ISyE6414 Project

Video transcoding analysis and prediction

Byron Kim, Minghan Xu,
Shishir Suman

Table of Contents

1. Introduction

I. Motivation

With recent technological advancements in entertainment and their adoption by a wider section of society, digital media has evolved in numerous ways. Companies working at the cross-section of this industry have developed their own devices and operating formats, and various content representation formats have therefore evolved. The need of interoperability has become paramount in the industry. Especially in computer vision, including videos and images, where consumers need seamless interaction between different devices, transcoding between different media formats and associated time are key concerns. The project uses a video dataset to understand the relationship between transcoding time and video characteristics. Results of such a study will be crucial for understanding the resource requirements and providing an efficient transcoding service.

II. Background

Transcoding is the conversion of one coded format to another. Following flow chart displays the sequence of steps involved in the process:

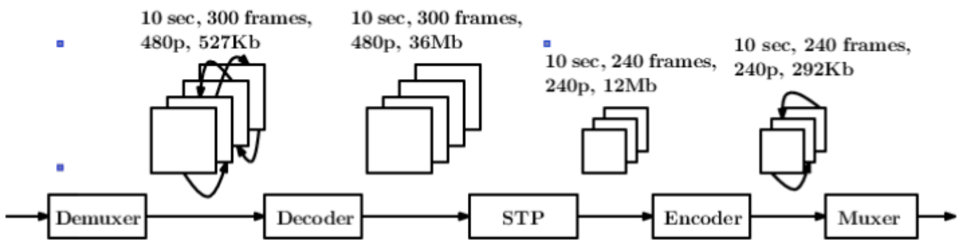


Figure F1: Transcoding process for a video/ audio

Coder algorithms are used to encode video and audio data when a multimedia file is created. A muxer puts the streams together into a file (container). To play the file, a demuxer takes apart the streams and feeds them into decoders to obtain the video and audio data.

General characteristics defining any video are its codec, resolution, bitrate and framerate. Each of these characteristics can have multiple possible values, as illustrated in the table below and explained in the section below.

Parameter	Value
Codec	H264, Mpeg4, Vp8, H263
Resolution	144p, 240p, 360p, 480p, 720p, 1080p
Bitrate	56k,109k, 242k, 539k, 820k, 3000k, 5000k
Framerate	12, 15, 24, 25, 29.97

Table T1: Possible feature characteristics of any video

Codec: Codec defines the coding standard used for any video. It's an electronic circuit or software that compresses or decompresses digital video. In the context of video compression, "codec" is a concatenation of "encoder" and "decoder".

- *flv:* flash video aging container that is being phased out, used to be the de facto standard for web-based streaming video.
- *MPEG-4:* Advanced Video Coding (MPEG-4 AVC) is a block-oriented motion-compensation based video compression standard. It is one of the most commonly used formats for the recording, compression, and distribution of video content
- *H.264:* is a new standard for video compression which has more advanced compression methods than the basic MPEG-4 compression. H.264 has high compression rate and is about 1.5 to 2 times more efficient than MPEG-4 encoding
- *VP8:* is an open and royalty free video compression format owned by Google. (mpeg-4) is not royalty free, slightly better performance than VP8

Bitrate: A video bitrate is the number of bits that are processed in a unit time. A higher bitrate will accommodate higher image quality in the video output. At the same bitrate, video in a newer codec such as H.264 will look substantially better than an older codec like H.263. For example - the bitrate for an HD Blu-ray video is typically in the range of 20Mbps, standard-definition DVD is usually 6Mbps, high-quality web video often runs at about 2Mbps, and video for phones is typically given in the kilobits (Kbps). It's the only thing that defines the video file size.

Resolution: Resolution of a video represents its sharpness. In short, it represents the number of pixels in horizontal and vertical direction. For example, a 480p video is made of 480 lines stacked on top of another with each being 852 pixels wide. Similarly, a 720p video has 720 lines, each being 1280 pixels wide. This means that it is more than twice as sharp as a similar 480p video and can be viewed on a much larger screen. However, since height and width are generally standardized, one can expect a strong correlation in height and width attributes of a video. We will investigate this relationship during exploratory analysis.

Frame rate: Frame rate (measured as frames per second) is the rate of displaying consecutive images in a video. For a human vision, the frame rate will define video's sensitivity. Though it varies across individuals, human visual system can generally process 1 to 5 images per second and perceive them individually. Higher frame rates are perceived as continuous motion.

Another important aspect of video transcoding is the frame characteristics involved. A video consists of sequence of distinct frame types: keyframes (I), forward-predicted frame (P) and bi-directionally predicted frame (B). Based on their characteristics, these frames have a different role to play in video compression. An illustration of frame types ordering for MPEG has been shown in the figure below:

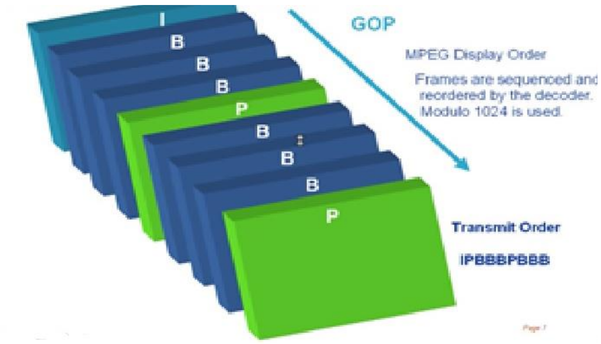


Figure F2: Frame type characteristics for a video transcoding

- I-frame (Intra-coded picture) is a complete image, like a JPG or BMP image file. Because of its nature, it requires more information to be stored in the output file as compared to P and B frames.
- P-frame (Predicted picture) holds only the changes in the image from the previous frame. For example, in a scene where a car moves across a stationary background, only the car's movements need to be encoded. The encoder does not need to store the unchanging background pixels in the P-frame, thus saving space.
- B-frame (Bidirectional predicted picture) saves even more space by using differences between the current frame and both the preceding and following frames to specify its content.

Based on the discussion so far, it's easy to infer that the number of total frames is equal to the sum of the number of i, b and p frames. Similarly, the total size of video equals the sum of sizes of three types of frames. Such characteristics create linearly dependent variables and hence, they need to be removed during exploratory analysis.

Much of the focus of transcoding has been on reducing the bitrate to meet the output encoding needs. This has led to conversion methods that facilitates efficient transportation of video data.

Bit-rate reduction is to minimize bit-rate while maintaining low complexity and high quality at the same time. There are several techniques used in bit-rate reduction. One way to achieve this is to decode the video bit stream and fully reencode the signal at the output rate. This can be achieved by calculating new motion vectors and mode decisions for every MB at the new rate. Even more efficient techniques have been created to optimize for efficiency. With increase in mobile device use, processing power and limited display became key constraints researchers needed to account for.

Previous related work has been done by Akademi University, Finland to find the optimized performance of large-scale video transcoding. Their data was collected by web crawling on Youtube and recording key video characteristics. Data on over a million unique videos was recorded. They analyzed this data to make useful statistical insights and created video transcoding time prediction model. This study utilized three different machine learning algorithms.

Algorithm	R^2	Absolute error	Training time	Testing time
NN	0.958	1.757 ± 2.834	7 ± 2 min	5 ± 2sec
SVR	0.942	1.484 ± 3.594	40 ± 3 min	13 ± 6sec
LR	0.411	7.233 ± 9.997	15 ± 2 sec	2 ± 1sec

TABLE II
COMPARISON OF PREDICTION ALGORITHMS

Other research in this field includes the focus on the social aspects of videos such as popularity and likes to study consumer access pattern, active life pattern, growth pattern and request pattern. Also, network edge analysis has been done to study YouTube traffic characteristics in social networking.

III. Contribution

All the team members (Byron Kim, Minghan Xu, Shishir Suman) have contributed equally to the project.

2. Methods

I. Data Source

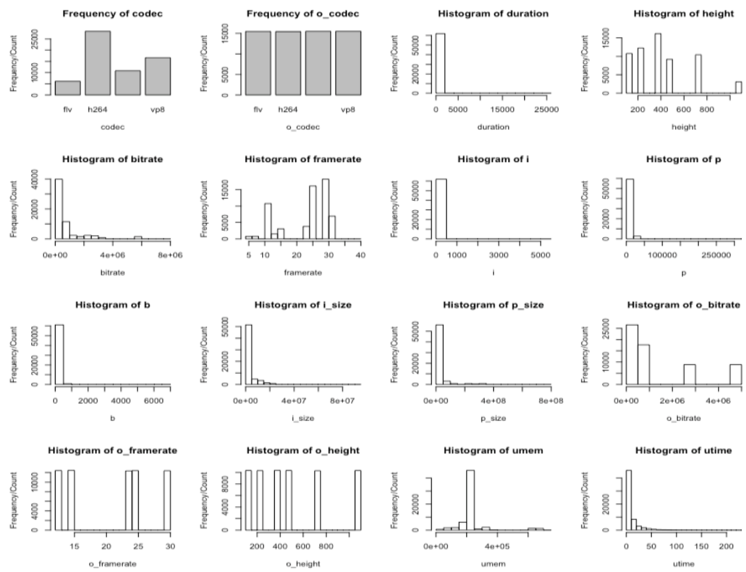
The dataset has been obtained from UCI Machine Learning Repository *Online Video Characteristics and Transcoding Time Dataset*. It features 20 columns containing input and output video characteristics along with memory allocation and transcoding time. Descriptions of the features have been summarized in appendix A.1.

The outcome for the study is video transcoding time (a continuous variable) and has been derived from performing ~68,000 video transcoding tasks on an Intel Core i7 Quad-Core processor with specified input and output video characteristics

II. Exploratory Analysis

This section discusses the exploratory analysis performed on the input variables. Input video codec (codec) and output video codec (o_codec) characteristics have been transformed into factor variables as these have categorical inputs such as flv, h264 etc. Based on the frequency distribution plot, output codec values are uniformly distributed. However, for the input video codec, h264 is the most commonly occurring type. This can be expected as this is new standard for video compression and has advanced compression methods. So, there is a possibility that because of this reason, more data for this form has been collected.

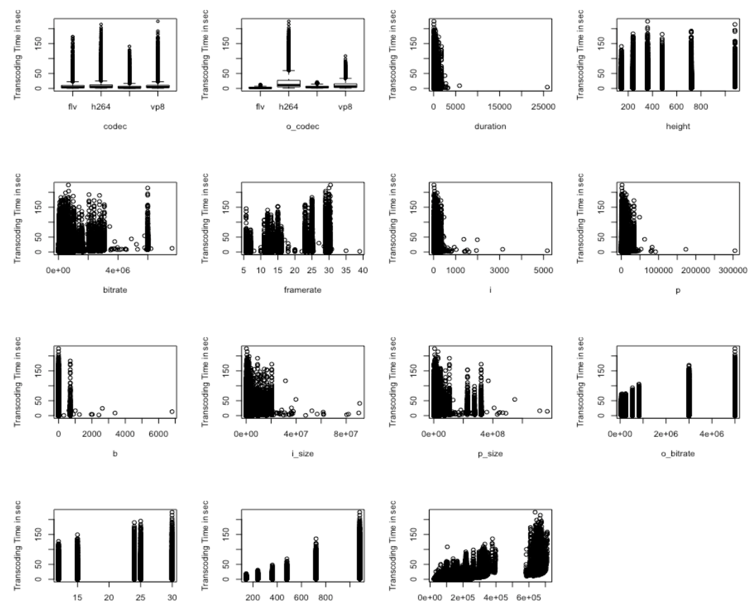
For multiple other predictors, the distribution seems to be discretized. For example, in the plot for video height, one can observe bars only at specific values. This is seen because there are commonly accepted sizes for such variables and though its technically possible to create finer intervals, it’s rarely seen practically. However, these are still ratio variable characteristics and hence, they have not been converted into factors for model generation. Another important thing to notice is that for certain predictors, skewed distributions are being observed with majority of values lying in a small range. Majority of these trends are driven by the popularity and adoption of specific features values.



Certain leverage points are also observed in the following plots. However, they will be evaluated later using Cook’s distance to confirm if they will be influential to the model output.

Box plots for the factor variables and scatter plots for continuous predictors have been shown below to understand the relationship between each of the predictors and transcoding time characteristics.

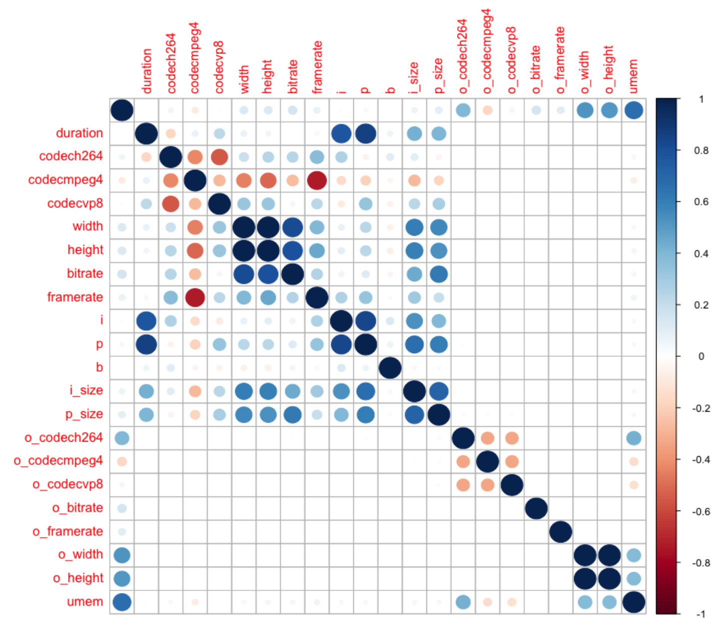
Based on the box plots, one can observe differences in transcoding time and o_codec variable with h264 consuming more time than the other formats. Because of the skewed distributions of outcome variable (transcoding time), one can observe significant number of data points lying outside the largest quartile.



For continuous predictors, the relationship with outcome variables are mostly linear. Specifically, umem shows a strong positive correlation with transcoding time. Like the observation with histogram of predictors, one can see discrete characteristics in certain cases such as height, o_height etc.

To understand relationship between outcome and predictors, correlation matrix plot has been generated with strong colored bubbles representing stronger correlation. Few observations are noteworthy. Strong correlation has been seen between height and width as well as between o_height and o_width. As mentioned before in background section, these occur because the resolutions for videos are standardized and defining one results in a specific value of the other. Hence, width and o_width variable has been removed from any further analysis.

Additionally, based on the earlier explained relationship between total number of frames and number of I,p and b frames, the total number of frames is a linearly dependent variables and has been removed from the analysis. Similarly, size of the video was also removed. Based on exploration, the dataset provided did not contain any non-zero values in the predictor size_b (size of b frames) and hence, the column was dropped from the study.



One can observe a strong positive relationship between transcoding time (output) and umem, o_height variable. Relatively, weaker positive correlation is seen between outcome and o_codech264 while mpeg4 codec in the output shows weak negative correlation with transcoding time.

III. Model Generation and Analysis

With a continuous outcome variable and predictors suggesting acceptable linear relationship with the outcome variable based on exploratory analysis, linear regression model has been selected as the model for this study. For constructing models, data has been divided intro train, validation and test data. The model generation has been done only on the training and validation data. Test data will be used as unseen data for evaluating the performance of final selected model.

A base MLR model was built and all the predictors and response variable were included in the MLR without any transformation. Following figure summarizes the model obtained.

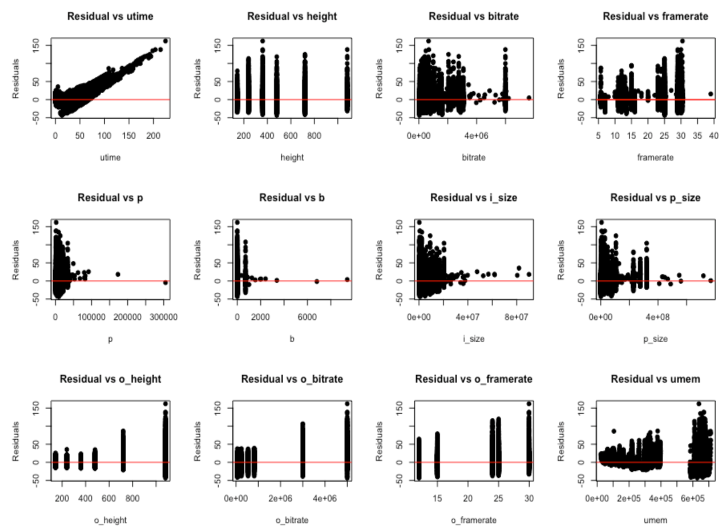
```
Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept) -3.106e+01  3.077e-01 -100.928  < 2e-16 ***
duration     1.343e-03  4.059e-04    3.308 0.000939 ***
codech264    1.762e-01  1.525e-01    1.156 0.247867    
codecmpeg4  -1.148e-02  1.794e-01   -0.064 0.948968    
codecvp8    -1.053e+00  1.862e-01   -5.654 1.58e-08 ***
height       9.212e-04  4.189e-04    2.199 0.027882 *  
bitrate      1.811e-06  8.369e-08   21.639  < 2e-16 ***
framerate    6.697e-02  1.050e-02    6.379 1.80e-10 ***
i           -1.343e-02  1.332e-03  -10.086  < 2e-16 ***
p            1.535e-04  2.754e-05    5.575 2.48e-08 ***
b            1.425e-03  4.205e-04    3.388 0.000704 ***
i_size      -1.537e-07  1.803e-08   -8.528  < 2e-16 ***
p_size       5.635e-09  1.608e-09    3.504 0.000458 ***
o_codech264  1.201e+01  1.174e-01  102.295  < 2e-16 ***
o_codecmpeg4 2.777e+00  1.079e-01   25.728  < 2e-16 ***
o_codecvp8   8.646e+00  1.079e-01   80.127  < 2e-16 ***
o_bitrate    1.425e-06  2.184e-08   65.264  < 2e-16 ***
o_framerate  2.482e-01  5.726e-03   43.355  < 2e-16 ***
o_height     1.810e-02  1.333e-04  135.809  < 2e-16 ***
umem         7.053e-05  4.785e-07  147.418  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.494 on 61885 degrees of freedom
Multiple R-squared:  0.6518,	Adjusted R-squared:  0.6517 
F-statistic:  6098 on 19 and 61885 DF,  p-value: < 2.2e-16
```

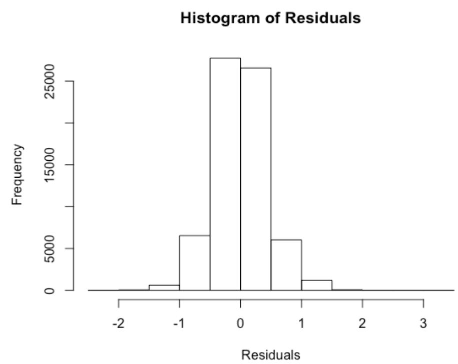
Most of the predictors, except codec mpeg4 and codech264 have very small p-values and therefore, are considered significant in the presence of other predictors at alpha =0.05. Results suggest that 65.18% of the variation in the response variable can be explained by the given predicting variables in the base form.

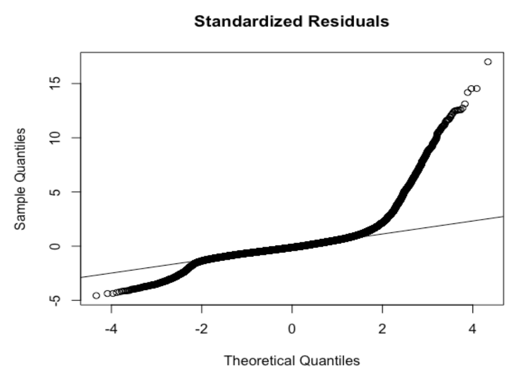
To evaluate goodness-of-fit tests for the model, residual plots have been generated against the outcome variables and various predictors. The plot highlights severe assumption violations:

- Strong trend is observed between residuals and predictors such as umem suggesting a variable transformation might be required.
- Non-constant variance can be seen in the plot of residuals and predictors such as o_height and o_bitrate

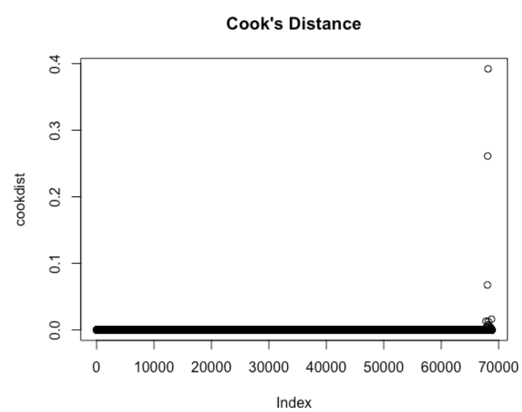


Lastly, normality is an important assumption in the context of Multiple Linear Regression. QQ plot and histogram of the residuals have been shown below. One can observe that very high deviations from normality has been observed and the video dataset has much thicker tails than expected normal distribution.





Lastly, to understand the presence of influential points, Cook's distance analysis was performed. One can observe 3 clear outlying points with cook's distance > 0.05. These points were removed from any further analysis.



Since there is no ordering impact in the data collection of transcoding experiments, plotting of residuals vs index will not be insightful.

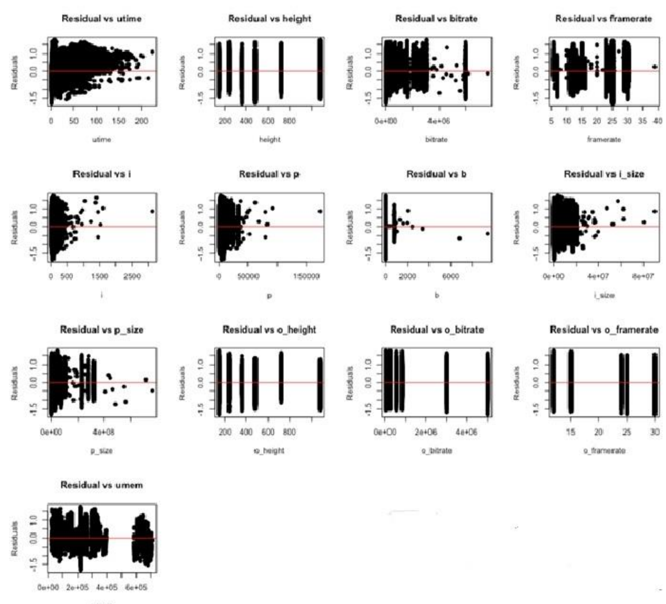
In summary, though the model was able to achieve R^2 score of 66%, it severely violated the underlying assumptions when evaluated for its goodness of fit. For improving the goodness of fit, different predictor and outcome variable transformations was experimented with. MLR with log transformation of the response variable seemed to cause the highest improvement in the fit and was selected for further analysis.

```
Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept)  -1.851e+00  1.371e-02 -135.046  < 2e-16 ***
duration      2.323e-04  1.910e-05   12.163  < 2e-16 ***
codech264     8.580e-02  6.761e-03   12.690  < 2e-16 ***
codecmpeg4   -1.830e-01  7.948e-03  -23.025  < 2e-16 ***
codecvp8     -9.268e-02  8.286e-03  -11.184  < 2e-16 ***
height        5.811e-04  1.855e-05   31.336  < 2e-16 ***
bitrate       1.604e-07  3.705e-09   43.302  < 2e-16 ***
framerate     1.097e-02  4.689e-04   23.383  < 2e-16 ***
i            -1.307e-03  6.034e-05  -21.667  < 2e-16 ***
p             1.259e-05  1.248e-06   10.092  < 2e-16 ***
b             1.322e-04  1.862e-05    7.101 1.25e-12 ***
i_size       -1.216e-08  8.011e-10  -15.181  < 2e-16 ***
p_size        7.327e-10  7.118e-11   10.294  < 2e-16 ***
o_codech264   1.879e+00  5.195e-03  361.679  < 2e-16 ***
o_codecmpeg4  8.585e-01  4.777e-03  179.735  < 2e-16 ***
o_codecvp8    1.372e+00  4.775e-03  287.405  < 2e-16 ***
o_bitrate     6.074e-08  9.664e-10   62.856  < 2e-16 ***
o_framerate   2.120e-02  2.534e-04   83.653  < 2e-16 ***
o_height      2.314e-03  5.899e-06  392.305  < 2e-16 ***
umem          4.454e-07  2.118e-08   21.033  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4201 on 61883 degrees of freedom
Multiple R-squared:  0.8752,	Adjusted R-squared:  0.8751
F-statistic: 2.283e+04 on 19 and 61883 DF,  p-value: < 2.2e-16
```

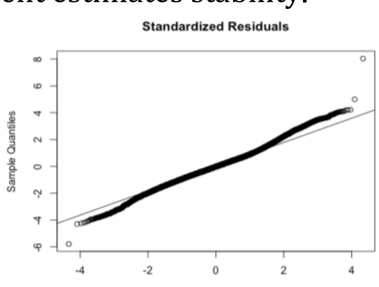
Transformation of outcome variable significantly improved model's R-squared value. 87.52% of the variation in the response variable can be explained by the given predicting variables as compared to 65% in the base form. P-values for all the coefficients are extremely small and hence, all the coefficients are considered statistically significant even at alpha level 0.001.

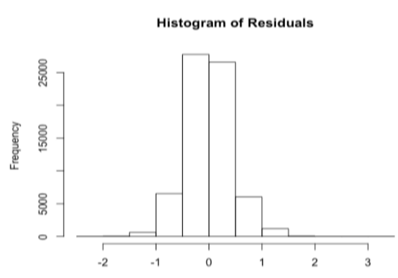
Residual plots have been generated for the new MLR to evaluate its goodness of fit. Significant improvement can be observed as compared to the base model. Residuals are randomly distribution around zero line and the band of residuals have constant width suggesting no deviation from linearity and constant variance assumption.

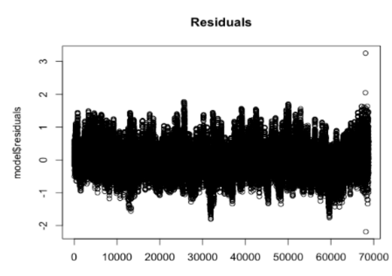


Residual histogram and QQ plot suggest much lesser deviation from normality as compared to the model with the transformation of outcome variable. One can still see some deviation as residuals have thicker tails as compared to a normal distribution.

Given the size of data (~70000), the deviation is still acceptable for the model generated. Stability experiments will be conducted with different dataset sizes later in the study to understand the coefficient estimates stability.





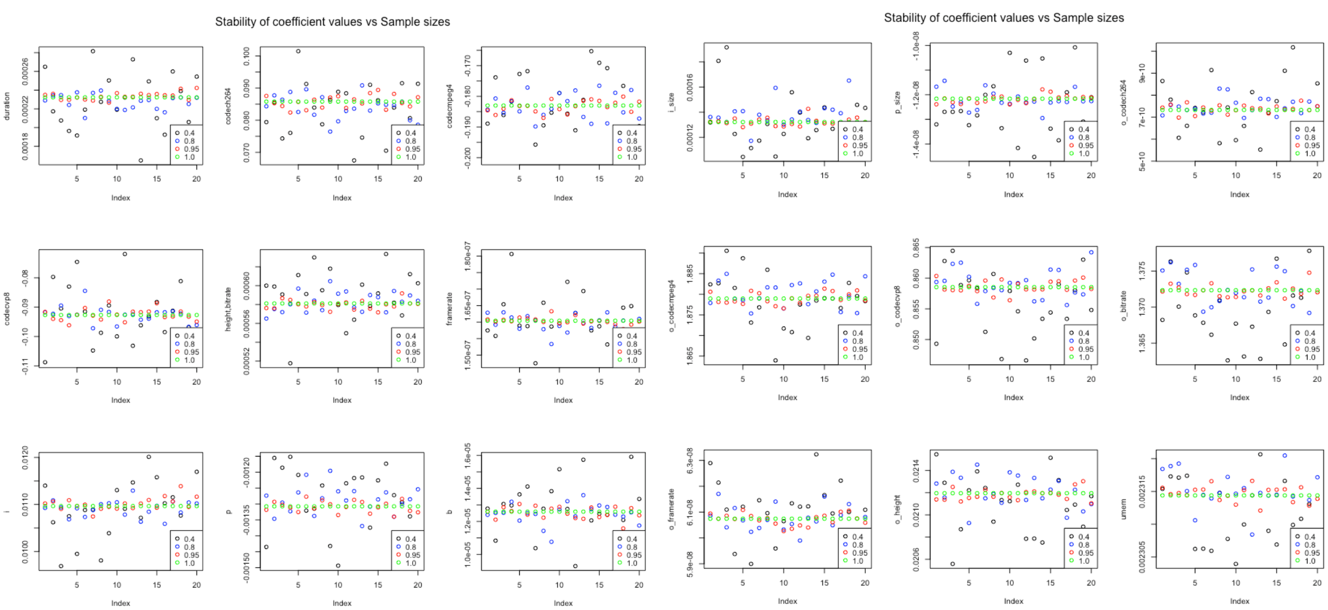


Lastly, the residuals independence can be confirmed using the following residual plot and no violating trends are observed based on the plot.

IV. Evaluating Model coefficients stability

To evaluate the variability of model coefficients corresponding to different predictors, repeated model generation was performed with different dataset sizes (40%, 80%, 95% and 100% of the training + validation data). With each data set size, multiple repeats (20) were carried out and the reduced dataset was randomly selected from the entire (training & validation) dataset.

For each of the predictor, a scatter plot was generated to illustrate the stabilizing variance observed with increasing training dataset size. With only 40% of training + validation dataset, the variance is relatively much higher than other cases for all the predictors. Model coefficient estimates are much stable when 80% or more of the (training + validation) dataset is used. This can be an indication that even though slight normality deviation was seen during goodness of fit evaluation, with the amount of data being used for training, the estimates are still reliable.



V. Variable Selection

With the model with necessary ready as benchmark, variable selection and regularized regression using techniques were implemented: Forward Stepwise Regression, Backward Stepwise Regression, Forward Backward Stepwise Regression, Lasso Regression, Ridge Regression and Elastic Net.

To compare various models' performance and select the best-performing one in terms of accuracy, multiple train-validation-test frameworks have been implemented.

- Original dataset is randomly split into 90% (Training & Validation) and 10% (Final Test).

- Training & Validation set is then split into 80% Training and 20% validation. Repeat the training & validation process for n times, report the mean and median accuracy metrics of models.
- Perform K-fold cross validation within Training & Validation Set, report the mean and median accuracy metrics of models

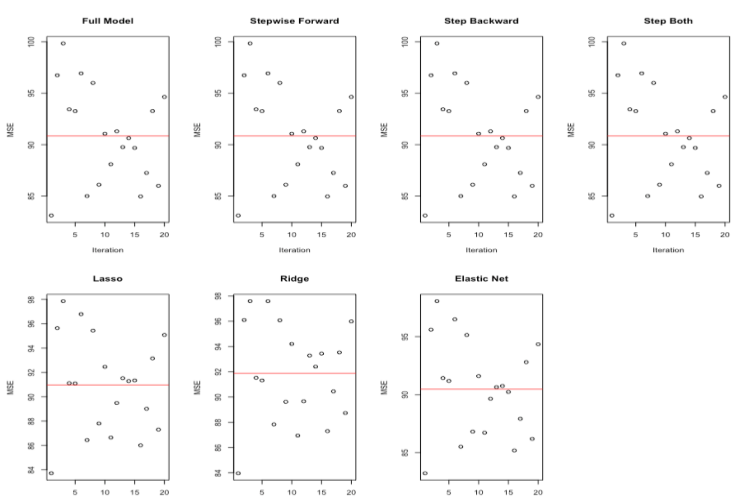
Mean Squared Error (MSE) has been used as the accuracy metrics in validation & model selection, as better performing model is expected to have lower Mean Squared Error (MSE). Mean Squared Error (MSE) is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$$

Randomized Training & Validation

Randomized training and validation using 90% of the original dataset was simulated for 20 iterations and MSE distribution was observed across models. The results show stepwise Regression models have very similar performance, indicating they all agree on the variables selected and have the same coefficients.

Lasso and Ridge Regressions perform slightly worse than the full model, explained by their penalization characteristics. Lasso implements L_1 penalty and could forces coefficient values to be zero. Ridge implements L_2 penalty and accounts for multicollinearity. As multicollinearity issue has been identified and mostly addressed earlier by removing highly correlated variables based on theoretical guidelines and correlation matrix plot, ridge regression does not show any performance edge. Elastic Net model shows the best accuracy as it simultaneously performs variable selection and coefficients regularizations. Based on this randomized training and validation, Elastic Net model has been selected.



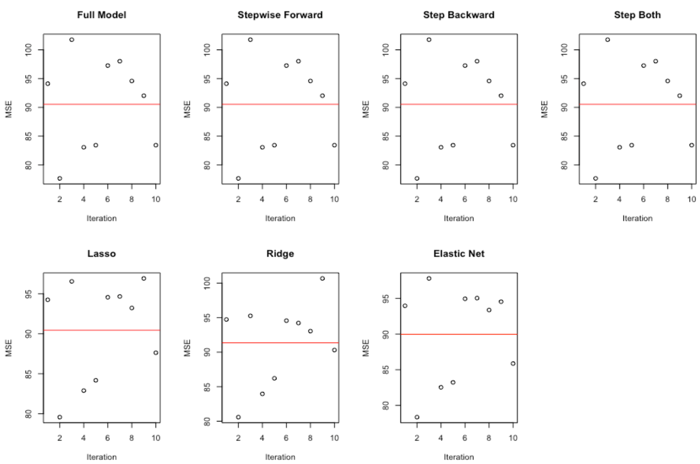

```
MSE Summary of Full Model After 20 Runs :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  83.10   86.96   90.85   90.85   93.74   99.85
MSE Summary of Stepwise Forward After 20 Runs :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  83.10   86.96   90.85   90.85   93.74   99.85
MSE Summary of Step Backward After 20 Runs :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  83.10   86.96   90.85   90.85   93.74   99.85
MSE Summary of Step Both After 20 Runs :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  83.10   86.96   90.85   90.85   93.74   99.85
MSE Summary of Lasso After 20 Runs :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  83.72   87.68   91.21   90.96   93.63   97.85
MSE Summary of Ridge After 20 Runs :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  83.96   89.40   91.96   91.88   94.65   97.59
MSE Summary of Elastic Net After 20 Runs :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  83.23   86.79   90.70   90.48   93.19   98.06
```

10-Fold Cross Validation

A second framework based on cross-validation was also implemented. 10-fold cross validation (k=10) was performed by combining training and validation dataset. The results are aligned with the randomized training and validation.

Full Model and Stepwise Regression models select the same set of variables with same coefficients values and therefore, have identical performance. This also illustrates that the full model (baseline model) already has decent performance so that stepwise greedy search is not able to further reduce the number of predictors. The result also aligns with the fact that all model coefficients are statistically significant at α=0.01. Again, Ridge does not perform well in terms of prediction accuracy and Lasso is on-par with the full model with slightly higher Median and lower mean.

Elastic Net still performs the best in 10-fold cross validation and will be selected and tested on the test dataset




```
MSE Summary of Full Model With 10 Fold Validation :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  77.65   83.43   93.07   90.53   96.59  101.76
MSE Summary of Stepwise Forward With 10 Fold Validation :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  77.65   83.43   93.07   90.53   96.59  101.76
MSE Summary of Step Backward With 10 Fold Validation :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  77.65   83.43   93.07   90.53   96.59  101.76
MSE Summary of Step Both With 10 Fold Validation :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  77.65   83.43   93.07   90.53   96.59  101.76
MSE Summary of Lasso With 10 Fold Validation :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  79.57   85.04   93.73   90.44   94.63   96.92
MSE Summary of Ridge With 10 Fold Validation :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  80.59   87.24   93.63   91.36   94.69  100.68
MSE Summary of Elastic Net With 10 Fold Validation :
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  78.36   83.89   93.66   89.96   94.84   97.80
```

3. Results

This section summarizes the performance of best performance model settings (Full model with the elastic net based model). It also highlights the impact of implementing elastic net on model parameter estimates and variable selection in the model.

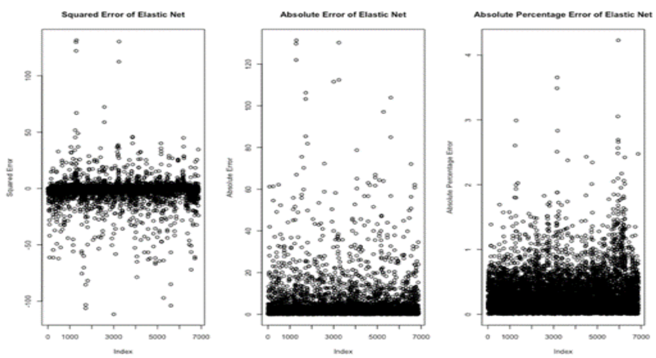
The final model has been trained and compared with other potential models using entire training and validation dataset (90% of the original dataset). This model is then tested and evaluated on the test dataset (10% of the original dataset) to assess its performance on an unseen dataset. The below table summarizes the prediction accuracy metrics of the Elastic Net:

	Elastic Net	**Full Model (Reference)**
Mean Squared Error (MSE)	88.64382	89.08701
Mean Absolute Error (MAE)	3.648751	3.621569
Mean Absolute Percentage Error (MAPE)	0.345424	0.3360915
Precision Measure (PM)	0.3593617	0.3611584

While the accuracy measures are defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y_i})^2 \ , \ MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \widehat{Y_i}\right| \ , MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \widehat{Y_i}|}{Y_i} \ , \quad PM = \frac{\sum_{i=1}^{n}|Y_i - \widehat{Y_i}|^2}{\sum_{i=1}^{n}(Y_i - \overline{Y_i})^2}$$

MSE of Elastic Net reported on the test set agrees with the MSE values in validation and cross validations. Hence, the model doesn't seem to overfit and the accuracy metrics reflects the performance of the Elastic Net model in prediction. On analyzing the accuracy metrics, one can observe MAE as 3.648 and MAPE as 0.35 indicating an average absolute deviation of 3.648s and an average percentage deviation of 34%. Such error characteristics could be explained by the fact that the model does not predict well only on the extreme values (observations with very high transcoding time). However, the model does reasonably good prediction in majority of the cases.



Benchmarking with the linear regression model reported the original research paper, our Elastic Net model has better accuracy (3.65 vs 7.23 in absolute error).

Final Variables selected using different methodologies have been summarized below. No difference in variables selected is observed. (■ : Selected)

	id	duration	codec	width	height	bitrate	framerate	i	p	o_width	o_height
Elastic Net	NA	■	■		■	■	■	■	■		■
Full Model	NA	■	■		■	■	■	■	■		■

	b	frames	i_size	p_size	b_size	size	o_codec	o_bitrate	o_framerate	umem
Elastic Net	■		■	■			■	■	■	■
Full Model	■		■	■			■	■	■	■

Following table compares the effect of elastic on parameter estimates. With elastic net, model coefficient estimates, which are less important, should tend to shrink. As can be seen in the table below, similar trend is observed. For example, the parameter estimate corresponding to i_size has shrunk from -1.2e-08 to -9.5e-09.

	Intercept	duration	codech264	codecmpeg4	codecvp8	height	bitrate	framerate	i
Elastic Net	-1.72e+00	1.66e-04	8.60e-02	-1.61e-01	-4.70e-02	5.58e-04	1.51e-07	1.06e-02	-8.18e-04
Full Model	-1.84e+00	1.98e-04	8.51e-02	-1.80e-01	-9.13e-02	5.79e-04	1.59e-07	1.08e-02	-1.27e-03

	p	b	i_size	p_size	o_codech264	o_codecmpeg4	o_codecvp8	o_bitrate
Elastic Net	6.59e-06	1.13e-04	-9.54e-09	8.75e-10	1.69e+00	7.32e-01	1.23e+00	5.77e-08
Full Model	1.33e-05	1.25e-04	-1.20e-08	7.36e-10	1.87e+00	8.55e-01	1.37e+00	5.99e-08

	o_height	umem	o_framerate
Elastic Net	2.20e-03	7.47e-07	2.03e-02
Full Model	2.31e-03	4.57e-07	2.11e-02

Final Model (Elastic Net):

$$\log(utime) = -1.725 + 0.000166 * duration + 0.0861 * codech264 - 0.161 * codecmpeg4 - 0.0470 * codecvp8 + 0.000558 * height + 0.000000151 * bitrate + 0.0107 * framerate - 0.000818 * i + 0.00000660 * p + 0.000113 * b - 0.00000000954 * i_{size} + 0.000000000876 * p_{size} + 1.697 * o_{codech264} + 0.732 * o_{codecmpeg4} + 1.233 * o_{codecvp8} + 0.0000000578 * o_{bitrate} + 0.0204 * o_{framerate} + 0.00220 * o_{height} + 0.000000747 * umem$$

From estimated MLR model, we can identify that several key attributes: umem, codec, input & output framerates and output height have strongest impacts in determining video transcoding time. Holding other predictors constant, 1 pixel increase in output height will result in 0.0022 increase in log(utime); For example, by keeping other input & output video parameters fixed and changing output height from 720px to 1080px, (720p to 1080p), the expected log transcoding time will increase by 0.792.

The findings align with the correlation matrix plot and general principles on video signal processing that video transcoding time primarily depends on input & output formats and the output size of the video.

4. Conclusions

The team studied and built multiple linear regression models on video transcoding dataset from UCI machine learning repository. Using R, detailed exploratory data analysis was performed by analyzing predictor distributions and removing multicollinearities and outliers. Model assumptions were validated, and different variable selection techniques were applied to identify the final model. Production accuracy was estimated using a new dataset and better accuracy was observed as compared to the original research paper MLR model (3.65 vs 7.23 in absolute error). The model well explains and predicts video transcoding time based on the input and output parameters of the video file. Umem, Input and output video codec and output video dimension were identified as key attributes and have the strongest impact on the transcoding time.

The analysis can be used by online video transcoding services or cloud storage companies as guideline to deploy and optimize their resources and time allocations. YouTube can also utilize this model to predict transcoding times as well. The model itself can be easily scaled and updated using different processor or computing platforms.

One can extend this research from video transcoding to video streaming as the industry is moving towards cloud computing and video consumption on-demand, along with incorporating newer generation of codecs like HEVC (H.265) that supports 4K and more storage-efficient compression.

Citations

1. Tewodors, Deneke, et al. “Analysis and Transcoding Time Predictions of Online Videos.”*Akademi University, Finland*, IEEE International Symposium on Multimedia, 2015, pp. 318–322.

2. Vetro, Anthony, et al. “Video Transcoding Architectures and Techniques.” *Signal Processessing Magazine*, Mar. 2003, pp. 18–29.

Appendix A.1

General video characteristics	
Id	Youtube video id
duration	Duration of video in seconds
i	Number of i frames in video
p	Number of p frames in video
b	Number of b frames in video
frames	Total number of frames
i_size	Size of i frames
p_size	Size of p frames
b_size	Size of b frames
size	Total size of all frame types
umem	Memory allocated for transcoding
Source characteristics	
codec	Codec format (such mpeg8, flv) for source
width	Width of video in pixels at input
height	Height of video in pixels at input
bitrate	Number of bits that are processed per unit time at source
framerate	Frequency (rate) at which frames are displayed in video at source
Target characteristics	
o_codec	Codec format (such mpeg8, flv) for output
o_bitrate	Number of bits that are processed per unit time at output
o_framerate	Frequency (rate) at which frames are displayed in video at target
o_width	Width of video in pixels at output
o_height	Height of video in pixels at output
Outcome variable	
utime	Total time consumed for transcoding process

Appendix A.2

Github Repo: https://github.com/mxu007/ISYE6414

mxu007 Update README.md — Latest commit 7ab71e5 25 minutes ago

graphs	add prediction results plot	3 hours ago
.Rhistory	add prediction results plot	3 hours ago
ISYE6414.R	add prediction results plot	3 hours ago
LICENSE	Initial commit	a day ago
README.md	Update README.md	25 minutes ago
transcoding_measurement.tsv	add data file	a day ago

README.md

ISYE6414 Regression Anlaysis Final Project

Final Group Project for ISYE6414 Regression Analysis Fall'17.

Project Team Members: Byron Kim, Minghan Xu and Shishir Suman

Original Dataset can be found:
https://archive.ics.uci.edu/ml/datasets/Online+Video+Characteristics+and+Transcoding+Time+Dataset