Data Science Bowl® 2019: Predicting kids performance in educational game apps

Juan Jose Garau (*garau@mit.edu*), Inigo de la Maza (*delamaza@mit.edu*)

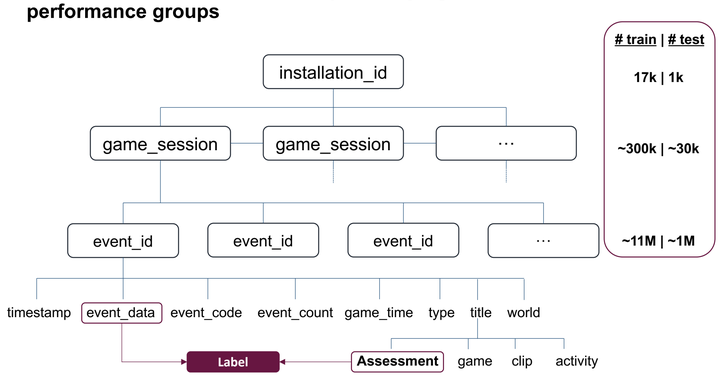
15.095: Machine Learning under a modern Optimization Lens

1. Background and Objectives

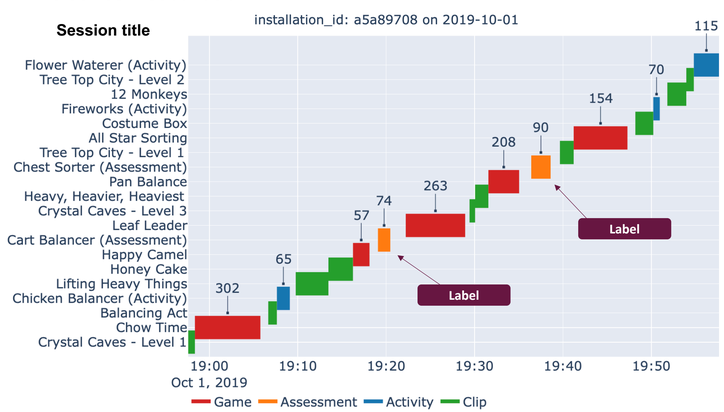
- The Data Science Bowl® is the world's premier **data science for social good** competition, it brings together experts across industries to take on the world's challenges with data and technology
- The 2019 edition partners with **PBS KIDS**, a trusted name in early childhood education (3-5 years old), who aims to gain **insights** on how **media** can **help children learn** important skills for success in school and life
- Based on historical data from its **app** *PBS KIDS Measure UP!*, the challenge focuses on **predicting scores** on in-game assessments and **understanding** how to improve learning outcomes

2. Data

- The dataset contains **timestamp data** from the app activity of approximately **17,000 kids**, divided into different **sessions** (~300k) and **events** (~11M)
- Some of the sessions are **Assessments**, in which the kid must complete an **evaluation task** related to some measurement concept (length, weight…)
- The **goal** is, given **incomplete information** of a kid's actions, **predict how well** he/she will do in an Assessment, by **classifying** him/her into **one** out of **4 performance groups**

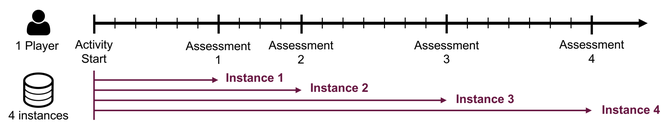


Time series visualization



3. Feature Engineering

- For each player (installation ID), we create different **instances** that include all previous events of that player until the beginning of every completed assessment



- We **clean** and **enrich** the data by:
  I. Getting rid of the players who haven't completed any assessment
  II. Using completed assessments in the test data as part of the training dataset
  III. Creating a function that computes the classification label for **completed** assessments

- We then extract the following features for each instance:

Assessment-related features (*how hard the assessment is*)

Which assessment | Which world | Mean & median score | Mean & median label

Player-related features (*how good the player is*)

Assessment hour | Assessment day of the week | Num. days played so far | *Title counter* | Num. played sessions

Num. previous completions | *Title* type counter | Same *world's title* type counter | *Event code* counter | Acc. score

Total minutes per *title* type | Total minutes played | Avg. minutes per *title* type | Avg. minutes per session | Acc. Num. correct actions

% correct actions in games | % correct actions in assessments | Acc. Num. incorrect actions | Previous score same assessment

- Final dataset:

n = ~23.5k, p = 129, k = 4

4. Performance results

- We test **OCT** and **OCT-H** models with and without class *autobalance* and compare its out-of-sample performance with logistic regression and *xgboost*

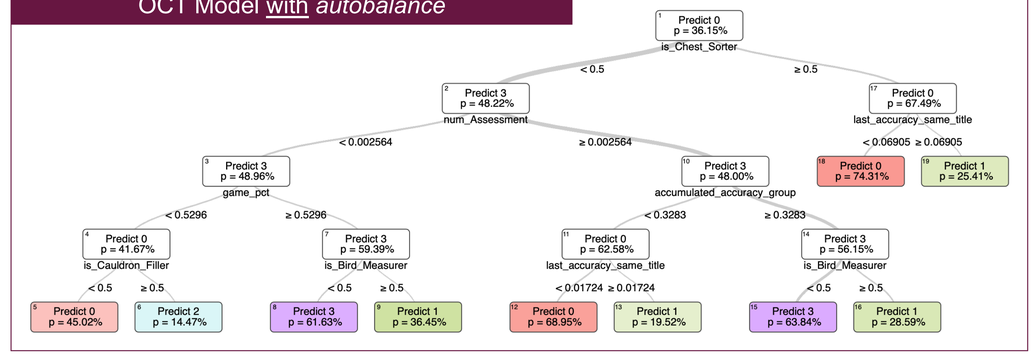
	Model					
	With autobalance		Without autobalance		Python-based	
Metric	OCT	OCT-H	OCT	OCT-H	Logistic regression	xgboost
Accuracy	0.569	0.548	0.620	**0.633**	0.602	0.607
Quadratic kappa	0.473	0.490	0.514	**0.540**	0.501	0.472

- Models **with *autobalance*** provide **more interpretable** solutions at the **expense** of **predictive power**
- **OCT-H without *autobalance*** proves to be the **best** model in terms of out-of-sample performance, surpassing *xgboost* and scoring **top 3%** in the official competition

Interpretability

- **OCT with** *autobalance* makes a **decision** based on **which assessment**, the **player's experience** on that assessment and the **overall performance** of the player
- **OCT without** *autobalance* **includes** the **assessment average difficulty** and **specific event** performance
- **Both** OCT models are able to **identify** the **hardest assessments**

OCT Model with *autobalance*



OCT Model without *autobalance*