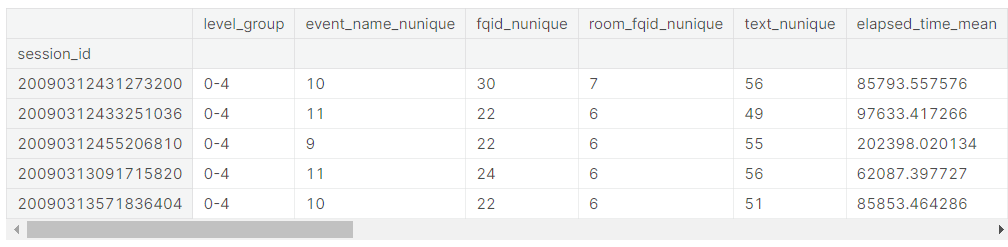


Fig 8: Session\_id of all unique sessions with level\_group

**CORRELATION ANALYSIS**

We needed to know how each attribute among the 30 we had **were dependent** on each other. So, we constructed a **Heat-Map** to allow for a visual aid to see through it. Click [here](https://vivacious-energy-2f4.notion.site/Heat-Maps-ce0c5ef5455948c3829ce3e319953673?pvs=4) for Large View.

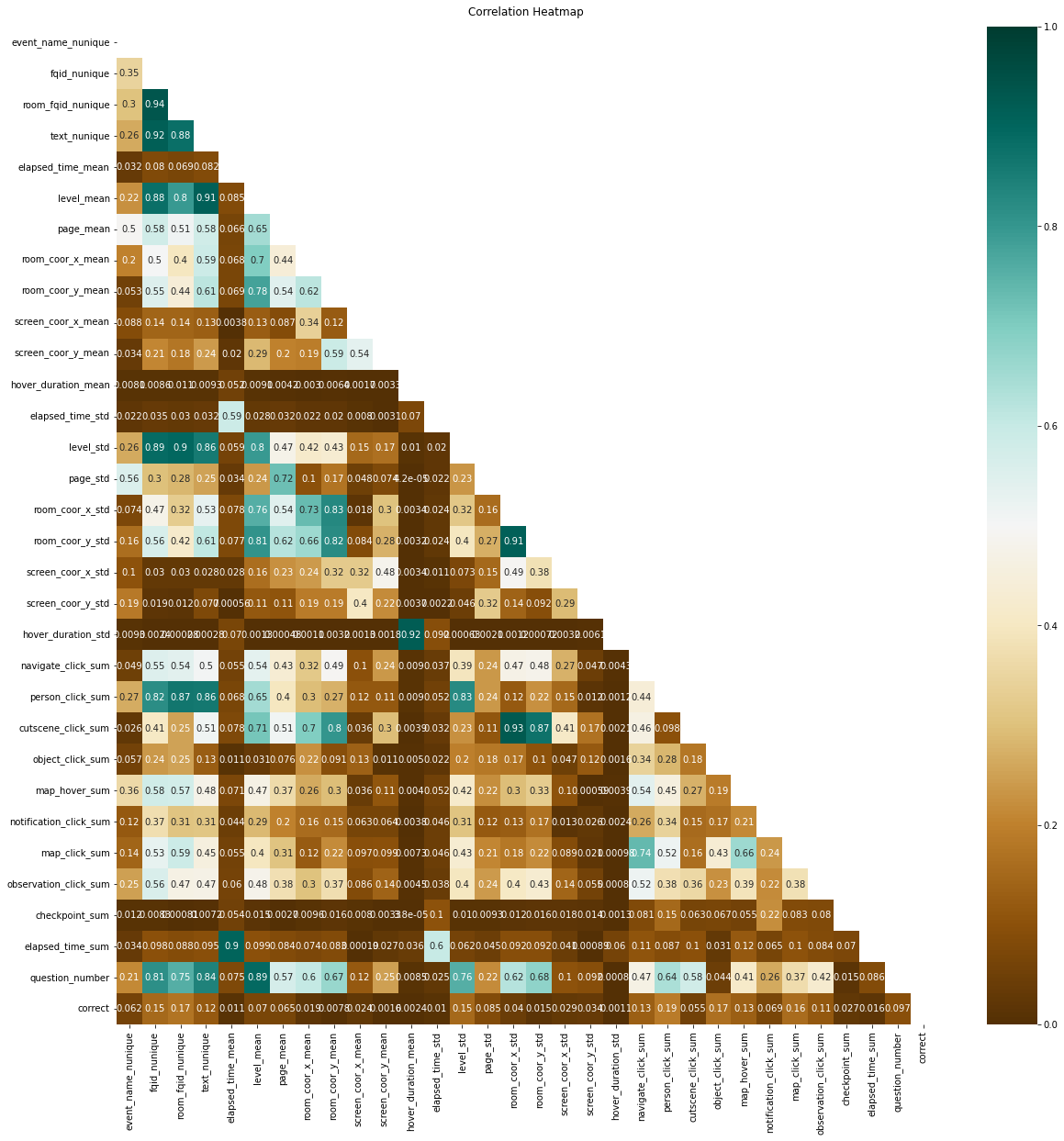


Fig 9: Correlation matrix for the dataset

The above was an **overall heat-map** for the "**df\_analysis**", representing the correlation. But we also were interested in knowing how it would differ *with-in* “**level\_group**”, so the heat-maps for those are as below.

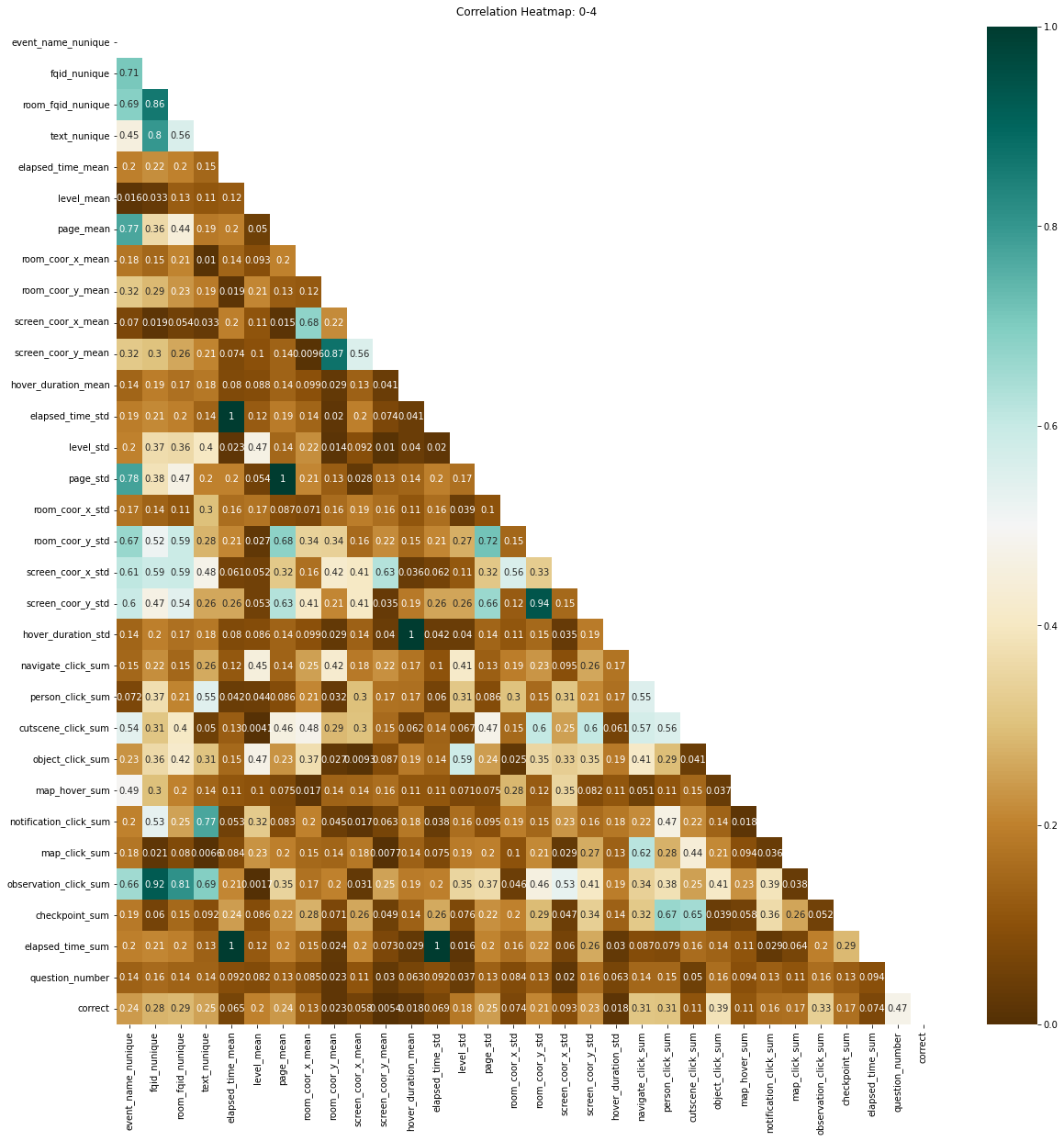


Fig 10: Correlation matrix for level group-1

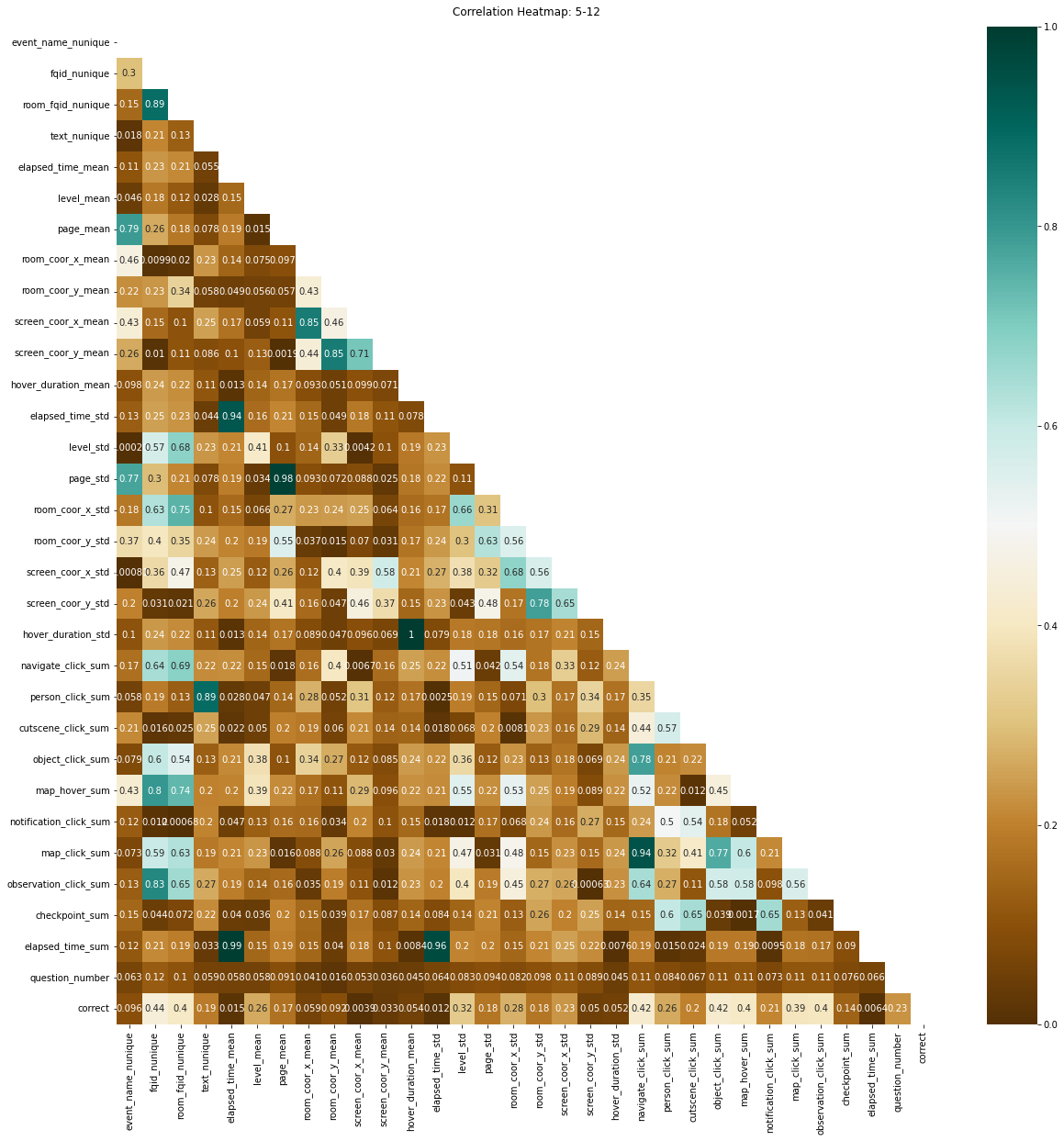


Fig 11: Correlation matrix for level group-2

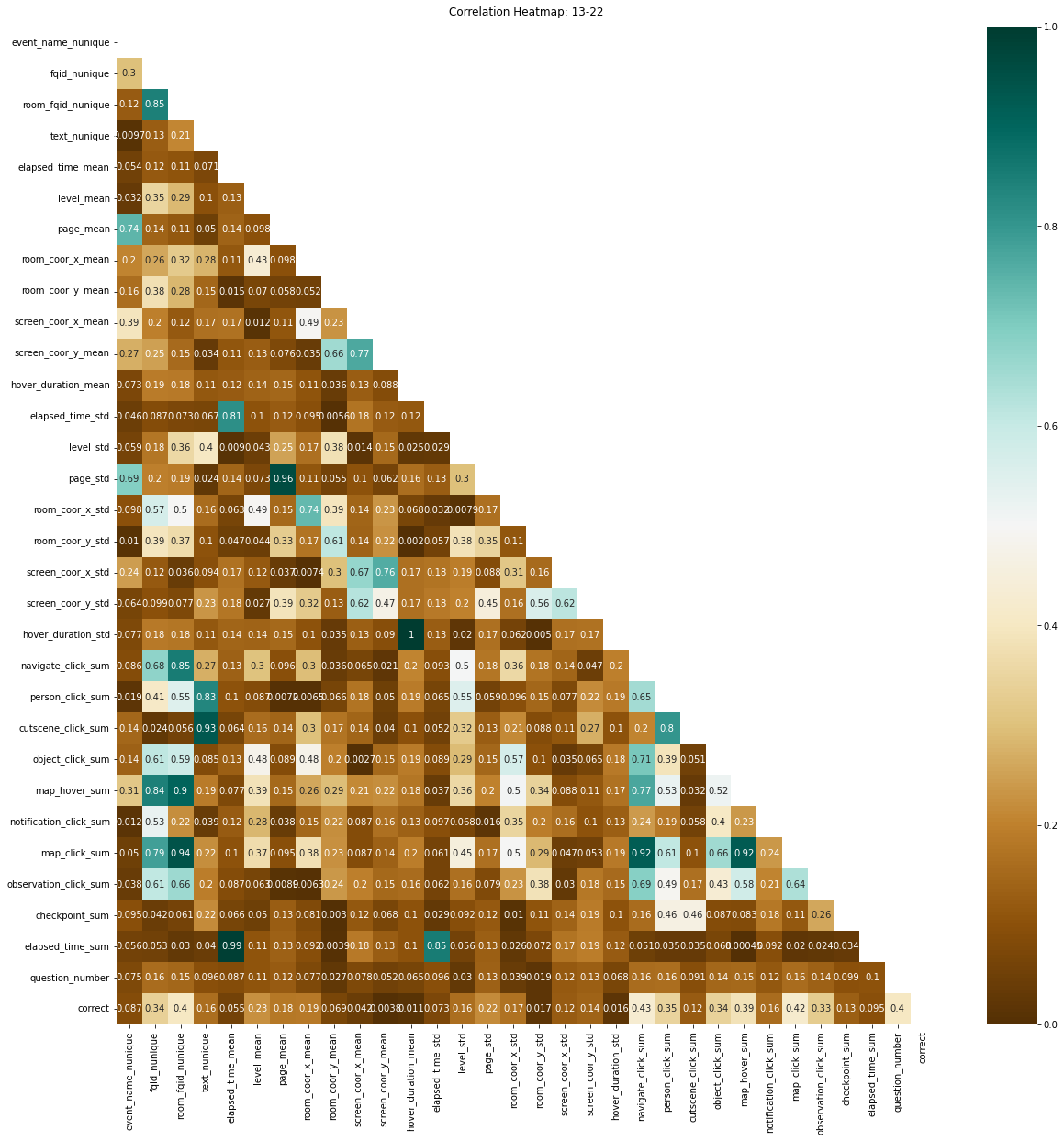


Fig 12: Correlation matrix for level group-3

**MODEL TRAINING**

As we need to predict whether the student will answer the question correctly or not, we will use **Logistic Regression** to do this, as it facilitates the **binary classification** easily.

**Q)** On what features do we train our model?

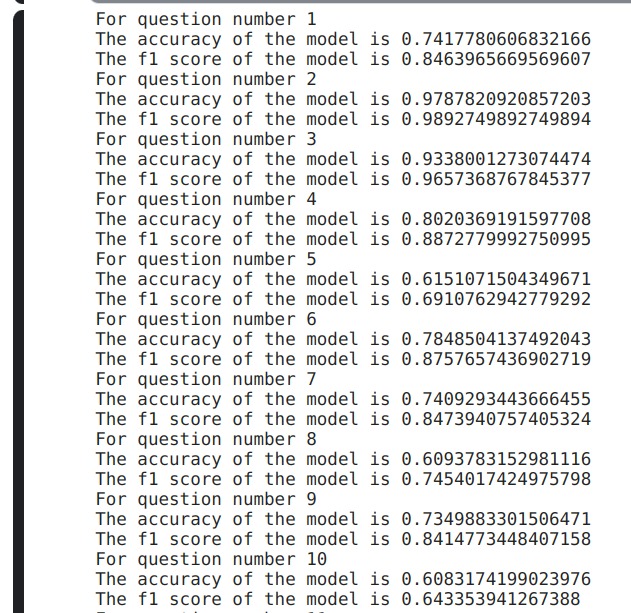
**A)** We will train it on all the **30 features** (*except* **the level\_group** and the **session\_id** as the latter is the index of the data-frame, and the former is not helpful because we have grouped accordingly already in the feature engineering).

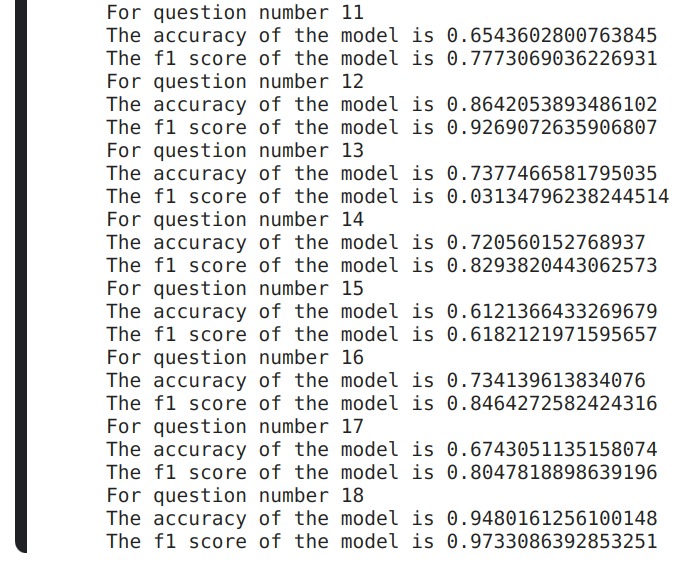
We have 23562 students on which we will train the model, because there are that many sessions.



We have opted for a **K-fold Validation Process**, where we train the model on the many available sub-sets of data. We have chosen the train sample to be 80% of the data for each sub-set. The model will be trained on the 80%, and then tested on the 20%.

After training using Logistic Regression on each question, we found that our model, on average, was able to predict 80% of the time for any given data.

The individual analysis is given below in the next page:



The **f1 score** is a very good measure for a Classification Model.

**Q) *What is f1 score?***

**A)** The **f1 score** is the **weighted harmonic mean** of *precision and recall*,

where **precision** is:

*"What percent of our predictions were correct?"*

And **recall** is:

"*What percent of the positive cases did we catch?*"

After having trained the models, we wanted to know **what** features *are important*, and *what are not* based on the Logistic Regression models that we had trained.

An example of **2 models** trained on Q1 and Q2 is shown below. The other 16 graph plots can be found in the Notebook submitted for the competition. You can find all the others in [this](https://www.kaggle.com/code/floatee/first-notebook-trail?scriptVersionId=137586549) notebook, at the end.

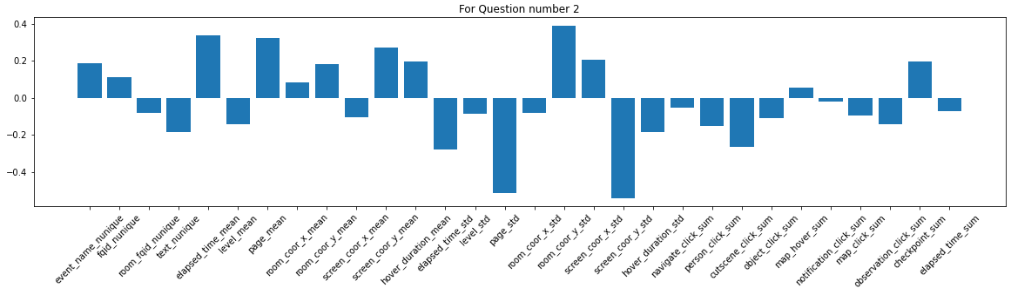
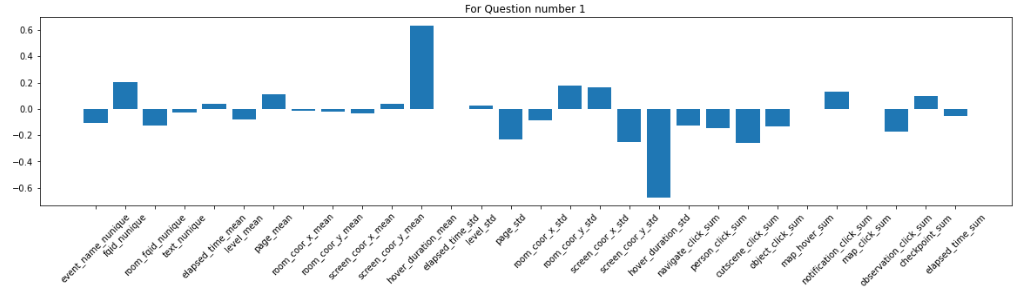
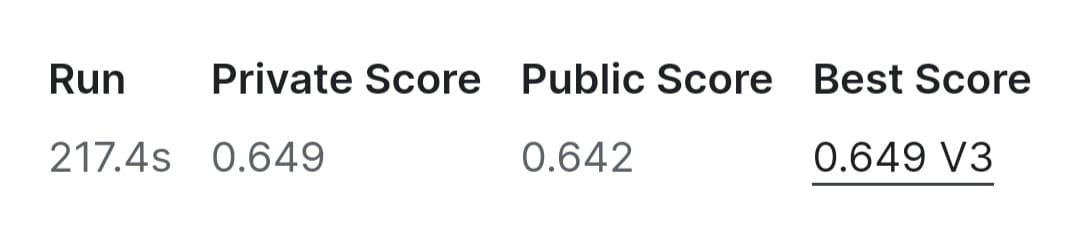


Fig 13: Effect of attributes on the question.

**CHALLENGE REPORT**

****

Submission:

https://www.kaggle.com/code/floatee/first-notebook-trail/notebook?scriptVersionId=137794814

**CONCLUSION**

As a part of this project, we were able to make logical decisions as and then necessary.

Initially when we began with the EDA process of analyzing the “***reversed index***”, “***reversed level***” and “***duplicate pairs*** of (**session\_id, index**)”, we were then thrown off-guard because we realized that it was not going to contribute majorly to the Model that we would be building later-on. But this experience taught us that we needed to look for a lot of angles before moving on, and then accordingly allocate time for each of the paths possible.

When we looked for the other alternative afterwards, during the Feature Engineering process, we were then challenged on “how” many features would be essential for training the model. But we figured this out with trial and error, as a team and finally were able to build an efficient model.

This course project has engrained in us deep insights on how to look at data, how to visualize it and then how one can choose to build an efficient model. We are grateful that we were given a Challenging Project.

Thank you.