Predicting Student Performance From Game play

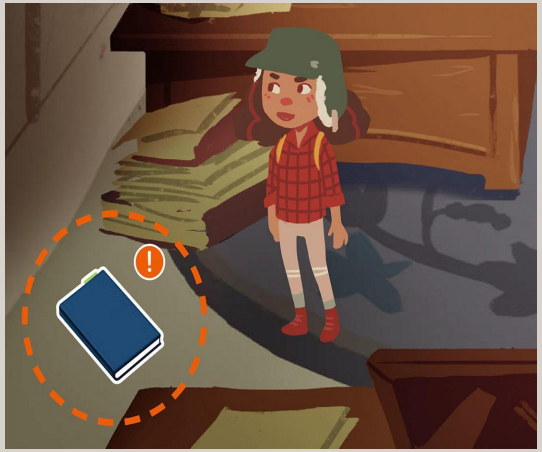
Paul Merica

Wayne State Advanced Analytics

Masters

01 Objective

The goal of this competition is to see whether we can predict whether student will answer questions correctly (given in a binary format of (1 = correct, 0 = wrong) in a game after playing over time. This is a competition sponsored by Field Lab Day based on a educational game they developed which can be seen on the right. Game has 3 checkpoints where you need to answer a series of questions.



02 Scoring Metrics For the Competition

- This competitions used two scoring tracks
  - Efficiency + accuracy
    - Algorithm to determine if your model can be accurate & efficient
  - Accuracy
    - No efficiency needed; Score by F1 Score
      - F1 Score is determined by the harmonic mean of precision and recall, where precision is the number of true positive results divided by the number of all positive results, and recall is the number of true positive results divided by the number of actual positives.

F1 is calculated as follows:

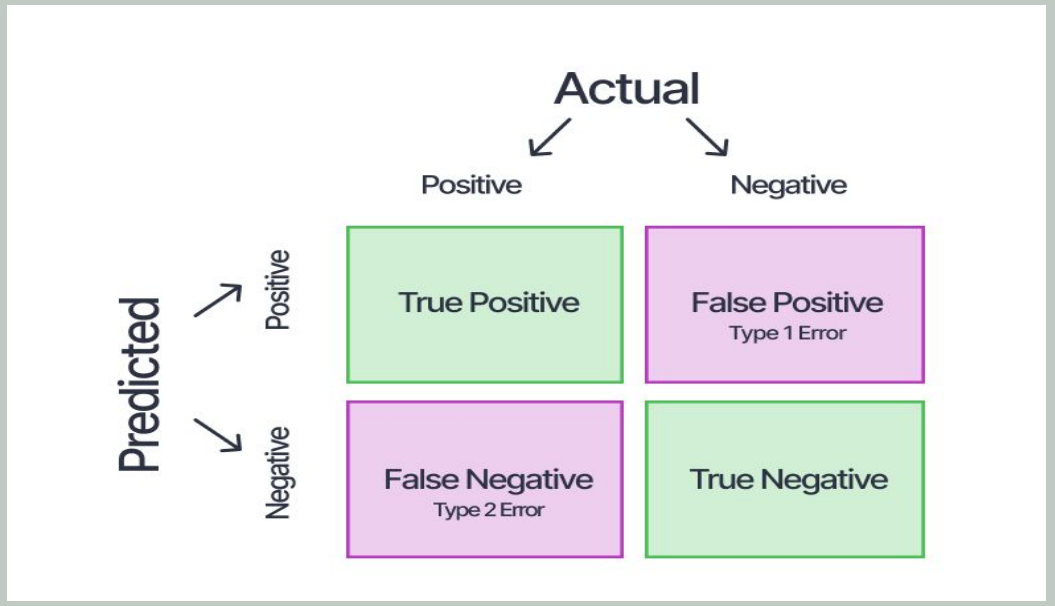
$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

where:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

In "macro" F1 a separate F1 score is calculated for each `species` value and then averaged.

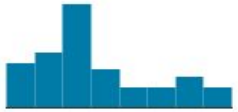


Data Overview

Columns

- **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)

Data Overview cont.

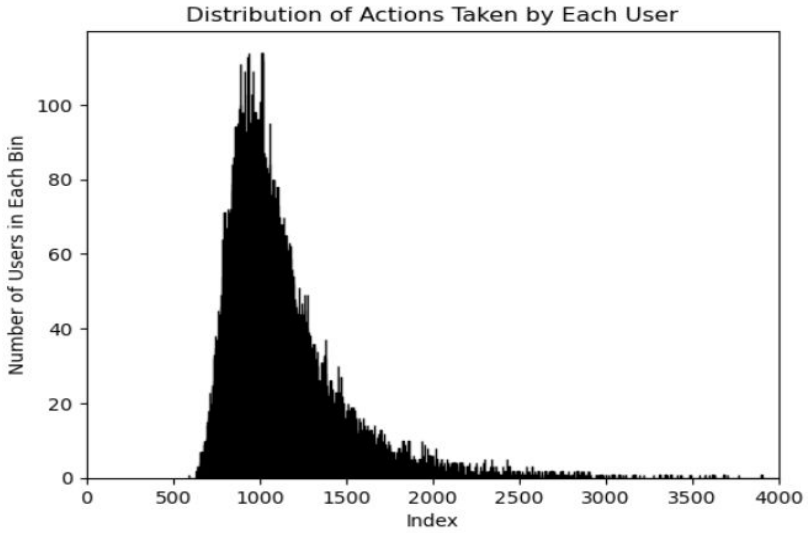
session_id	index	elapsed_time	event_name	name	level
 20090312b 22100221b	 0 20.5k	 0 1.99b	navigate_click 43% person_click 23% Other (8917660) 34%	undefined 48% basic 48% Other (942692) 4%	 0
20090312431273200	0	0	cutscene_click	basic	0
20090312431273200	1	1323	person_click	basic	0
20090312431273200	2	831	person_click	basic	0
20090312431273200	3	1147	person_click	basic	0
20090312431273200	4	1863	person_click	basic	0
20090312431273200	5	3423	person_click	basic	0
20090312431273200	6	5197	person_click	basic	0
20090312431273200	7	6180	person_click	basic	0
20090312431273200	8	7014	person_click	basic	0
20090312431273200	9	7946	person_click	basic	0
20090312431273200	10	9133	navigate_click	undefined	0
20090312431273200	11	10263	person_click	basic	0
20090312431273200	12	12030	navigate_click	undefined	0
20090312431273200	13	13030	observation_click	basic	0

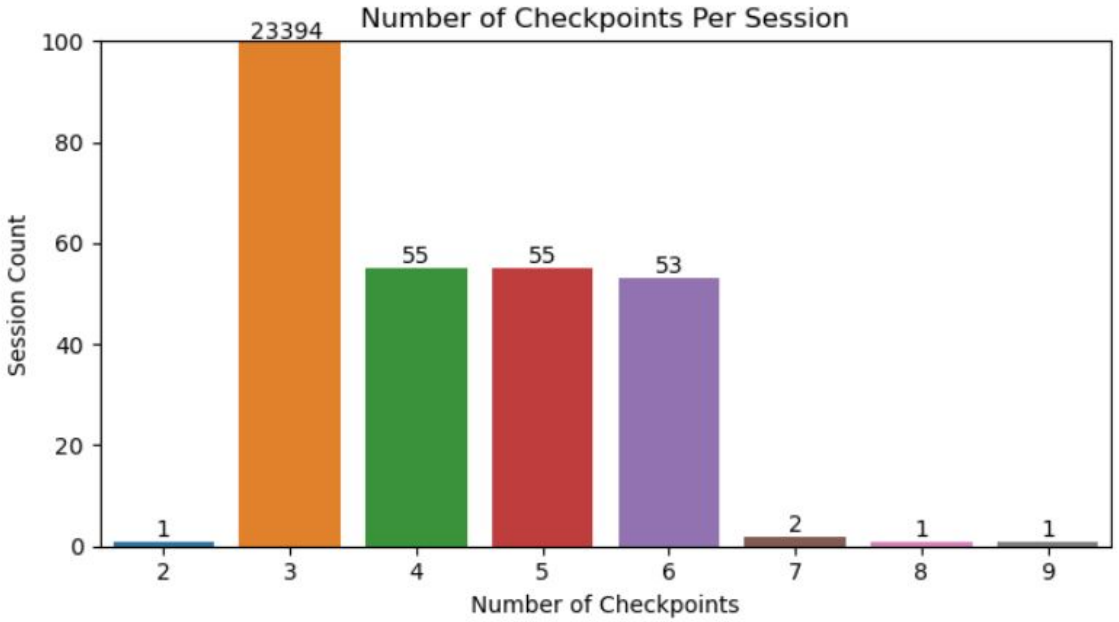
Challenges with The Competition

- Data size has increased from start of competition. It is now ~ 23 million rows of training data. Thus code takes a long time to run.
- Competition requires you to send in code so it can score the data manually through a loop iterating through the 18 questions.
  - This leads to us needing to use one notebook to train the model (maximum allowed competition Ram is 8 GB) in an outside Kaggle notebook with 32 GB. Then infer the given modele weights/parameters to a separate notebook to score the testing data.

Data Challenges

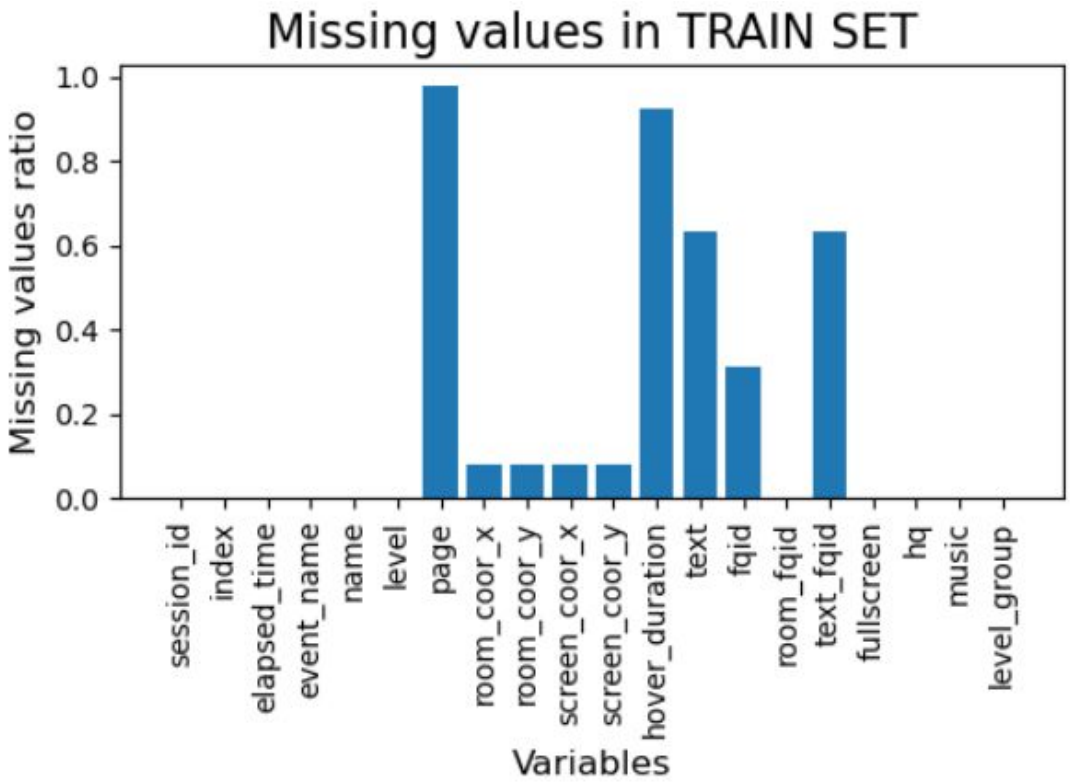
- Each Session takes a different amount of time so there is no set number of indices before reaching a “checkpoint” where you will answer questions
- Every user should only have 3 checkpoints but this is not necessarily true in the data.

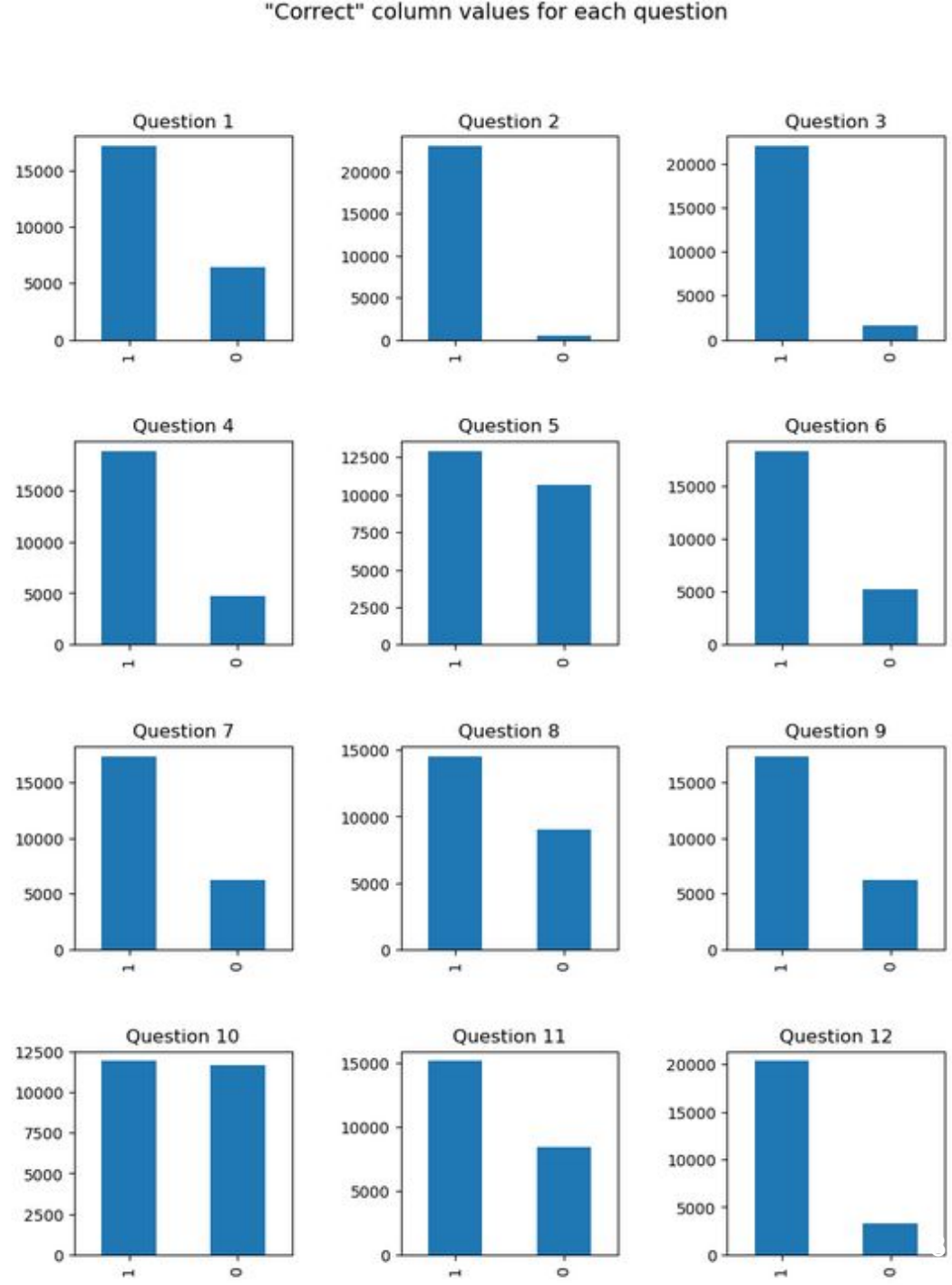




Data Challenge Cont.

- High degree of missingness the data as some columns only populate when other columns do.
- Class Imbalances in each of the questions





Feature Selection/Creation

- Due to the size of the data, lack of computational resources and the amount of dimensionality there was a need to flatten the data.
- This meant creating new features that were the sum, mean and standard deviation of actions taken. This should give a relative idea of how the user played the game.
- I also employed forward selection to see the most important variables, but this was computationally very intensive and hard to use effectively.

Models Used

- Logistic Regression
  - Pros
    - Easily interpretability
    - Fast and Efficient
  - Cons
    - Bad on imbalanced datasets
    - Sensitive to Noise

- Random Forest
  - Pros
    - Easily interpretability
    - Fast and Efficient
  - Cons
    - Slow on large datasets
- XGBoost
  - Pros
    - Highly efficient and scalable, especially for large datasets
    - Built-in regularization techniques to prevent overfitting
  - Cons
    - Lacks interpretability

Modeling Approach

- Building one model to answer all questions did not work well and the API used to score the test data worked better with building a different model for every question
    - This also allowed us to solve for class imbalances in each question
- I built 18 models for each question in each of the 3 methods used before to score the data.

Model Accuracy and Model Improvement/Tuning

K-fold Cross Validation

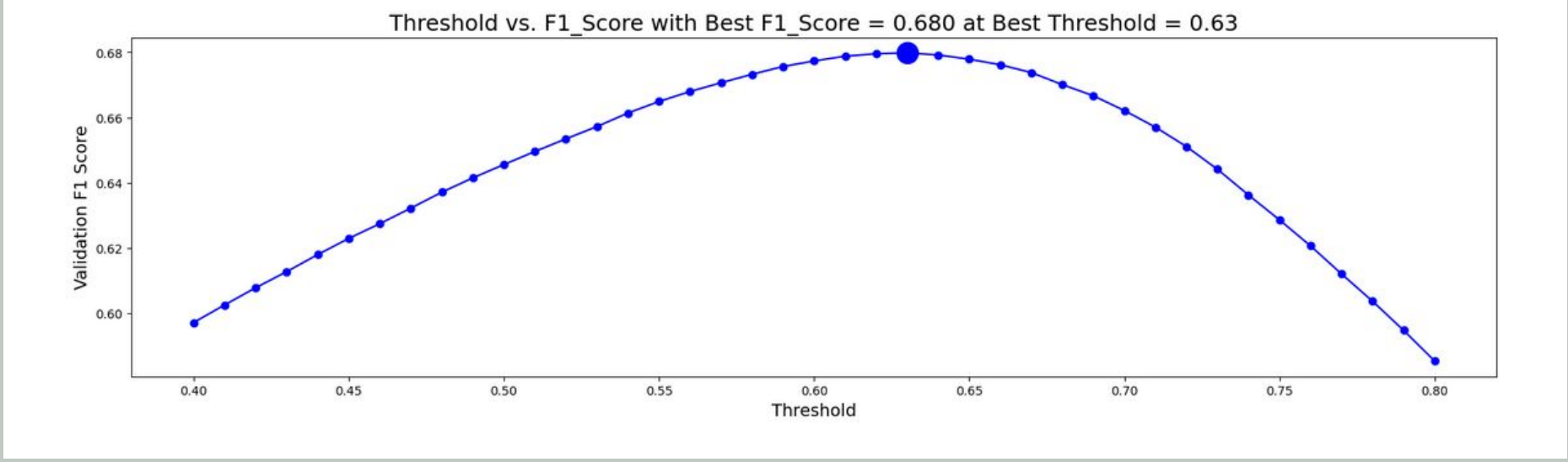
- The algorithm then trains the model K times, each time using a different fold as the validation set, and the remaining folds as the training set.
- Out of Fold Predictions
- Helped to evaluate the model's performance on the entire dataset.

Correcting for Data imbalance

- XGBoost has built in function to account for imbalances by adjusting the weights inversely proportional to the dara
- Oversampled minority classes in Logistic Regression
- Random Forest has built-in function

Results

- XGBoost ended up being the best model with F1 score of .680 and optimal threshold of .63.. Scored .677 on public dataset.
- RandomForest model scored the second best at an F1 Score of .677 and optimal threshold of .63
- Logisitic Regression scored an F1 Score of .55 and optimal threshold of .87
- All models performed better before class imbalances were fixed except XGBoost which performed the same.



Accomplishments

- Published Notebook on how to infer model parameter to final testing environment
- Scored in middle of leaderboard
- Evaluated 3 Modelling techniques to do the best scoring on this dataset.
- Attacked the problem without using deep learning.

Future Possibilities

- More intelligent feature creation
- RNN or Deep Learning Model
- Mixing and matching the best performing models for each question
- Using Polars instead of PanDas for better run times

```
When using optimal threshold...
Q0: F1 = 0.6329314531694038
Q1: F1 = 0.4966297565518459
Q2: F1 = 0.4956819829916277
Q3: F1 = 0.6130362234577051
Q4: F1 = 0.5723262182403723
Q5: F1 = 0.6080634207407511
Q6: F1 = 0.5910405833116157
Q7: F1 = 0.5296943154440827
Q8: F1 = 0.6016343983692867
Q9: F1 = 0.5171825616643508
Q10: F1 = 0.585113614684866
Q11: F1 = 0.4927437978887068
Q12: F1 = 0.43443641789988063
Q13: F1 = 0.6167888452753806
Q14: F1 = 0.5308563290303532
Q15: F1 = 0.4698157606273046
Q16: F1 = 0.5478808340725085
Q17: F1 = 0.4922038731147179
==> Overall F1 = 0.6797811221010304
```