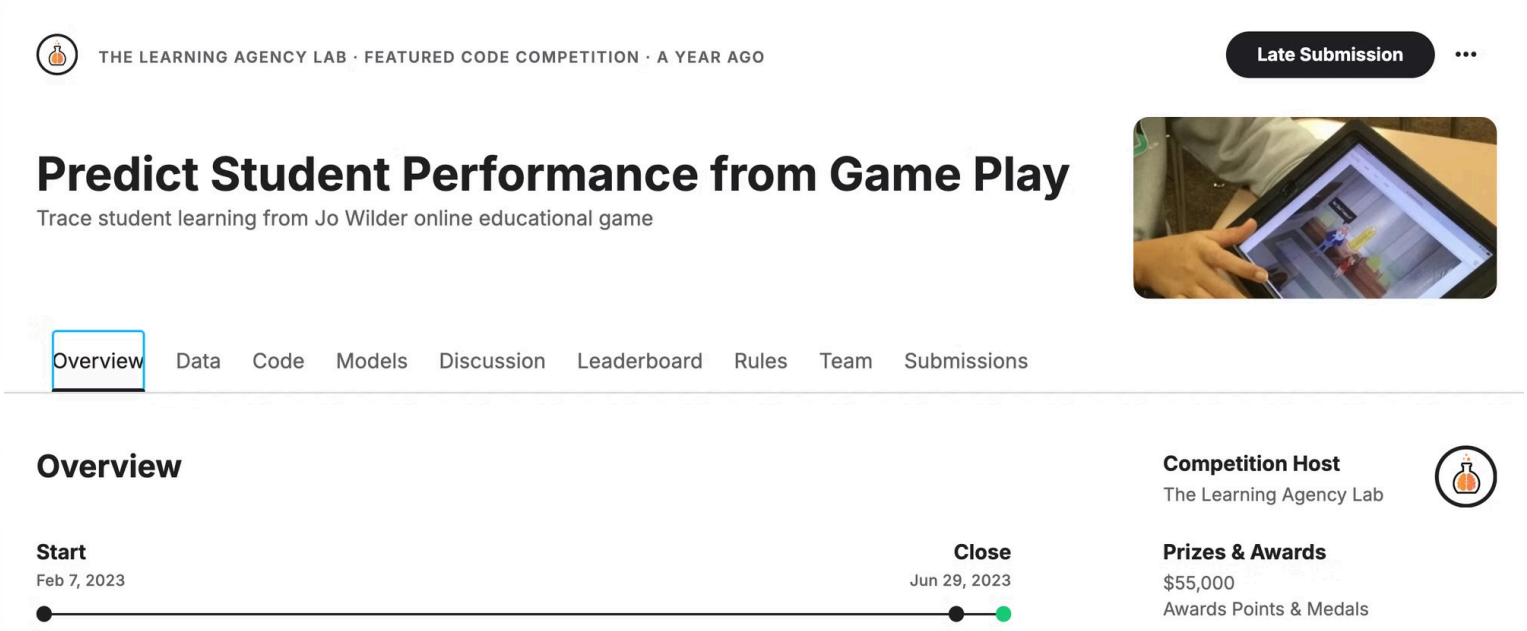
Intelligent Data Analysis

PROJECT REPORT

by 21127038 Võ Phú Hãn

Data Discovery Process



Dataset Overview

This project analyzes a Kaggle dataset on predicting student performance in educational games. The data, from Field Day Lab, includes detailed game logs. The goal was to build models predicting student performance to improve game design.

Data Included:

- Training game sessions (train.csv)
- Correct answers (train_labels.csv)
- Test game sessions (test.csv)
- Sample submission (sample_submission.csv)

Meaning of Columns

Column	Meaning
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 is opening or closing the notebook)
level	what level of the game the event occurred in (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 the event took place in
text_fqid	the fully qualified ID of the
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)

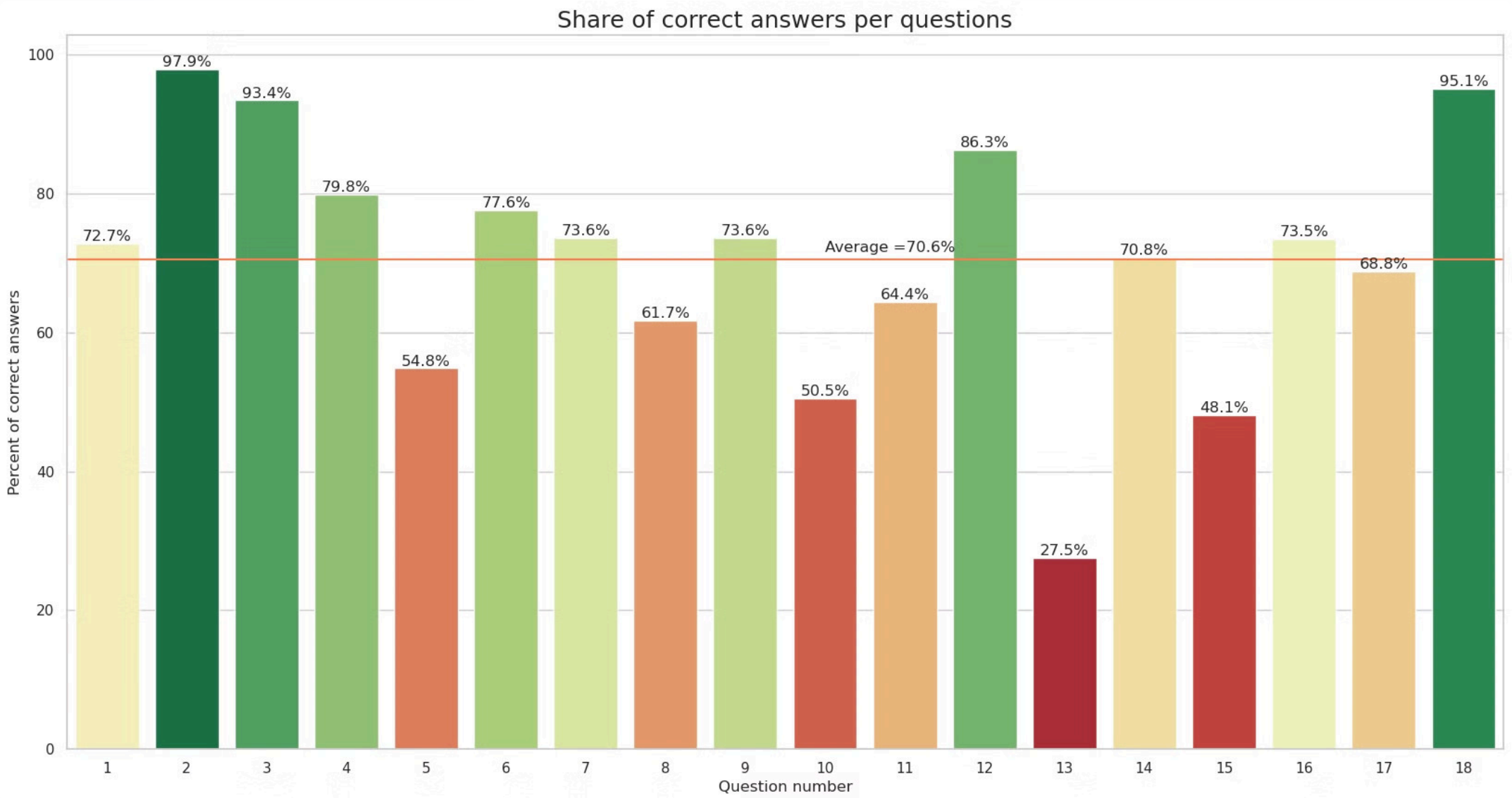
Train Data

Train data shape: (26296946, 20)
Test data shape: (3728, 21)
Labels data shape: (424116, 2)

Train Data Scheme

```
dtypes = {'session_id': 'category',
          'elapsed_time': np.int32,
          'event_name': 'category',
          'name': 'category',
          'level': np.uint8,
          'page': 'category',
          'room_coor_x': np.float32,
          'room_coor_y': np.float32,
          'screen_coor_x': np.float32,
          'screen_coor_y': np.float32,
          'hover_duration': np.float32,
          'text': 'category',
          'fqid': 'category',
          'room_fqid': 'category',
          'text_fqid': 'category',
          'fullscreen': np.int8,
          'hq': np.int8,
          'music': np.int8,
          'level_group': 'category'}
```

Labels Data



About the Game

Jo Wilder is a point-and-click game where players progress by finding hidden objects and answering quizzes.

Key data points:

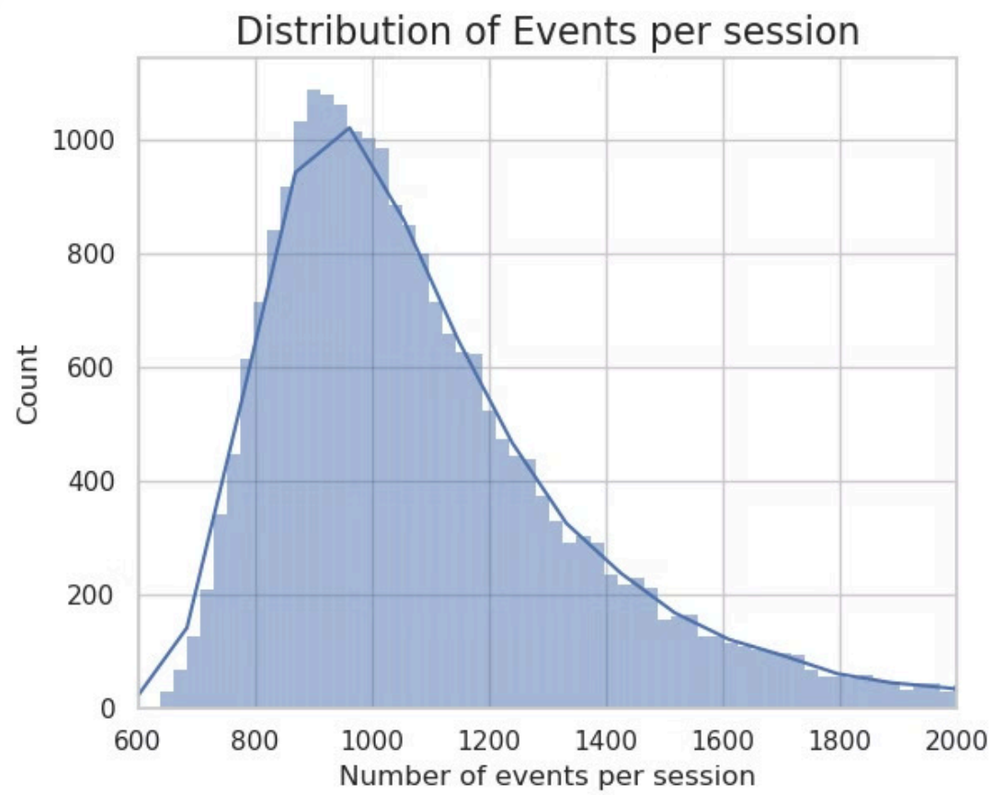
- Each game session is identified by `session_id`.
- Each row represents a game event (`name`, `event_name`, `fqid`, `index`).
- Player progress involves moving between rooms (`room_fqid`) and levels (`level`).
- Other columns detail specific events: click coordinates (`room_coor_x`, `room_coor_y`, `screen_coor_x`, `screen_coor_y`), notebook page numbers (`page`), and hover durations (`hover_duration`).

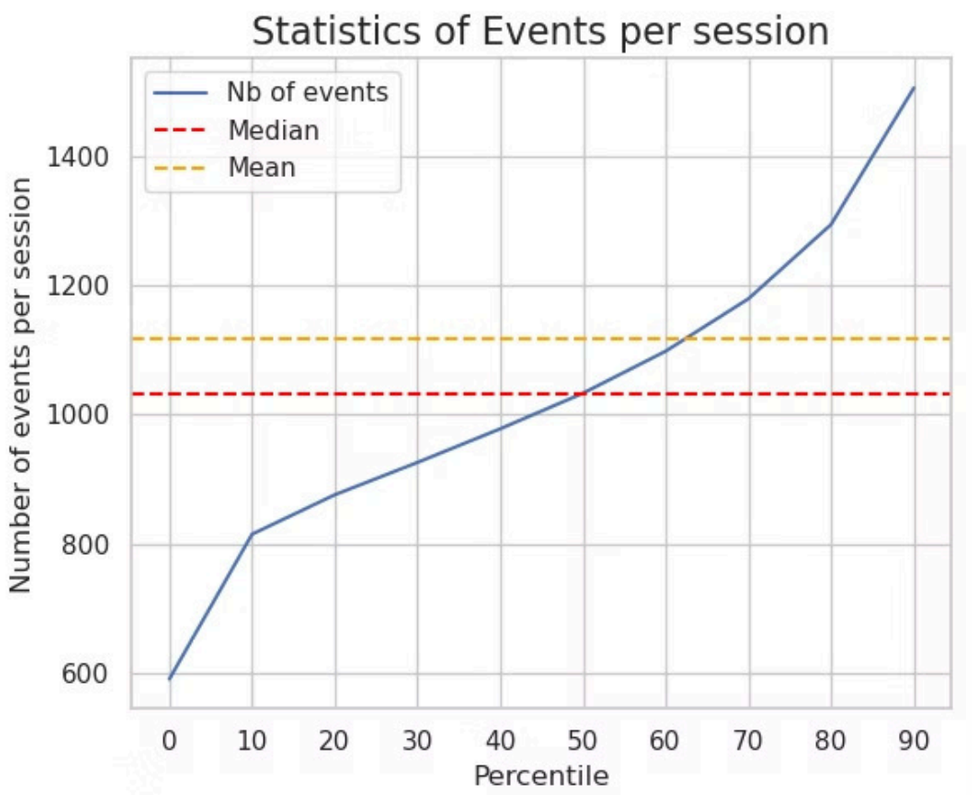
session_id

the ID of the session the event took place in

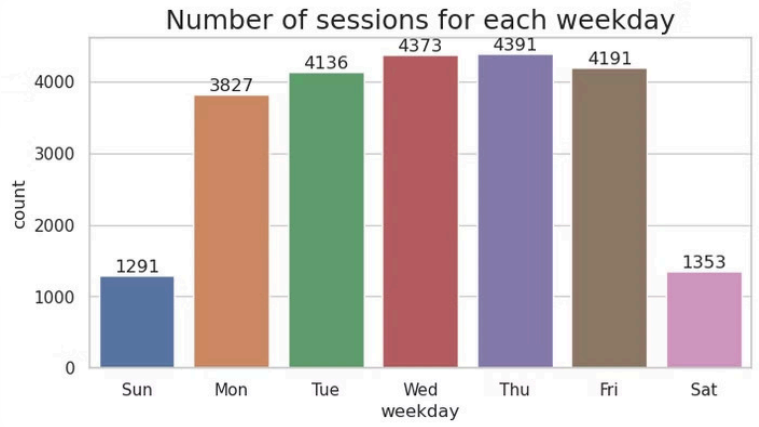
Unique sessions in train: 23562
Unique sessions in test: 3

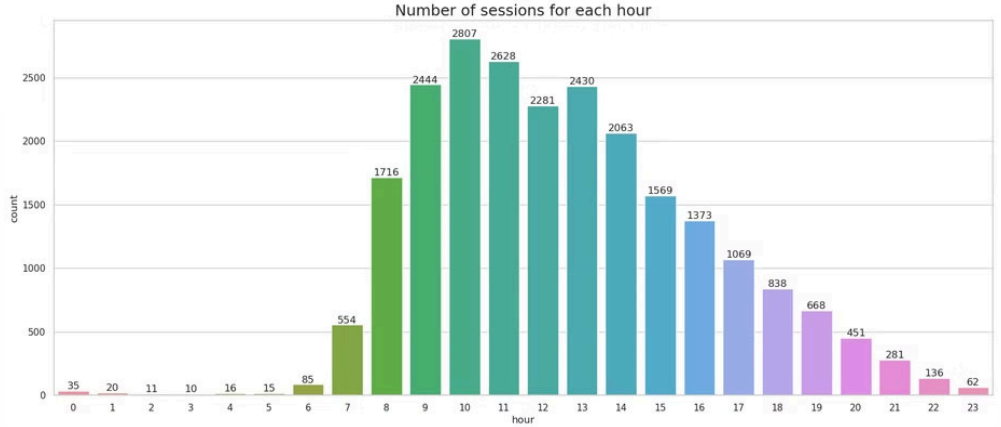
A session typically encompasses anywhere between several hundred to a few thousand individual events





`session_id` likely contains date and time information





The game is mostly played during the school days (Mon to Fri) and working hours (8 am. to 18 pm.)

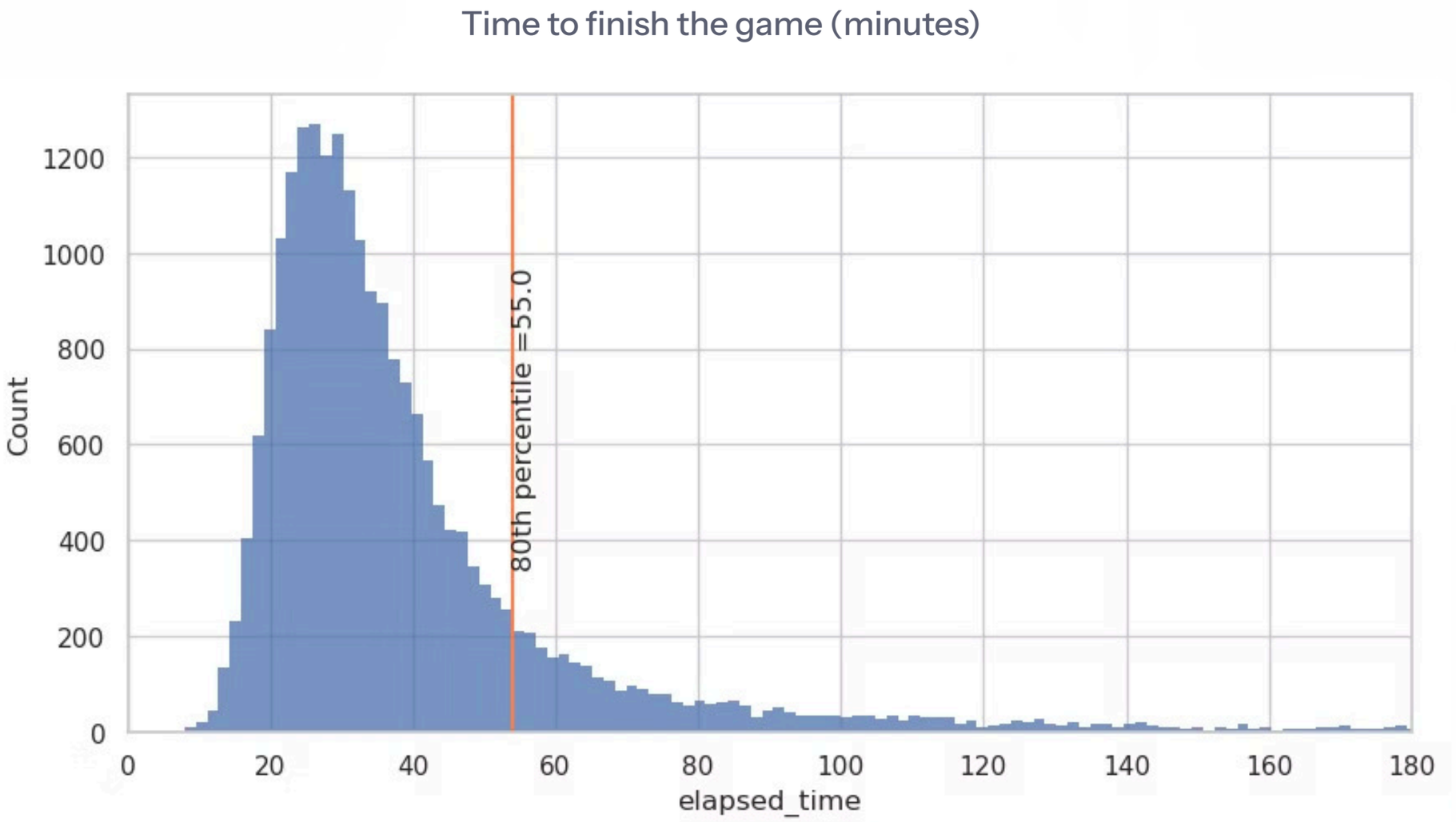
index

the index of the event in a session.

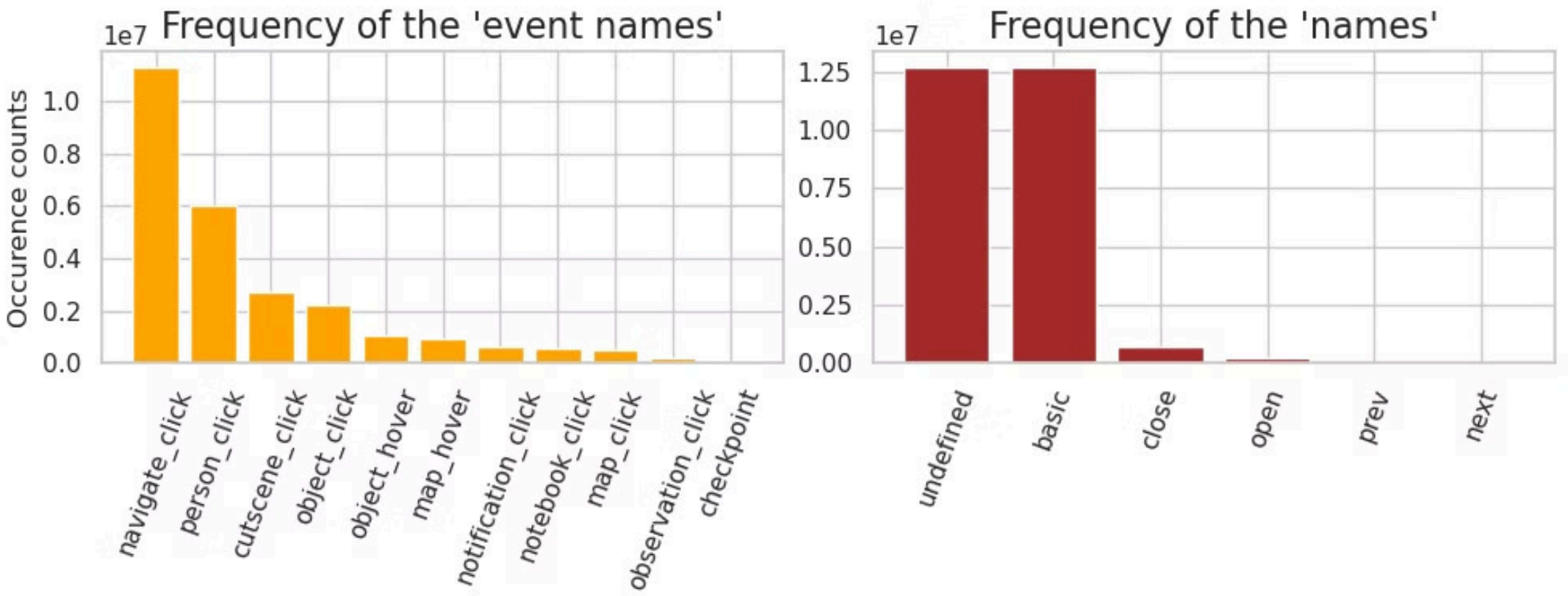
- Combination of session_id and index is not unique, there are duplicates.
- Some sessions have no index = 0, which is the start event of the game.
- Some sessions have the "reversed index" phenomenon which is the index in that session do not monotonic increase.

elapsed_time

shows how much time has passed (in milliseconds) between the start of the session and when the event was recorded

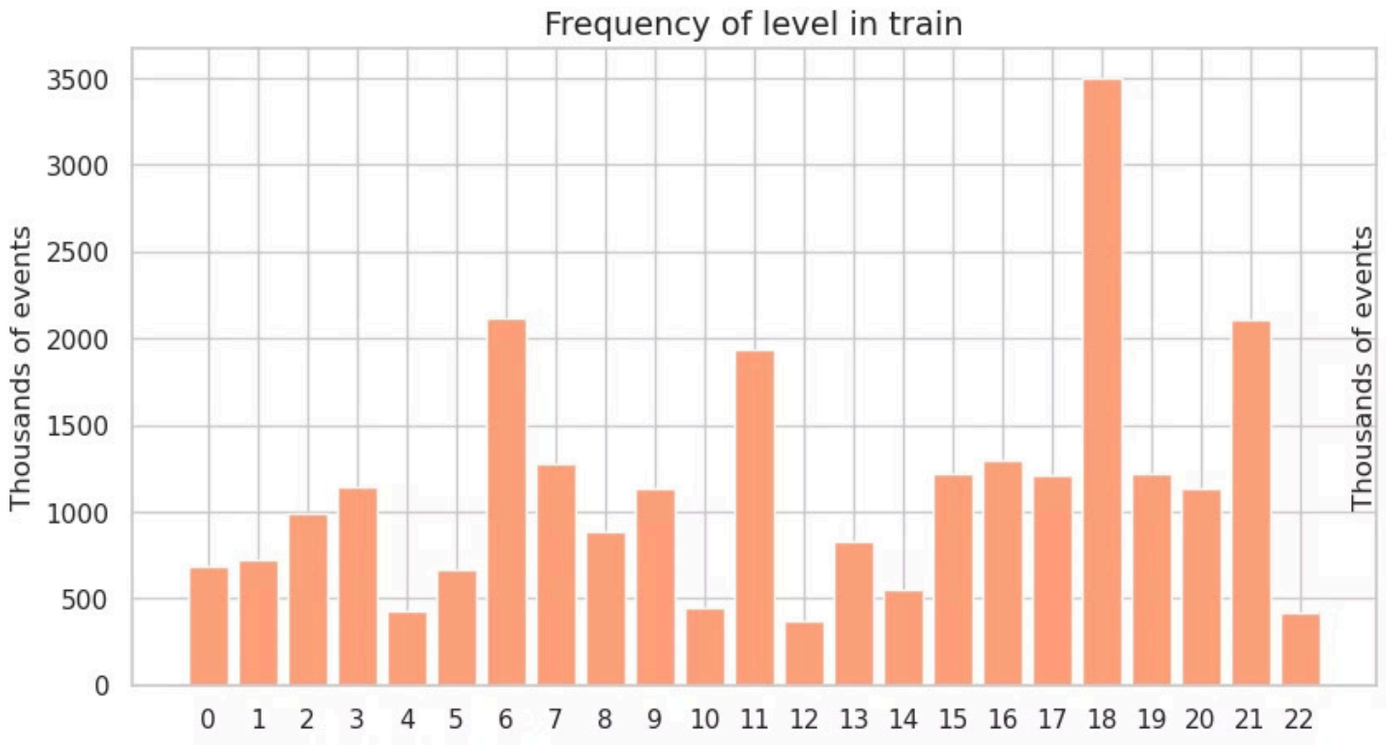


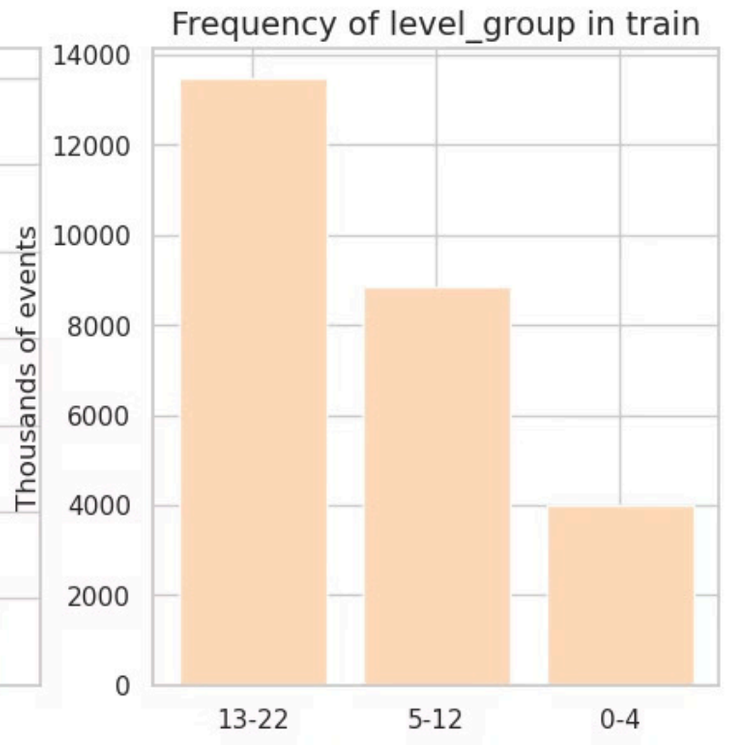
event properties: event_name, name



level and level_group

level_group	questions	number of questions
0 to 4	1 to 3	3
5 to 12	4 to 13	10
13 to 22	14 to 18	5



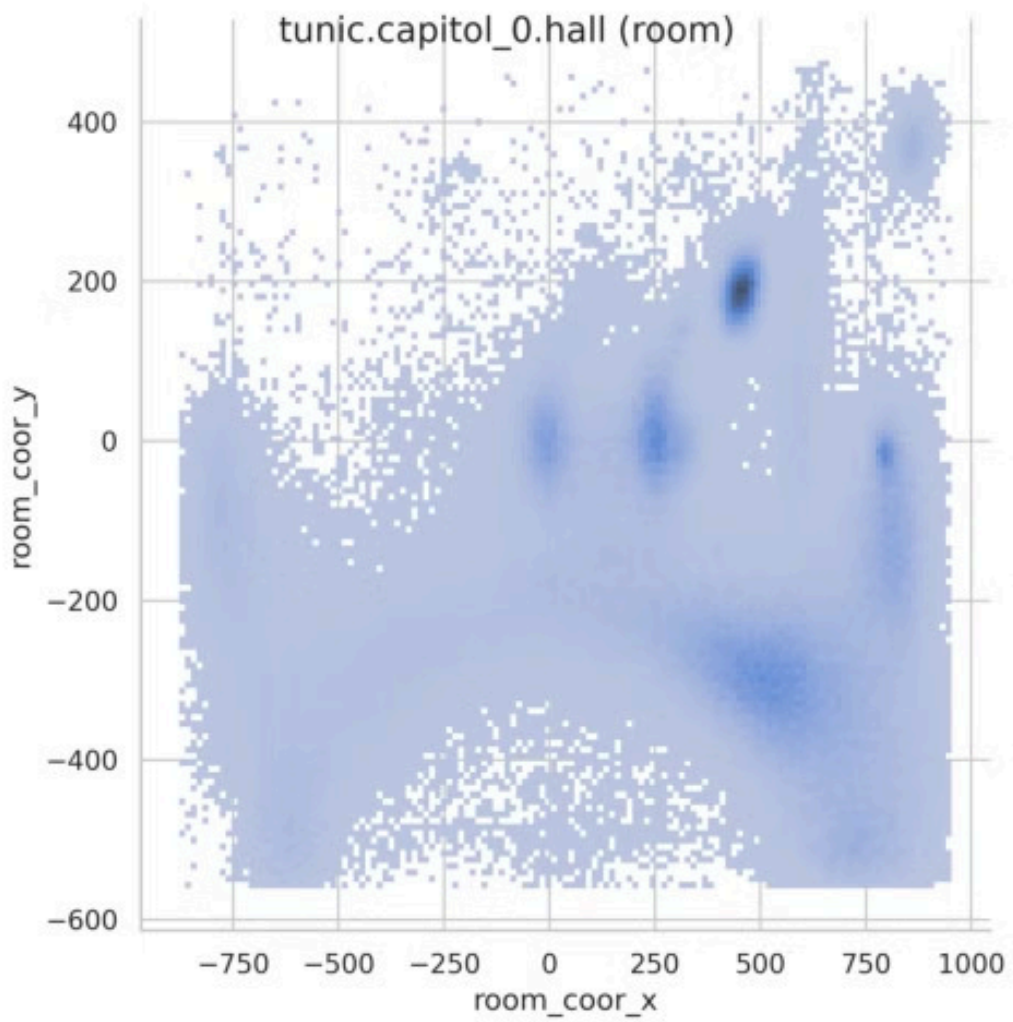


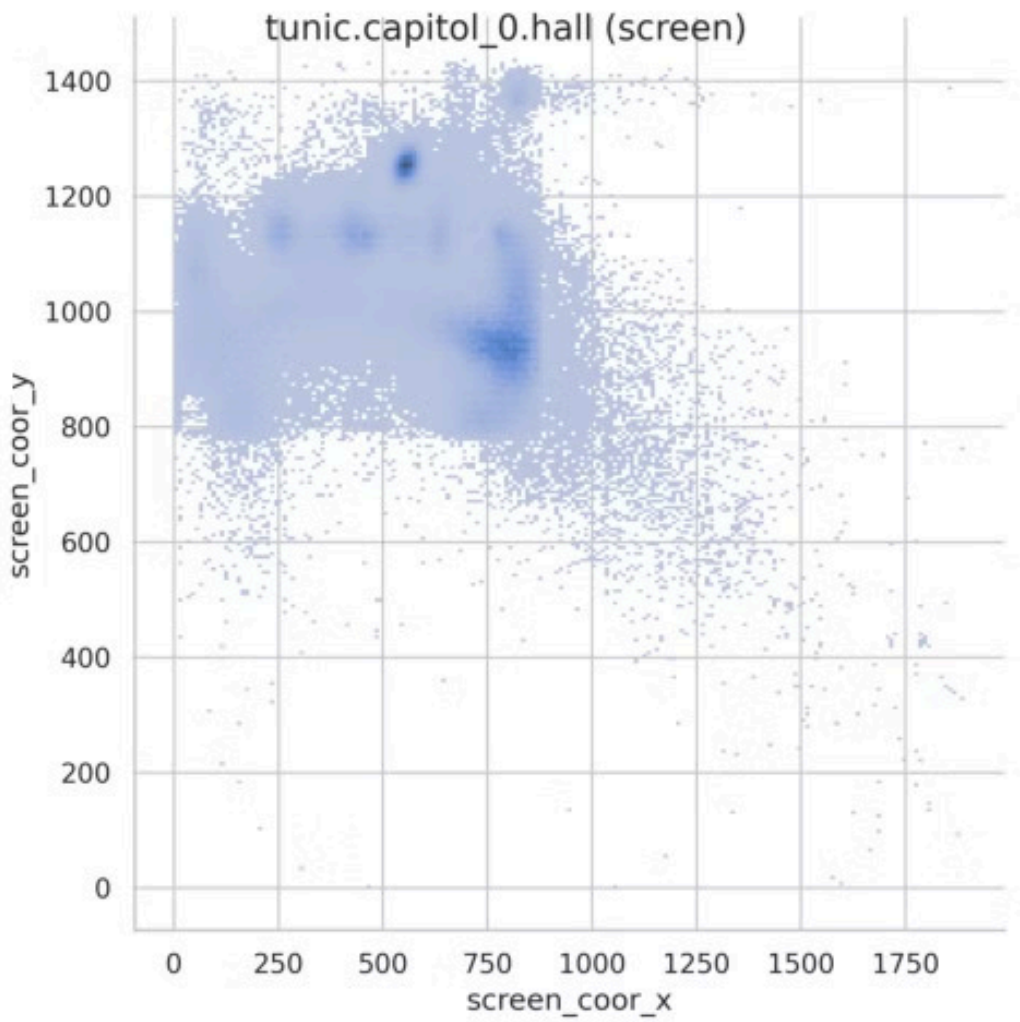
click geo-location

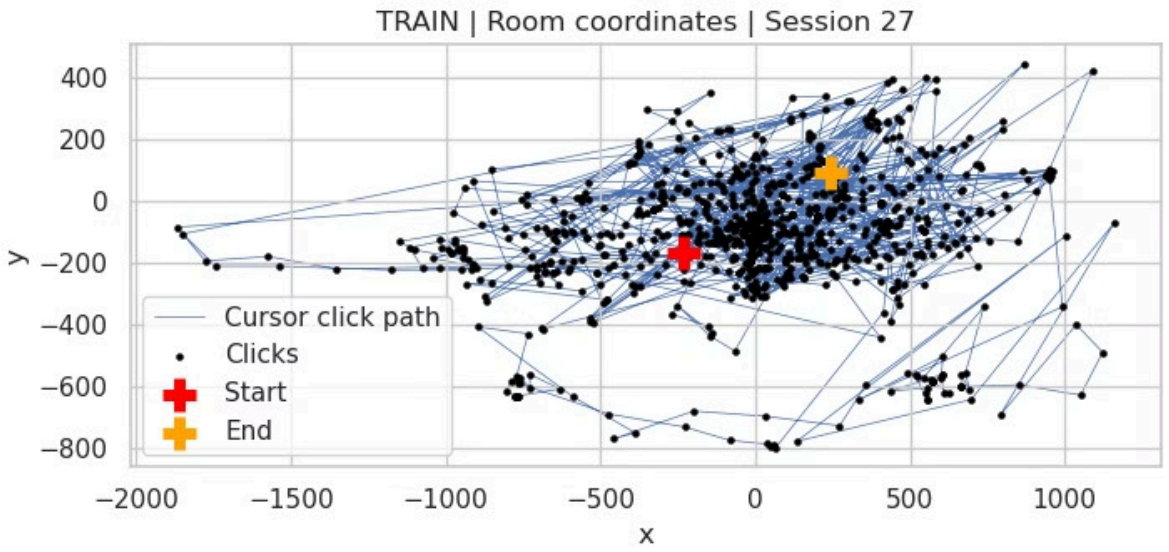
Click coordinates are provided relative to both the in-game room (`room_coor_x`, `room_coor_y`) and the player's screen (`screen_coor_x`, `screen_coor_y`).

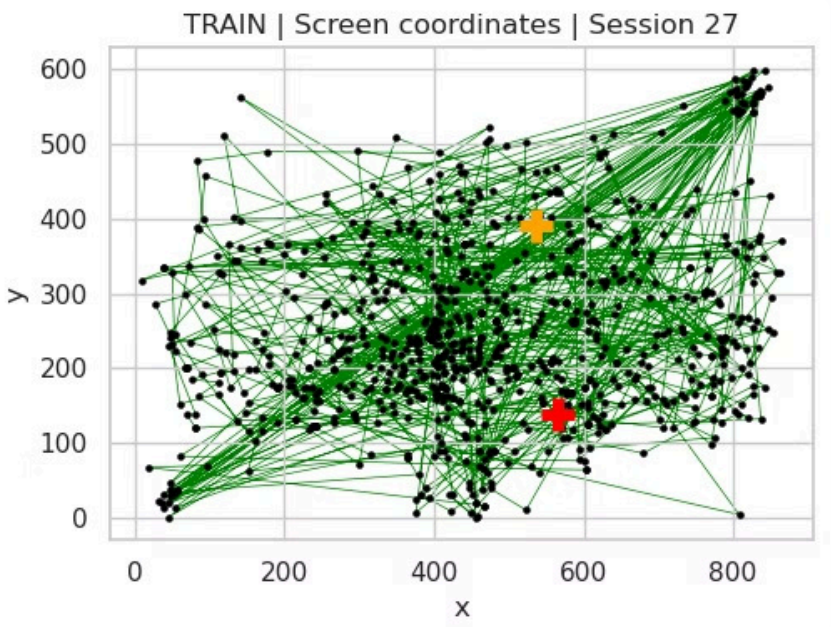
Click events comprise approximately 90-92% of all recorded events.

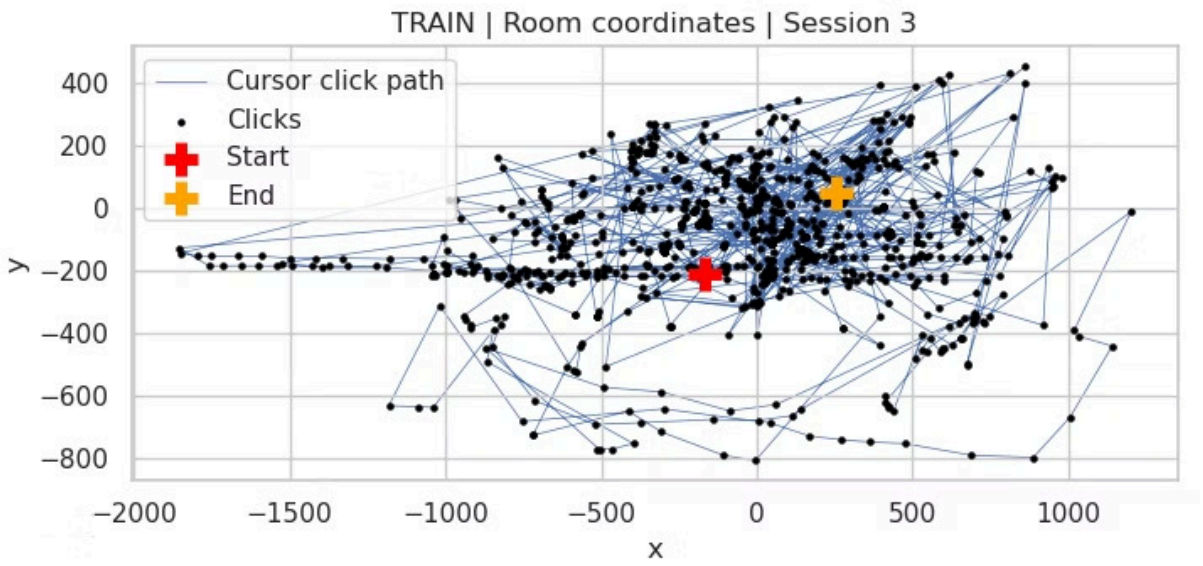
Click Density of room `tunic.capitol_0.hall`

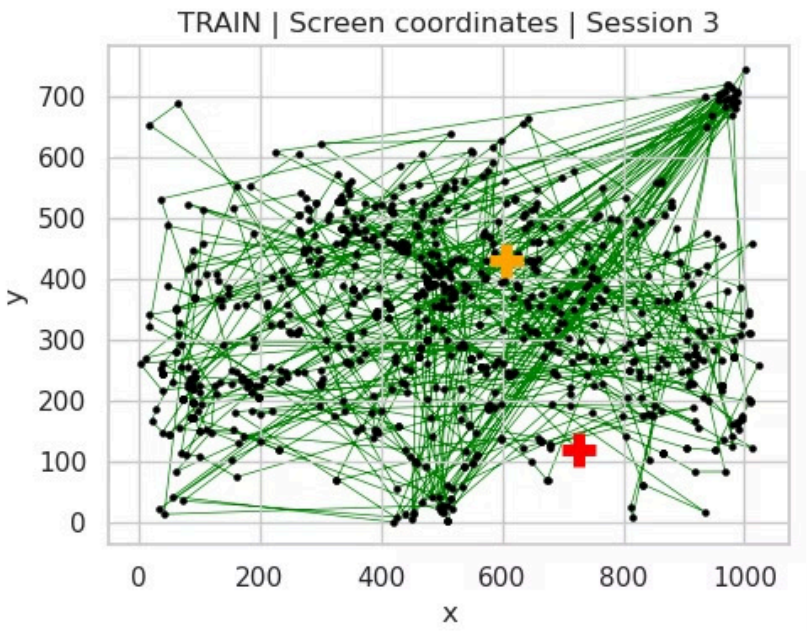


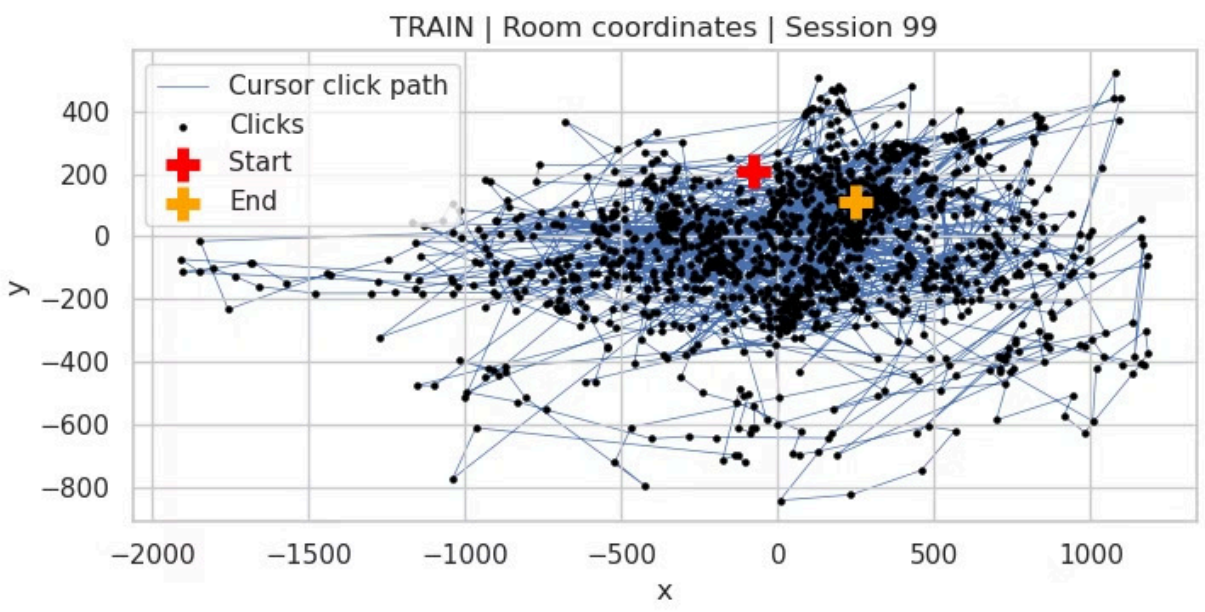


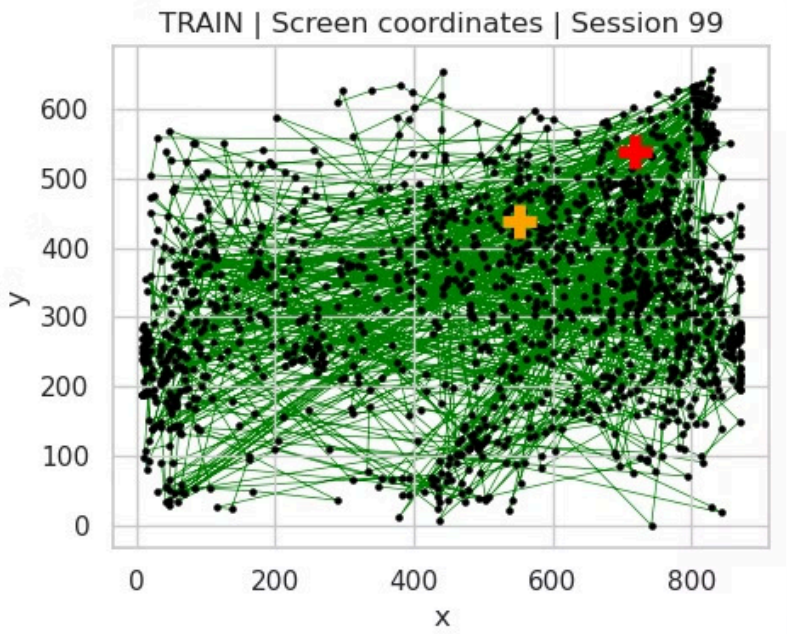






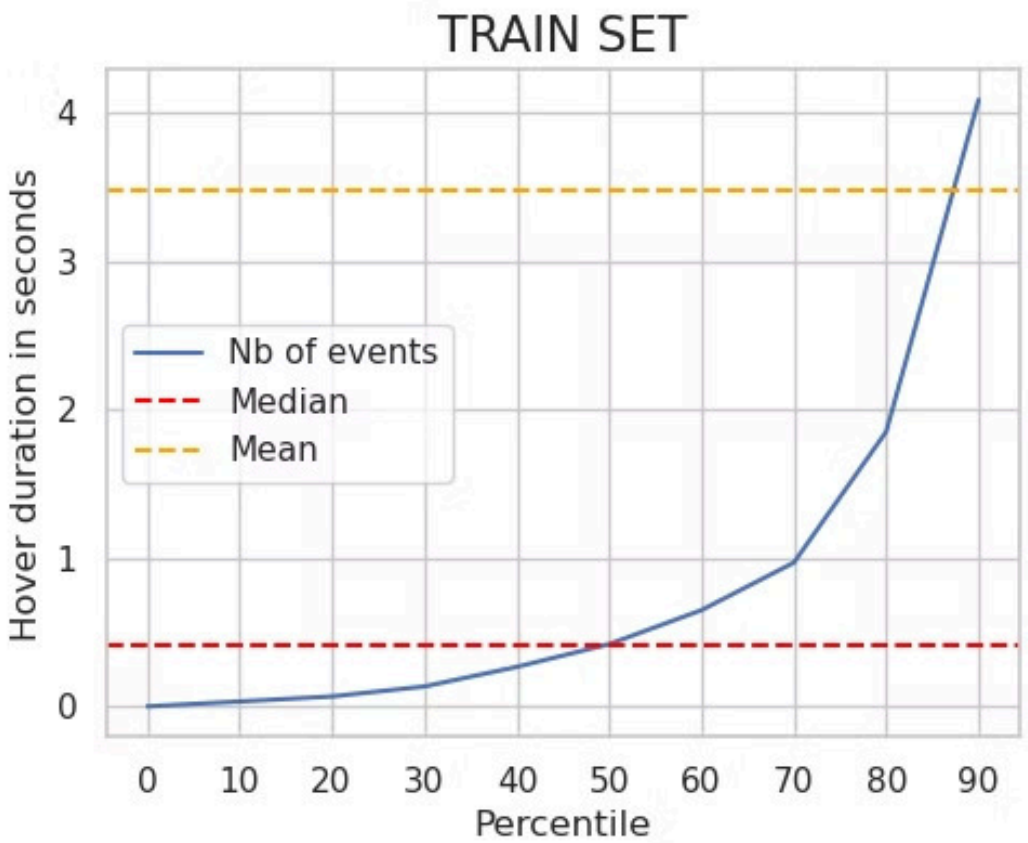


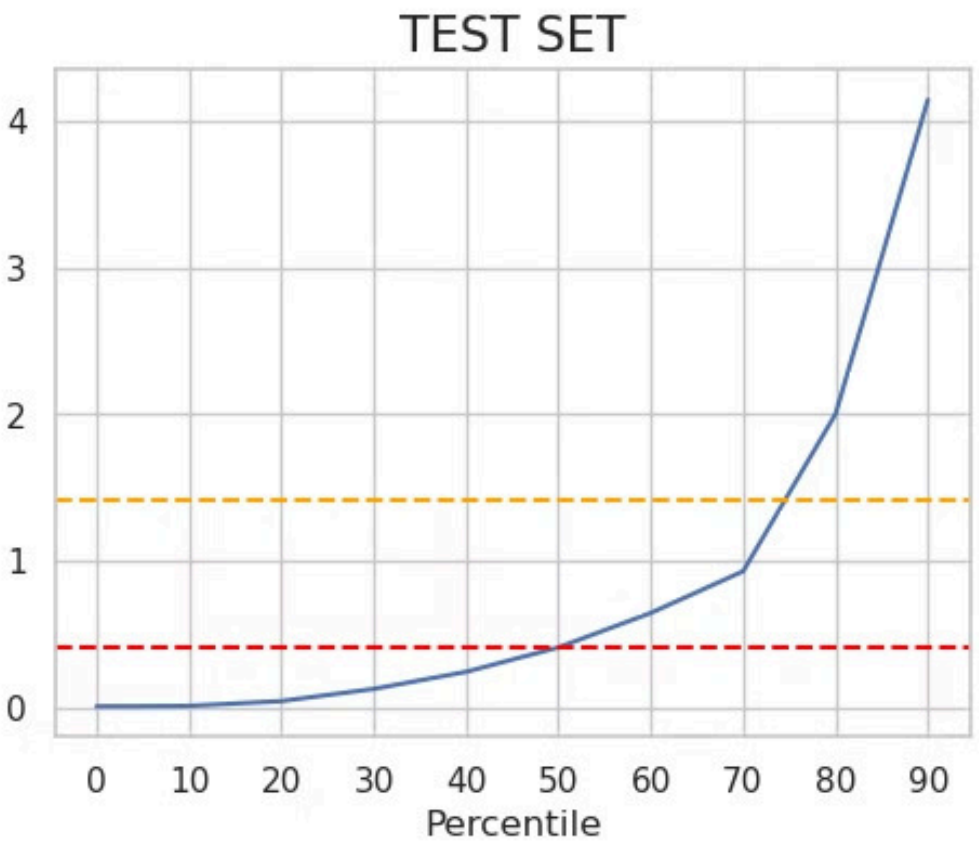




hover_duration

how long (in milliseconds) the hover happened for (only for hover events)





DATA PREPROCESSING

Train Data – train.csv, train_labels.csv

There are **3 level groups** and every questions will happen **after a level checkpoint (4, 12, 22)**. The idea is **splitting data into 3 level groups.**

And **all the records happens before a checkpoint** will be used to predict the questions at the checkpoint.

- dataset_df_1 uses data from **level 0 → 4**
- dataset_df_2 uses data from **level 0 → 12**
- dataset_df_3 uses data from **level 0 → 22**

```
dataset_df_1: (3981005, 20)
dataset_df_2: (12825243, 20)
dataset_df_3: (26296946, 20)
```

	session_id	correct	session	q
0	20090312431273200_q1	1	20090312431273200	1
1	20090312433251036_q1	0	20090312433251036	1
2	20090312455206810_q1	1	20090312455206810	1
3	20090313091715820_q1	0	20090313091715820	1
4	20090313571836404_q1	1	20090313571836404	1

DATA PREPROCESSING

Feature Selection Data – feature_sort.csv

a dataset capturing detailed relationships between features for this particular competition

	nabor	tip	quest	kol_col	col1	val1	col2	val2	col3	val3	...	kach4	k
0	5		18	1	text_fqid	tunic.flaghouse.entry.flag_girl.symbol_recap	0	0	0	0	...	0.551005	(
1	1		18	1	name	undefined	0	0	0	0	...	0.546731	(
2	4		18	1	room_fqid	tunic.drycleaner.frontdesk	0	0	0	0	...	0.546318	(
3	1		18	1	name	close	0	0	0	0	...	0.543156	(
4	8		18	1	text	Make sure to get some old photos for the exhib...	0	0	0	0	...	0.551282	(

FEATURE ENGINEERING

Generate a new dataframe where each row represents a session, and columns are the aggregations summarizing all session events happens before the checkpoint level.

- Statistic features related to `elapsed_time` between 2 consecutive events.
- *Categorical* and *numerical* aggregates
- Date and time extracted from `session_id`
- Statistic features generated based on *feature selection data.*

```
We'll train 'dataset_df_1' with 128 features
We'll train 'dataset_df_2' with 353 features
We'll train 'dataset_df_3' with 523 features
```

MODEL BUILDING

Train GroupKFold models with XGBoost baseline for 18 questions in the game

```
xgb_params = {
    'objective': 'binary:logistic',  # Specifies binary classification (logistic regression)
    'eval_metric': 'logloss',        # Evaluation metric used during training (log loss for binary classification)
    'learning_rate': 0.05,           # Step size during training (smaller values require more boosting rounds)
    'max_depth': 4,                  # Maximum depth of trees (helps control model complexity)
    'n_estimators': 6000,            # Number of boosting rounds (trees) to build
    'early_stopping_rounds': 50,     # Stop training early if validation score doesn't improve for 50 rounds
    'tree_method': 'hist',           # Tree-building method (histogram-based method for faster training)
    'subsample': 0.8,                # Fraction of samples to use for each tree (helps prevent overfitting)
    'colsample_bytree': 0.4,         # Fraction of features to use for each tree (helps with regularization)
    'use_label_encoder': False       # Disables label encoding for the target variable (to avoid warnings)
}
```

MODEL TRAINING

For each of 18 questions, select the corresponding dataset and features, then perform 5-fold cross-validation using GroupKFold (for none-overlapping groups) and split data into train and validation sets.

For each fold, train an XGBoost classifier on the training set and validate the model using the validation set.

```
FOR question IN 18_questions:
  group = select_group(question)
  group_train = raw_train[group]
  FOR fold IN 5_folds:
    train_set, valid_set = group_train[fold]
    xgb_model.fit(train_set, valid_set)
```

```
### q_no 16 grp 13-22 feats :  523
Done for  13-22 16 0
Done for  13-22 16 1
Done for  13-22 16 2
Done for  13-22 16 3
Done for  13-22 16 4
### q_no 17 grp 13-22 feats :  523
Done for  13-22 17 0
Done for  13-22 17 1
Done for  13-22 17 2
Done for  13-22 17 3
Done for  13-22 17 4
### q_no 18 grp 13-22 feats :  523
Done for  13-22 18 0
Done for  13-22 18 1
Done for  13-22 18 2
Done for  13-22 18 3
Done for  13-22 18 4
```

MODEL TRAINING

After having the models, I calculate out-of-fold (OOF) probability predictions for all sessions and questions.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
20090312431273200	0.881475	0.992872	0.967973	0.891234	0.667184	0.898649	0.895482	0.680301	0.851501	0.631158	0.779715	0.912305	0.321548	0.828352	0.6
20090312433251036	0.815894	0.989989	0.957041	0.487681	0.168589	0.415500	0.487956	0.543411	0.444420	0.228523	0.481028	0.762087	0.120662	0.270547	0.0
20090312455206810	0.804497	0.981032	0.969205	0.558306	0.709395	0.794033	0.755876	0.696865	0.792604	0.658639	0.706066	0.890314	0.460072	0.519252	0.3
20090313091715820	0.536271	0.971299	0.926895	0.827294	0.533711	0.731084	0.791153	0.555691	0.696316	0.492798	0.675896	0.893706	0.129585	0.731431	0.4
20090313571836404	0.942762	0.998094	0.990665	0.979001	0.852007	0.952956	0.937087	0.842838	0.927818	0.777648	0.799010	0.942125	0.509552	0.882008	0.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22100215342220508	0.826889	0.996679	0.959871	0.968835	0.725543	0.853032	0.825838	0.643127	0.823795	0.526427	0.575858	0.922747	0.202798	0.791189	0.6
22100215460321130	0.462524	0.976916	0.853742	0.885819	0.588612	0.816780	0.723744	0.594588	0.820762	0.542596	0.711282	0.905633	0.171986	0.770272	0.5
22100217104993650	0.754615	0.985878	0.974652	0.872002	0.693690	0.890691	0.879477	0.668096	0.846325	0.604576	0.707722	0.911637	0.218484	0.911347	0.7
22100219442786200	0.650519	0.988792	0.932136	0.816956	0.477832	0.835379	0.715597	0.648719	0.770222	0.579957	0.685784	0.904950	0.321814	0.789533	0.5
22100221145014656	0.464392	0.956740	0.911421	0.750425	0.394571	0.664749	0.644435	0.477266	0.658065	0.310452	0.523531	0.840168	0.101881	0.567071	0.2

23562 rows × 18 columns

Then, find best threshold that

IF oof.values > threshold
THEN correct = 1
ELSE correct = 0

```
Best threshold  0.6200000000000002      F1 score  0.696613666809234
```

MODEL TESTING

Before testing with new data, apply model with best threshold to measure the F1-score

```
When using optimal threshold...
Q0: F1 = 0.6678643298891116
Q1: F1 = 0.5261196890476701
Q2: F1 = 0.5079421982478146
Q3: F1 = 0.6773461025874492
Q4: F1 = 0.6369255385736874
Q5: F1 = 0.6432572373431327
Q6: F1 = 0.6300663952001252
Q7: F1 = 0.5714563228701321
Q8: F1 = 0.6283704217674289
Q9: F1 = 0.5856005244390425
Q10: F1 = 0.6102647885269563
Q11: F1 = 0.5186331466589013
Q12: F1 = 0.45966236142531197
Q13: F1 = 0.6354450526291678
Q14: F1 = 0.6053883336992303
Q15: F1 = 0.49842194969282805
Q16: F1 = 0.5541053108013009
Q17: F1 = 0.4956494787254747
==> Overall F1 = 0.696613666809234
```

Then, predict with new test data:

```
session_id,correct
20090109393214576_q1,1
20090109393214576_q2,1
20090109393214576_q3,1
20090109393214576_q4,1
20090109393214576_q5,0
20090109393214576_q6,1
20090109393214576_q7,1
20090109393214576_q8,1
20090109393214576_q9,1
```

FINAL RESULT

0.700

Private Score

0.698

Public Score

Overview Data Code Models Discussion Leaderboard Rules Team Submissions

Submissions

0/3

You selected 0 of 3 submissions to be evaluated for your final leaderboard score. Since you selected less than 3 submissions, Kaggle auto-selected up to 3 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

Submissions evaluated for final score

All Successful Selected Errors Recent

Submission and Description	Private Score	Public Score	Selected
HCMUS-2425-21127038 Model Building - Version 15 Succeeded (after deadline) · 1d ago · Notebook HCMUS-2425-211270...	0.700	0.698	☐

CONCLUSION

1

Success

The model achieved promising results, demonstrating the effectiveness of using game logs to predict student performance.

2

Future Work

Future research could explore ensemble methods to further improve model performance and delve deeper into feature selection, aiming to create more accurate and insightful predictive models.