Machine Learning Task: Cloud Gaming

Viacheslav Blinov
v.blinov@innopolis.university

1 Motivation

Cloud gaming companies tend to expand and deliver quality service. Thus they collect gaming sessions data and analyse it to improve users experience. In this paper, an Innopolis University partner company issued data-sets and provided a task to predict continuous bit-rate and classify stream quality.

2 Data

Throughout this paper two data-sets are being analysed for training and testing of regression algorithms. The data-sets are split into two folders, both containing a read-me file and two .csv files, representing train and test slices of the data. Totally there are over 600 000 observations for each task. The concrete features are derived using statistics, such as mean, standard deviation and maximum (i.e. "fps_mean", "dropped_frames_std", etc). The data-set contains numerical and categorical attributes.

3 Exploratory data analysis

First of all, the data-sets were concatenated together to reduce the amount of operations and capture the whole context.

Maximum values of several features exceed 75 % quantile, which indicates outliers. Refer to the **Table 1** for more details.

Secondly, data visualization was performed and analysed. Thereby outliers were found which do not impact models. Furthermore, a correlation matrix (heat-map) showed few features' dependency regarding the target variable. Such features were kept but transformed. While it is crucial to predict bit-rate, it is possible to drop "fps_mean" and "fps_std" features, since they are derived from the target which makes no sense.

Thirdly, a pair plot was used to showcase trends and ideas for features engineering.

Table 1. Data-set description

Feature	Mean	75 %	Maximum
fps_mean	35.1	43.4	125.8
fps_std	1.6	2.2	307.1
rtt_mean	48.8	56.4	12898.4
rtt_std	12.8	5.3	40721.9
dropped_frames_mean	0.1	0.0	540.0
dropped_frames_std	0.4	0.0	291.8

Classification data-set has target imbalance which leads major False Negative errors. It was chosen to set up model's hyper-parameters to balance the classes using its own strategy.

4 Task

In order to accomplish the task, it was necessary to train and test (validate) several regression algorithms. The algorithms were evaluated using the results acquired by test (validation) data-sets against linear and classification metrics.

4.1 Regression

The mandatory elements which are considered in the paper:

- Linear Regression.
- Ridge, Lasso Regularization.
- Polynomial Features, GridSearchCV.
- Cross-validation.

4.2 Classification

The mandatory elements which are considered in the paper:

- Logistic Regression, GridSearchCV.
- Ridge, Lasso Regularization (Cross-validation versions).
- Weighted, Non-weighted Variants.

Weighted algorithms put high coefficients into an observation, representing minor classes.

5 Models comparison

5.1 Regression

For cross-validation 9 K-folds were used. It means that each model was trained 9 times. As the result, polynomial regression (2rd degree) model revealed the best scores while keeping the stability across different train (validation) data-sets. Refer to the **Figure 1** and **Table 1** for more details.

Overall models showed enough stability when predicting using the test (validation) data. The following evaluation metrics were used: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R$\hat{2}$ Score.

R$\hat{2}$ shows a universal indicator ranging from 0 to 1 (common scenario). Root Squared Mean Error acts as a model error (tends to minimize). Mean Absolute Value acts as a task error. The results of the evaluation are presented in the **Table 2**.

According to the source code, Lasso regularization could be redundant as its performance is slightly worse.

Polynomial model was over-fitted and is low in Root Mean Squared Error on the test (validation) data-set. Thus it would be unstable on other data-sets. Otherwise, models seem severely under-fitted as R$\hat{2}$ score is low and Mean Absolute Error is high against the mean value of the target.

Figure 1. Regression cross-validation results

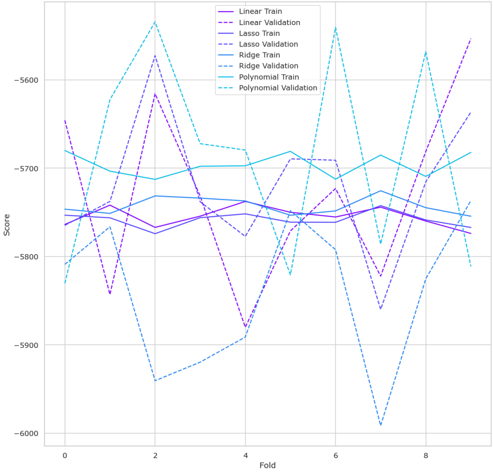


Table 2. Regression scores

Model	MAE	MSE	RMSE	$R\hat{2}$
Linear	4431.50	3.29e+07	5736.38	0.09
Lasso	4433.45	3.29e+07	5736.92	0.09
Ridge	4431.78	3.29e+07	5736.25	0.09
Polynomial	4380.29	3.25e+07	5702.67	0.108

Polynomial Features degrees may vary depending on the context. The higher the better principle does not always apply.

5.2 Classification

For cross-validation GridSearchCV was used as it estimates the best predictor automatically based on chosen scoring.

The next scores were used: accuracy score, precision score (weighted), recall score and F1 score. Total evaluation is presented in the **Table 3** or **Figure 2**.

Table 3. Grid Search score

Model	Acc.	Prec. (W.)	Rec.	F1
Logistic	0.93	0.92	0.93	0.91

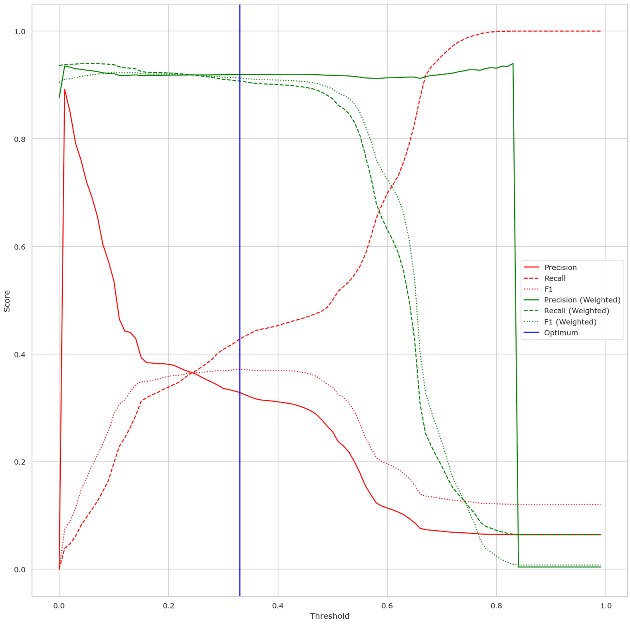
Refer to **Model Validation** GridSearchCV section from the attached Jupyter Notebook for extra details.

As the result, the best parameters for Logistic regression are: {"class_weight": None, 'max_iter': 200, 'penalty': "l2"}. The model's extra complexity can be added without loosing performance on test (validation) set.

6 Conclusion

Several models were trained and tested (evaluated) on the test-set (validation). The final scores are low-rank which demonstrates a limited set of tools and partial impossibility to predict bit-rate. Outliers removal helped the models, as it made them more stable. It is crucial to possess extra features for evaluation or use more advanced tools in order to understand patterns of the target bit-rate.

Figure 2. Logistic Regression cross-validation (weighted / unweighted), optimum


Score
Threshold
Precision
Recall
F1
Precision (Weighted)
Recall (Weighted)
F1 (Weighted)
Optimum