Detecting Unusable Network Using Machine Learning

Abdulmueez Emiola
a.emiola@innopolis.university

1 Motivation

The motivation of this work is to predict the bit rate of the stream and classify the stream quality into good or bad using machine learning. This will enable us detect unstable network which can be used to take steps towards adapting the data streaming and minimizing data packet loss.

2 Data

The data for the regression task contains 9 independent variables and 1 dependent variable. The independent variables include:

- the mean and standard deviation of the data packet's round trip time corresponding to rtt_mean and rtt_std.
- the mean, standard deviation and the the maximum number of the dropped video frames during data transfer corresponding to dropped_frames_mean, dropped_frames_std, dropped_frames_max.
- the mean and standard deviation of the frames per second corresponding to fps_mean, fps_std.
- the mean and standard deviation of the bitrate corresponding to bitrate_mean and bitrate_std respectively.

The target variable is a prediction of the bit rate and can take any values within the set of all positive real numbers. Our data contains 379021 measurements of these features.

The data for the classification task contains 11 independent variables and 1 dependent variable. The independent variables include

- the mean, standard deviation and lags of the frame per second represented by fps_mean, fps_std and fps_lags respectively.
- the mean and standard deviation of the round trip time represented by rtt_mean and rtt_std respectively.
- the mean, standard deviation and max of the dropped frames per second represented by dropped_frames_mean, dropped_frames_std and dropped_frames_max respectively.
- the mean and state of the auto forward error correction represented by auto_fec_mean and auto_fec_state respectively.
- the auto_bitrate_state which represents the state of the auto bit rate.

The auto_bitrate_state is a categorical variable that takes values from the set { off, full, partial } while the auto_fec_state is a categorical variable that takes values from the set { partial, off }. The target variable is the stream_quality and takes values in the set {0, 1} corresponding to {good, bad}. The data is highly imbalanced with over 93% of the observations corresponding to good stream quality.

3 Exploratory data analysis

During data analysis, about 2.2% of the classification data and 0.1% of the regression data are duplicates that we dropped in order to avoid biasing the model and reduce computation cost. Both data also contained highly correlated features and excessive multicollinearity. To reduce the degree of multicollinearity, dropped_frames_max feature was dropped in both data. It was also noticed that bitrates_mean correlates highly with the target variable. It was also noticed that certain features had larger scales when compared to the others, this suggests that the data should be scaled. Based on the above analysis, the categorical variables were encoded using one hot encoding and also scaled our data using a standard scaler.

4 Task

For the regression problem, the task is to predict the bit rate to send data packets in a stream using three or more appropriate machine learning models and then select the appropriate machine learning amongst them.

For the classification problem, the task is to select an appropriate machine learning algorithm for detecting stream quality and test the effect of removing outliers and balancing the data on the result.

4.1 Regression

We start solving our regression task using a linear regression model. In order to ensure that the model isn't under fitting, we transform the features into polynomial features and retrain the linear regression model using the polynomial features. To select the appropriate degree for the polynomial features, grid search algorithm was used and the result of the grid search produces 2 as the best degree. Eventually, in order to reduce the effect of over-fitting caused by transforming the feature space to a more complex space, L1 regularisation was introduced to the linear model by using the Lasso regression.

4.2 Classification

To solve the classification problem, logistic regression algorithm was used.

5 Results

5.1 Regression

The coefficients corresponding to the different features were obtained from the linear regression model(Table 1). This shows that the bitrate_mean influences the target variable far more than the other features. This is inline with the inference drawn from data analysis.

Table 1. Top 3 Coefficients of each feature in the linear regression model

Coefficients	Value
bitrate_mean	5613.283466
bitrate_std	209.500240
rtt_mean	-54.434663

Table 2. Model Results

Metric	Training	Test
Linear Regression		
R2 Score	8.931187e-01	8.934829e-01
MSE	3.935577e+06	3.798473e+06
MAE	1.104711e+03	1.077926e+03
Polynomial Regression		
R2 Score	8.945291e-01	8.941205e-01
MSE	3.883643e+06	3.775736e+06
MAE	1.075497e+03	1.052732e+03
Lasso Regression		
R2 Score	8.945260e-01	8.941292e-01
MSE	3.883758e+06	3.775736e+06
MAE	1.075675e+03	1.052730e+03

From the evaluation results of the models displayed in table 2, it can be seen that the polynomial regression produces better scores than the linear regression. It produces higher r2 scores, lower mean squared error and lower mean absolute error. It can also be seen that the lasso regression (L1 regularized polynomial regression) reduces the training R2 scores and increases the test R2 scores. It also gives an increased training MSE and MAE while also giving a decreased test MSE and MAE. This shows it reduces over-fitting.

5.2 Classification

From the recall value in Table 3, it can be seen that the logistic regression only correctly identified the bad streams 13% of the time. It can also be seen that upon classification of a stream as being bad, It has a 70% probability of being correct. Although the accuracy is over 90%, it can be seen that the model performs badly on predicting bad streams. This shows that the imbalance in the data is most likely skewing the accuracy value.

Table 3. Performance of the model in classifying bad streams

Model	Acc.	Precision	Recall	F1-score
Logistic Regression	0.94	0.71	0.13	0.21
After Removal of outliers	0.94	0.70	0.13	0.22
After balancing data	0.83	0.21	0.58	0.31

6 Outlier Removal

To improve our classification model, outliers were removed using local outlier factor. In table 3, it can be seen that the model performs quite similarly to the original regression model, showing that the removal of outliers doesn't have much effect on the performance of our model.

7 Data Imbalance

Data balancing is performed by combining oversampling with under-sampling using the SMOTEENN algorithm. After training the model with the re-sampled data, it can be observed that there's a significant increase in the recall value with a sharp decrease in the precision and accuracy value when classifying bad streams.

8 Conclusion

For Regression task, the lasso regression has the best performance, hence it should be used in predicting the target variables.

For the classification task, The task is to detect unusable network using machine learning, the model trained on the balanced data helps to greedily achieve this task. It's capable of identifying 58% of bad streams. The downside is that if a stream is classified as being bad, It's only correct 20% of the time.

To generously classify bad streams, the model trained without or with removal of outliers can be used. With these models, the probability that a stream classified as being bad is actually bad is 0.7 but It hardly classifies streams as being bad as it's only capable of identifying 13% of the bad streams. The choice of the model to be used is dependent on whether we wish to greedily or generously classify the streams.