Report: Datascience for AdTech 2018
Prediction of view time for a video of publisher

Yanlin DU Yu LI Ling JIN

21 mars 2018

1. Business case

The watched time of online advertising video provides a simple way to understand how long the video ads are being watched by potential customers and which videos keep their attention longer, even if the ad is skipped. This gives an additional insight into how people engage with the ads, beyond impressions and views.

For example, someone may decide to skip an in-stream ad after watching for 20 seconds, which (if they didn't click on any part of the ad) won't be counted as a view. We can evaluate and compare the watch time of different ads, regardless of skips or views.

The prediction of watched time of an advertising video is meaningful because strategies of campaigns and inventories can be optimized according to the prediction results. The main goal to get people to watch an advertising video can be better achieved.

Business metrics

Advertisers bid places from publishers to put their advertisements. Predicting the watched time of an advertisement helps publishers to know whether they can get paid by different kinds of advertisers and also helps advertisers to know whether a video campaign is successful.

Generally, several types of business metrics are used to evaluate it. We present three kinds of metrics. The first one is cost-per-view (CPV). Usually, a view means a complete view of an entire video. But other measures are considered too. Taking Youtube as an example, advertisers only pay to publishers when users watch the advertisements for at least 30 seconds. Or some other evaluations exist like click on a call-to-action overlay, a card or a companion banner, which we don't consider in our business case. However, ensuring users to watch an entire video or even more than 30 seconds is difficult nowadays. Some advertisers are satisfied when users watch for some some time but not the entire video. If a video is viewed for only one second, it doesn't make sense to advertisers. So we count only those videos which are viewed for no less than 2 seconds. We can increase this threshold value, but it depends on real cases.

In order to optimize the brand impact, advertisers can customize creative contents in the first few seconds so that the video is viewable and also makes impact on users so that an entire view is not required to impress users. This introduces the second metric viewable cost-per-view (vCPV). Compared to CPV, this metric includes another element, viewability. It takes not only a view into consideration but also the viewability, which is more realistic. In the case of outstream and pre-roll, users barely watch the entire video, so CPV is not suitable. While vCPV count that users have meaningful view, which is important to advertisers. As a result, vCPV is more appropriate especially for cross-format videos. Advertisers would be happy to see low vCPV which means their videos are viewed at a reduced cost.[1]

The third one is view rate, which is the number of users who watched videos divided by the number of times advertisers' impressions appeared. This implies how well the creative is performing. Higher the view rate, more attractive creatives are. [2] Consequently, for an advertiser, the key goal is to optimize video campaigns in order to get higher view rate and low CPV or vCPV.

More use cases

Some more ways to optimize video campaigns using provided data :

1. Estimate true value of views using CPV or vCPV and use true value to evaluate bid price and improve bid strategies.
2. Compare two similar video campaigns or ad groups with different video ads to see which ad earns more watch time. This can help advertisers to understand what kind of video content is

resonating with audience.

3. Try using the same ad creative in different campaign configurations to see which campaign better targets an audience that provides more average watch time per impression.
4. Watch time gives advertisers an idea of how many seconds of your ad creative is seen, on average. This can guide advertisers when they are creating new videos, and help inform the structure of video ad creative.

2. Data Exploration and preprocessiong

The training dataset has 3,000,000 observations, 20 features and one target to be predicted. The main goal is to predict the watched time of a video. Thus, the prediction problem attributes to be a regression problem. In the data preprocessing, we handle the timestamp feature by dividing it to hour, minute and second. Then NaN values are filled according to the data types and characteristics of features. Here, the average played seconds is filled by its mean rather than 0, because it can elevate a little bit performance during the modeling. Training data are ordered by timestamp. No normalization is applied to dataset for boosting algorithms, while for linear models, we use robustscaler to reduce outliers.

2.1 Correlations

At first, we get a glance at the correlation of features and the target. From the heatmap below, some significant correlations can be observed. As we can see, *advertiser_id* and *creative_duration*, *ua_browser* and *browser_version* have strongly negative correlations. While *campaign_id* and *creative_id*, *placement_id* and *website_id*, *ua_browser* and *ua_os*, *ua_device* and *ua_browser*, *creative_duration* and *seconds_played* are highly and positively correlated. Next, we would like to see the correlations between the target and other features in particular. Unfortunately, *seconds_played* has small correlation to other features. Only 6 features have correlations with it more than 0.05. The most correlated feature is *creative_duration* with a correlation 0.21. Then it is the *user_average_seconds_played*.

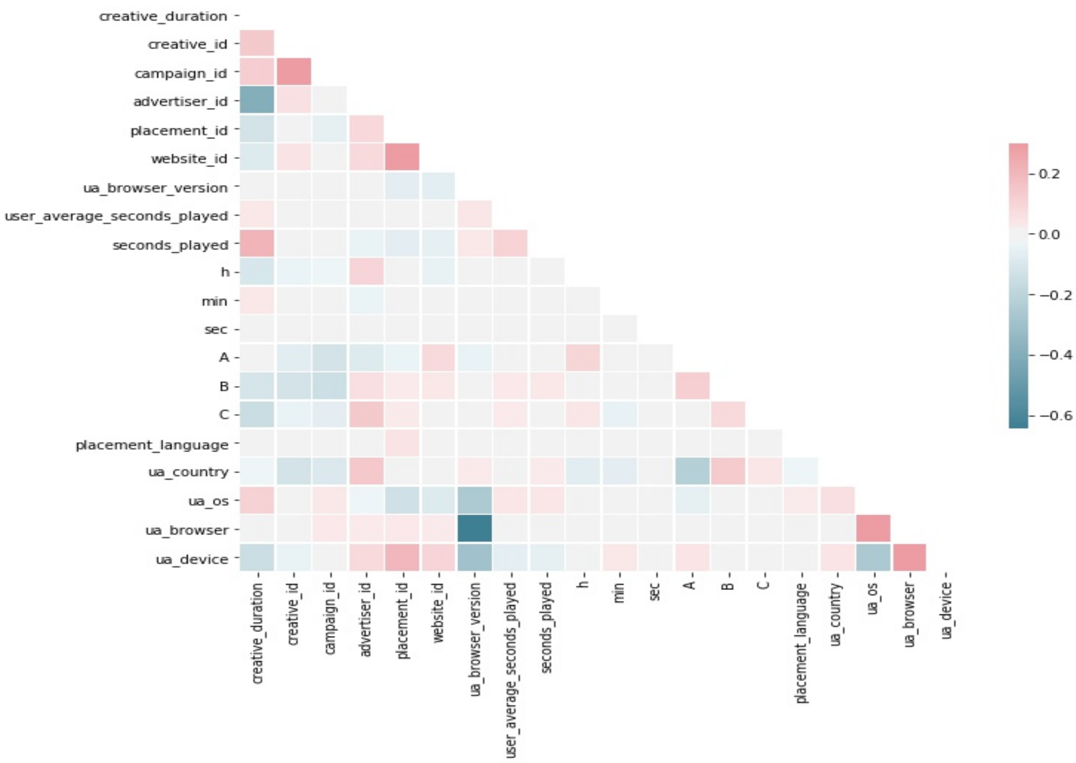


FIGURE 1 – Correlations of features

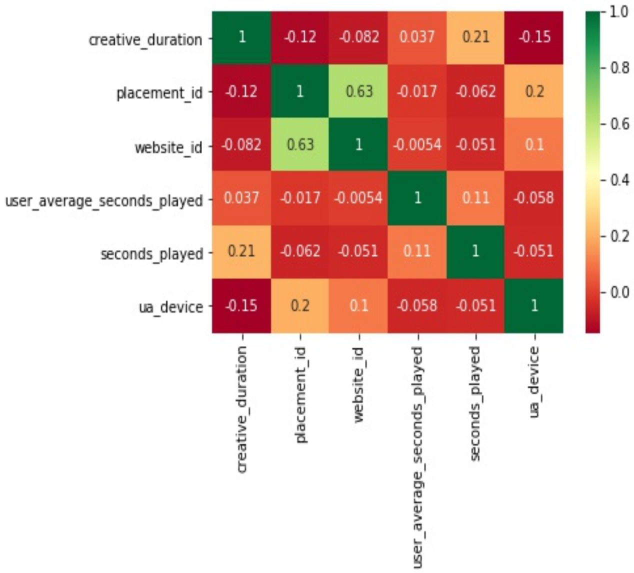


Figure 2 – Heatmap of the 6 most correlated features to target

If we discover more about the correlations between the target and these two features, as we can see the figures below, it is not surprisingly that the longer the duration of a video, the longer a user will watch in general. While, the averaged played seconds seems to have weak correlation to the target.

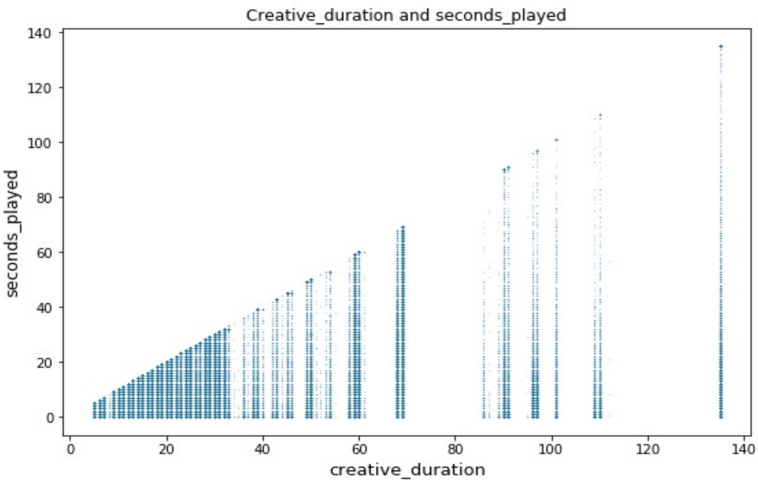


Figure 3 – Creative duration and played seconds

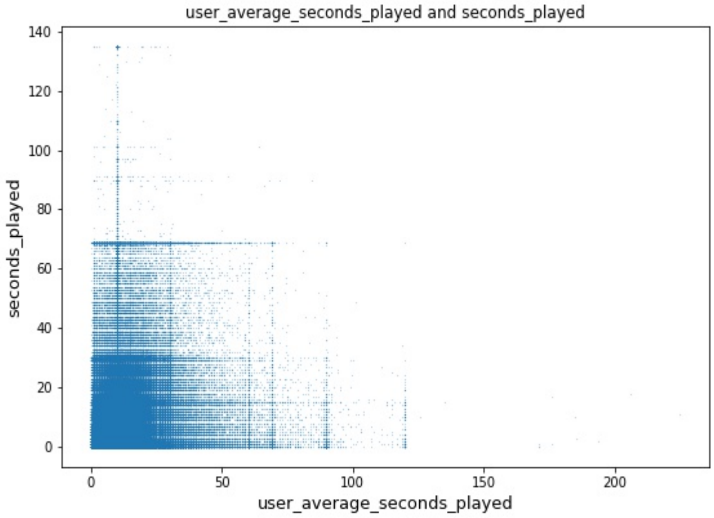


FIGURE 4 – Average watched seconds and played seconds

2.2 Distributions

The figure of the distribution fo creative duration shows an interesting fact that several peaks (14, 15, 29, 69 etc.) appears. It means that advertisements follow certain formats. The boxplot tells us that most advertisements are less than 60 seconds. While, the distribution of played seconds decreased very exponentially and has a long tail from around 30. These kind of distributions are difficult to predict well, we will mention that in the following sections.

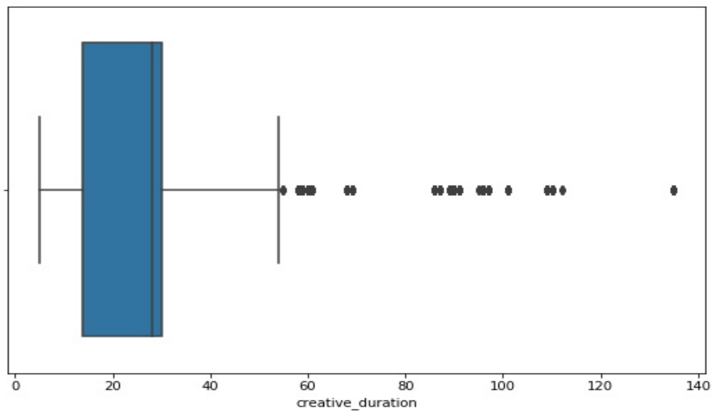


FIGURE 5 – Boxplot of the creative duration

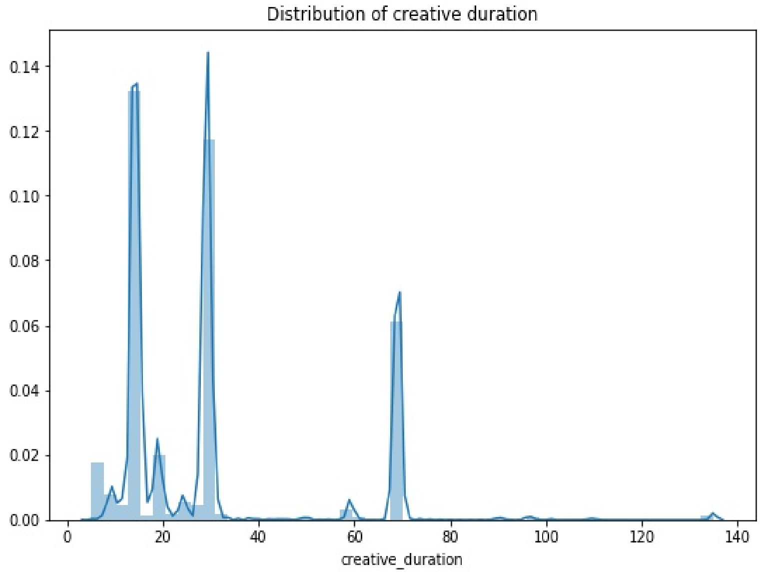


Figure 6 – Distribution of creative duration

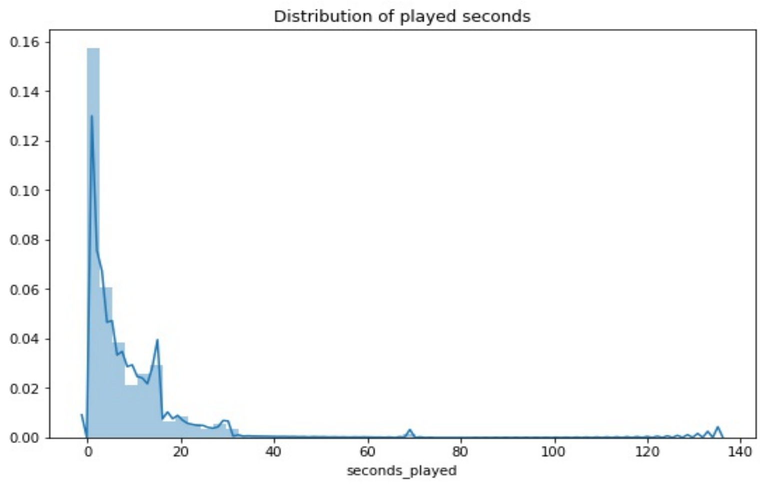
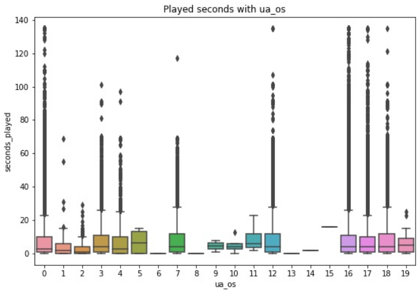


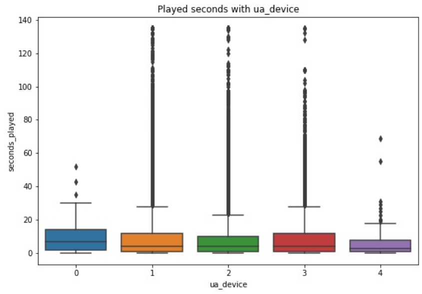
Figure 7 – Distribution of played seconds

2.3 Relations with other features

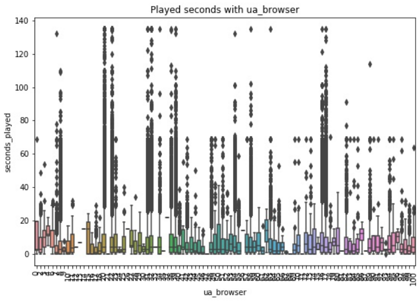
We select four features and draw boxplot of *seconds_played* with each feature. It turns out that for different values of features, the played seconds are different. Some values have significantly different behavior than others. For example, in *ua_os*, some operating system don't present watched videos or users are not interested in watching an entire video. Since we have lots of categorical features, tree model or tree based boosting models can be good choice for modeling.



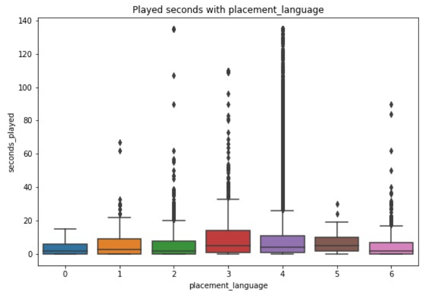
(a) *ua_os*



(b) *ua_device*



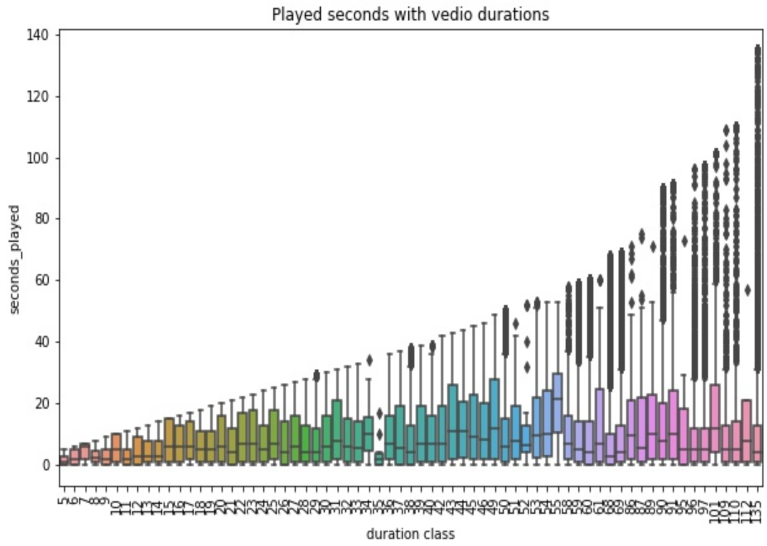
(c) *ua_browser*



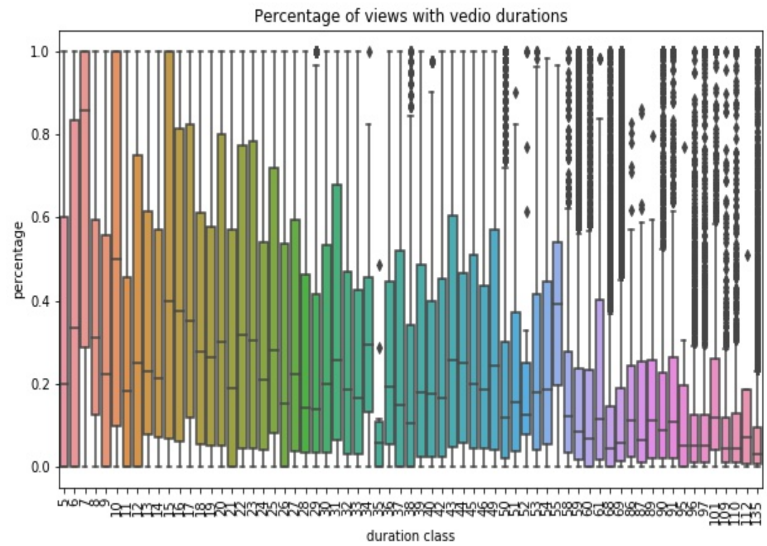
(d) *placement_language*

2.4 Watched time and creative duration

One of key goals to predict watched time is to optimize creative in order to get people to watch it. We consider 67 unique creative duration values which appear in dataset. We would like to see whether the behavior of users varies according to different creative durations. It turns out that most durations can keep certain users to watch entire videos. While some durations like 35s, 95s and 112s don't attract user until the end. The longer the advertisements are, the less user will finishing watching a video. It tells also an interesting fact that around 75% users would like only to watch less than 30s. The second figure who present the percentage of viewed time shows that 7s, 9s, 13s around 25% of their users watched entire ads. And the longer the advertisements are, the smaller average percentages are. These facts are very important to our business case because it affects the choice of business metric. CPV is not really suitable in this case because very few users watch for more than 30s or finish entire video. Maybe some users are impressed by brands even though they don't finish the advertisement. Thus, vCPV is more appropriate in our case.



(e) Played seconds with vedio durations



(f) Percentage of views with vedio durations

3. Evaluation

The evaluation metric for this competition is the RMSE :

$$RMSE^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2$$

where y_i is observed watched time and p_i is its prediction for line i

We split the training data into train and valid. Before submitting the solution to kaggle, we can evaluate the performance by valid score.

4. Modeling

In terms of modeling to predict the time a user will watch a video ad, the first thing for us to define is classify this challenge as a regression problem and during this challenge period we've tried several regression models. And next we will focus on introducing four models.

4.1 Random Forest Regressor

In the very beginning, we just use Random Forest Regressor to test the performance after processing all the data. And before submitting online, the first thing is to see whether the performance of chosen model is good or not. Therefore we split the subtrain and subvalid datasets based on the given training dataset and to test the quality of chosen model. For random forest, it is not easy to get overfitting. Here we take the performance of Random Forest as the baseline to tell us whether there is a need to continue tuning the parameter of the chosen model or not and then choose to train it with the whole training dataset or not. The chosen parameters in this model to be tuned are **n_estimators**, **max_features** and **max_depth**. And the model after tuning is as follows :

$$RandomForestRegressor(n\_estimators = 150, max\_features = 0.9, max\_depth = 7)$$

4.2 Gradient Boosting Regressor

Gradient boosting is a little bit better than random forest when we do validation in the subtrain and subvalid dataset. But we don't do additional parameter tuning and submit online a few times the best performance is around 8.47.

4.3 MLP Regressor

Known as Multi-layer Perceptron, a supervised learning algorithm used to have a good performance in some datasets, but this time it works much worse than random forest even after tuning parameter and we quit the further try in this model and didn't try to submit model using this model.

4.4 XGBoost Regressor

And we choose to use XGBoost and it gives us a performance obviously better than the three models above. The chosen parameters in this model to be tuned are **n_estimators**, **learning_rate**, **gamma**, **subsample**, **colsample_bytree** and **max_depth**. And the model after tuning is as follows :

$$xgboost.XGBRegressor(n\_estimators = 100, learning\_rate = 0.08, gamma = 0,$$

$$subsample = 0.8, colsample\_bytree = 1, max\_depth = 10)$$

And when we sort the time of training data before fitting it into the chosen model, it can even perform a little bit better in the submission score. We also tried ensemble model which combines linear regression with lasso, elasticnet, XGBoost and LGBoost and take hashing tricks to treat categorical features. But the performances are worse than that of XGBoost.

The feature importances of our xgboost model is showed as below. We can see that the most important features are "sec", "min", "user_average_seconds_played", "h", "placement_id".

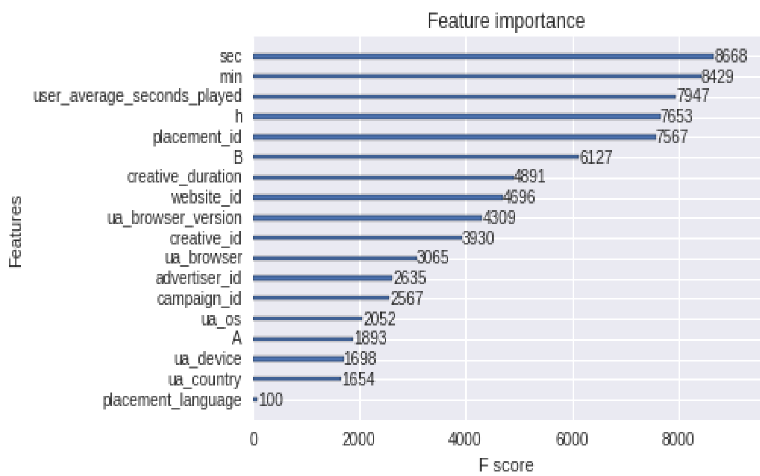


Figure 8 – Feature importance

TABLE 1 records the best performance of some models after parameter tuning.

Table 1 – Performance of some models

	RMSE
gradient boosting	8.47140
xgboosting	8.37187
log_xgb	8.96557
bagging_xgb	8.37949
ensemble	9.33943
xgb sorted training	8.37034
hashing	8.38813

4.5 vCPV

We choose the advertiser with the most records to calculate the vCPV for each campaign it has participated in. For advertiser_id = 1328, suppose the cost for each campaign is the same 100. Then for each campaign_id, we have their vCPV as **TABLE 2** shows below.

Table 2 – vCPV for advertiser_id=1328

campaign_id	view larger than 2	vCPV
214483	38780	0.002579
214490	18127	0.0055166
214626	3588	0.02787
214627	27784	0.003599
214630	2532	0.039494
214631	61450	0.001627
214633	16228	0.0061621
214634	14	7.1428571
215237	73	1.36986301
215312	141	0.7092198
215313	1848	0.054112
215963	150	0.0666667
215964	459	0.2178649
215967	479	0.2087682

5. Pros and cons

After data exploration and preprocessing, our final choice of model for this challenge is XGBoost. XGBoost is called as extreme gradient boosting, an implementation of gradient boosted decision trees designed for speed and performance. The pros and cons of our choice are as follows :
Pros :

1. It gives best performance in our test for validation data and the online submission part.
   As we have mentioned in the Table 1, we can easily find that XGBoost shows obvious advantage than other models.
2. The execution speed is quick.
3. This model is robust to outliers in the inputs.
4. In data preprocessing, categorical variables are not necessarily transformed to one-hot encoding. As most of our features have large cardinality, it is easier to handle categorical variables with only label encoder.

Cons :

1. It can overfit when faced with features with many categories. And in this regression problem, the preprocessing data have many different categories.

2. It gives large variance.
3. It shows bad performance when the watched time is big. Because this model tends to have limited predictive performance.

Bibliographie

[1] Viewable Cost Per View : The Ultimate Metric For Video Advertising `https://adexchanger.com/tv-and-video/viewable-cost-per-view-ultimate-metric-video-advertising/`.

[2] Optimizing your video marketing campaigns `https://www.youtube.com/yt/advertise/resources/optimizing-your-video-marketing-campaigns/`.