Machine Learning - Assignment 1

Eduardo Augusto Militão Fernandes
e.fernandes@innopolis.university

1 Motivation

Video games are an interactive medium, and so, it is crucial for the user to get a response on the screen to their input as fast as possible, making a high number of frames per second (FPS) and low input delay essential in any game setup. In this sense, to achieve such requirements, cloud gaming depends on a fast and reliable internet connection, which is not available for many of users, and thus providers of this service must be able to adapt to different kinds of connections. Therefore, being able to predict the bitrate to send the next data packets for an user in function of the characteristics of their connection in order to minimize lost of data enables the provider to optimize the use of its network resources and ensure a quality experience. Furthermore, classifying the quality of the stream allows them to measure user satisfaction and take actions to improve it when seen necessary.

2 Data

The regression data consists of 9 features measured during the stream, all of them numerical, and one also numerical target variable: the ideal bitrate to send the next data packets in the stream. The features are: mean number of FPS and its standard deviation, mean value of round-trip-time and its standard deviation, the mean number of frames lost (dropped frames), its standard deviation and maximum value, the mean bitrate used to send previous packages and its standard deviation.

The classification data consists of 11 measured features of the stream related to its quality and configuration, being 9 numerical and 2 categorical, and one binary target variable: the quality itself. The features are: mean number of FPS and its standard deviation, number of FPS lags that occurred, mean value of round-trip-time and its standard deviation, the mean number of frames lost, its standard and maximum value, auto bitrate ternary indicator (off, full or partial), auto forward error correction (FEC) mode binary indicator (partial or off) and auto FEC mean value.

3 Exploratory data analysis

3.1 Regression Task

Initially, the Spearman correlation matrix was calculated for all the features and target variable. Through it, it was observed that "bitrate_mean" was extremely correlated to the target. On the contrary, the three features related to the number of dropped frames, "fps_std" and "rtt_std" showed correlation values of less than 0.15 in absolute terms. Furthermore, when plotting the independent variables against the target, these features presented a large concentration of data points on the same region of the plot. Finally, a PCA projection was plotted against the target variable, and it was clear through the shape of the plot that the feature "bitrate_mean" had the biggest influence on the new subspace.

For the above mentioned reasons, plus the fact that over 90% of the "dropped frames" columns had zero value, it was decided to select only the following features for the task: fps_mean rtt_mean bitrate_mean bitrate_std.

3.2 Classification Task

Firstly, the categorical features were encoded using One Hot Encoder, in order to be able to verify how each category correlates to the target variable. Secondly, the Spearman correlation matrix was computed, and it was observed that none of the categorical features have any correlation to the quality of the stream. In fact, only the features related to "dropped frames", "fps_std" and "fps_lags" showed a significant correlation, being the latter the most correlated feature with a score of 0.32. Next, a PCA transformation was also performed on the data, and when plotting the first two principal components against each other, it is observed that a large number of positive points can be separated from the negative ones, but most of them are mixed together in one region of the plot. This fact indicates that a naive approach will likely obtain high precision, but low recall.

Finally, since "dropped frames mean", "dropped frames std" and "dropped frames max" are highly correlated to each other, only the first one was kept out of the three. Therefore, the remaining variables were: fps_std, fps_lags and dropped_frames_mean.

4 Task

4.1 Regression

The regression task is to, given a series of n-dimensional data-points, estimate a function $f : \mathbb{R}^{n-1} \to \mathbb{R}$ that, for the values in the first $n-1$ dimensions of a data-point, predicts the value in the nth dimension, and minimizes function $\mathcal{L}$ that measures the difference between the value returned by f and the true observed value for all data-points.

4.2 Classification

For the classification task, the goal is to estimate a function $f : \mathbb{R}^n \to [0, 1]$ that, given a n-dimensional data-point, returns the probability of it being a positive sample. Such function must also be one that minimizes another function, $\mathcal{L}$, that indicates how close the prediction returned by f is to being correct.

5 Results

5.1 Regression

The metrics chosen for this task were: Mean Absolute Error (MAE), Rooted Mean Squared Error (RMSE) and R2. The models tested were: Simple Linear Regression, using the feature "fps_mean", Polynomial Regression of third degree and Lasso Regression. The performance of each model on the training dataset was measured through KFold cross-validation, with 5 splits. Then, the models were executed on the test data, using all the training data for training. The results are presented in Tables 1 and 2. The Simple Linear Regression model severely under-fits the data, performing poorly in both training and test datasets. On the other hand, the two other models perform extremely well, not over- nor under-fitting, with Polynomial Regression presenting slighly better MAE and RMSE scores than Lasso.

Table 1. Results over cross-validation

Model	MAE	RMSE	R2
Simple Lin. Reg	4677.16	5923.84	0.04
Polynomial Reg.	1076.99	1972.03	0.89
Lasso	1105.92	1980.93	0.89

Table 2. Results over test dataset

Model	MAE	RMSE	R2
Simple Lin. Reg	4563.11	5863.63	0.04
Polynomial Reg.	1052.33	1944.10	0.89
Lasso	1078.32	1949.12	0.89

5.2 Classification

The metrics chosen for this task were: Accuracy, Precision, Recall and R2. The models tested were: Logistic Regression (LR) without Regularization, LR with L1 Reg., LR with L2 Reg. and One Class Support Vector Machine. The same procedure for testing used in the previous task was used on this one. The results are in Tables 3 and 4. Because the results achieved by all three Logistic Regression methods were virtually the same, they are only shown once on the forth-coming tables. The Logistic Regression methods achieved high precision but low recall, as initially predicted. The SVM model obtained different results, favoring its recall score. All models under-fitted, since none was able to achieve a high F1 score, indicating that the data may not be easily separable.

6 Outlier Detection

An outlier detection and deletion technique was performed in the classification dataset. For that, every row which contained a value whose absolute value was larger than 4 times the standard deviation of the corresponding feature was dropped. The results are in Table 5. The Logistic Regression methods were able to achieve a higher F1 with this step, but the SVM model had its performance hurt by it in all metrics.

Table 3. Results over cross-validation

Model	Acc.	Precision	Recall	F1
Logistic Regression	0.94	0.69	0.11	0.18
One Class SVM	0.51	0.04	0.51	0.07

Table 4. Results over test dataset

Model	Acc.	Precision	Recall	F1
Logistic Regression	0.94	0.71	0.12	0.21
One Class SVM	0.86	0.17	0.33	0.23

Table 5. Results after outlier removal on test dataset

Model	Acc.	Precision	Recall	F1
Logistic Regression	0.94	0.67	0.14	0.24
One Class SVM	0.80	0.12	0.32	0.17

7 Data Imbalance

Since the data in the classification task was so imbalanced, a Random Under Sampling technique was employed, in which only a proportion of the negative samples were used in training, in order to have the same number of positive and negatives samples in the dataset. The results are in Table 6. This step had a positive impact on the F1 score of the LR methods. However, the results achieved by SVM remained the same.

Table 6. Results after data balancing in test dataset

Model	Acc.	Precision	Recall	F1
Logistic Regression	0.86	0.23	0.52	0.32
One Class SVM	0.80	0.12	0.32	0.17

8 Conclusion

In this work, a regression and a classification task were solved using different ML models. In the former, the features used to predict the target variable were highly correlated to it, and thus it was possible to obtain models that performed extremely well. In the latter, a low-correlated feature set together with non-separable and imbalanced data resulted in models that were not able to achieve high scores in the selected metrics. Outlier detection and data balancing methods were used, which helped to improve the results in most models, but not by much.