Bitrate quality in cloud gaming

Roman Vetrin
r.vetrin@innopolis.university

1 Motivation

Cloud gaming market is rapidly growing and is expected to expand by 34 percent in the following years. Thus, understanding of stream quality and bitrate behaviour becomes more important issue with growing demand and higher infrastructure pressure. In this paper we will look into raw data of a cloud gaming service in Russia and try to solve the problem of detection of poor stream quality sessions.

2 Data

In this paper we analysed two datasets prepared by a local cloud gaming company. Datasets contain basic statistical data on each gaming session (i.e. fps rate, dropped frames, package round-trip time, etc.). Our target features are factual bitrate during the session and a binary representation of stream quality. Each dataset contains more than 600 thousand observations. Dataset has been prematurely separated into training and test datasets.

3 Exploratory data analysis

Data presented contains flags of numerous outliers. For example, maximum values of many features strongly exceeds respective 75 percent quantiles. Please, refer to the **Table 1** for details:

Table 1. Features' overview

Feature	Mean	75 per. quantile	Max
fps std	1.72	2.23	307.17
rtt mean	49.62	55.90	12898
rtt std	12.76	4.95	40721
dropped frames mean	0.18	-	540
dropped frames std	0.47	3.16	202

Thus, we need to further visualize the data and find outliers as those might add some non-representative behaviour to our model.

According to the correlation matrix (refer to **Section 2.1** of attached Jupyter-Notebook), most of the features do not correlate to our target feature. Feature "bitrate mean" and "bitrate std" however does correlate with our target. However, using such data will provide limited research value. Thus, we will not include those features into our model.

Next, we visualized our data against each other and the target feature to understand any common trends and potential additional features (refer to **Section 2.1** of Jupyter-Notebook attached.).

4 Task

In order to perform our predictions, we are going to train a several model and then evaluate the results using the test datasets. To achieve that, we will be using linear regression models with Ridge and Lasso regularization as well as those with polynomial features.

4.1 Regression

For bitrate prediction, we will consider the following models:

1. Simple Linear Regression model
2. Simple Linear Regression model with Ridge regularization
3. Simple Linear Regression models with polynomial features 2nd - 5th degree
4. Simple Linear Regression model with polynomial features 4th degree with Ridge regularization

Note that we do not use Lasso regularization as it performs slightly worse according to our test in **Section 2.3** of the Jupyter-Notebook attached.

4.2 Classification

As stated before, the target feature for classification is highly imbalanced which may strongly affect outcome of our predictions. Thus, we will use 'weighted' algorithms which put higher coefficient to observations, representing low-presence class. For stream quality classification, we will consider the following models:

1. Weighted Logistic regression model with l1 regularization
2. Weighted Logistic regression model with l2 regularization
3. Weighted Logistic regression model with Ridge regularization
4. Weighted Logistic regression models with polynomial features 2nd - 7th degree
5. Logistic regression model with l2 regularization

5 Data Imbalance

Classification dataset suffers from significant class imbalance having very few positive classes. This endanger models' results as false negative errors have very limited weight in an error function. We have at least two options to counter this:

1. We may train our model on smaller dataset in which label count is approximately the same for each class.
2. During the model training, we may apply some weights to our smaller class so observations with that class would contribute more.

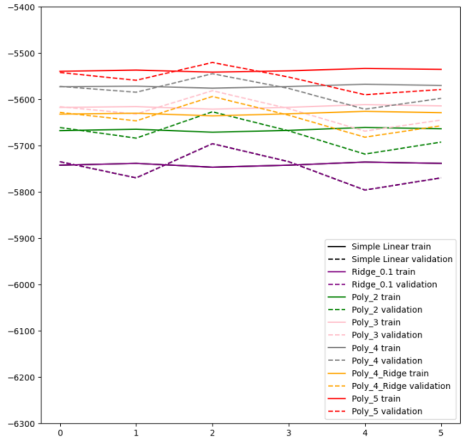
As the first option would deplete our dataset ten-fold, we will focus on the second option. Accordingly, we will apply weighted coefficients to one of our metrics: precision score as can be hardly gained due to true positive number will be much lower than false positive in the most cases.

6 Comparison of selected ML models

6.1 Regression

For cross-validation we will be using six folds, thus each model will be trained six times. As a result, polynomial regression showed the best performance with preservation of stability across different training datasets. Please, see the **Figure 1** for more details.

Figure 1. Regression cross-validation results



Overall, all models show enough stability to try predict our test data. To evaluate our final models, I am using two metrics - r2 score and root squared mean error. The first metric show universal indicator ranging from zero to one (in the most cases) and the second can represent absolute value of our models' errors. The results of the evaluation are presented in the **Table 2**.

Table 2. Regression test scores

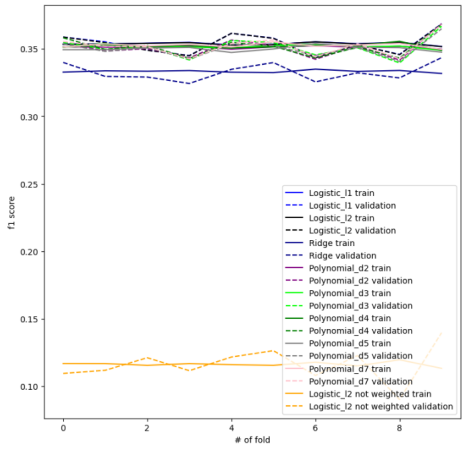
Model	R2	RSME	MAE
Simple Linear	0.100	5666	4318
Polynomial 4th	0.119	5605	4205
Polynomial 4th Ridge	0.120	5601	4238
Polynomial 5th	-0.200	6543	4215

We can note that Polynomial model with 5th degree is clearly overfitted as it shows decrease in outcome on the test dataset compared to training dataset (refer to **Section 2.3** of the Jupyter-Notebook attached). Other models demonstrate similar results, however, models are severely underfitted as r2 score is very low and mean absolute error is quite high compared to mean value of the target feature.

6.2 Classification

For cross-validation we will be using ten folds, thus each model will be trained ten times. As a result, 5th degree polynomial models showed the best performance with preservation of stability across different training datasets. Please, see the **Figure 2** for more details.

Figure 2. Classification cross-validation results



For final evaluation we will be using accuracy score, weighted precision score, unweighted recall score and unweighted recall score. Total evaluation is presented in the **Table 3**.

Table 3. Classification test scores

Model	Acc.	Prec. w.	Recall	F1
Logistic l2	0.900	0.923	0.501	0.390
Logistic Ridge	0.894	0.922	0.511	0.383
Polynomial 4th	0.885	0.920	0.506	0.363
Polynomial 5th	0.884	0.921	0.522	0.367
Polynomial 7th	0.900	0.923	0.506	0.395

According to the results, it seems that Logistic regression with 7th degree polynomial features demonstrate the best F1 score and the best accuracy. It seems that the model is still underfitted though and more complexity can be added without loosing performance on test set.

7 Conclusion

As a result, we trained a few models which and evaluated them on the test set with a quite depressing score results. This demonstrates that with the given data and limited set of tools used during the research, we cannot reliably predict bitrate. I note however, that outlier search has slightly improved our models as it made them more stable.

To understand patterns of a bitrate, we probably need to find other features for evaluation or use more sophisticated tools.