Assessment 2 –
Predicting YouTube video likes using publicly available data from the YouTube API

D. Rohan

Abstract

(5 marks)

Many influencers do not use their data to its fullest potential. In this report, an attempt was made to demonstrate the relevance and benefit that data science can have on the profession by predicting the amount of likes content will get based on available metadata sourced freely from an API. Five models were tested including: Linear regression, Bayesian-Ridge regression, Support Vector Machine regression, Random Forest regression, and a Neural Network regressor. Of these, it was observed that the Random Forest model performed the best (MSE = 0.00261, RMSE = 0.05105, R^2 = 0.89013), but the Linear regression model performed far quicker with minimal costs to accuracy. Data Science will undoubtably improve audience reach and content interaction, and its implementation is likely to become more prominent in the coming years.

Introduction (5 Marks)

In the world of social media, being able to predict the traction that content will get ahead of time can greatly improve an influencer's ability to sell themselves and market their brand. YouTube Channel data belonging to Chloe Hayden, Paige Layle, and 'The Aspie World' (TAW) was sourced to model the amount of likes a YouTube video would receive. Hayden is an Australian disability advocate, actress, author, and model. This information will provide her with intelligence relevant to her creative process and use of SEO to reach a specific target and optimize her success as an influencer. Like many creators and companies, she is not using the data available to its fullest potential. In this report, an attempt was made to demonstrate the relevance and benefit that data science can have on her profession.

The data was collated directly from the YouTube API, greatly reducing the number of missing values and inaccurate data. Layle and TAW are channels similar to Hayden's, and so their video metadata was used to extend the dataset. The data pipeline was established in a manner that will work with any channel id and can be updated in real time for future use by Hayden. The data was manipulated in a number of ways in search of meaningful variables; some were already available, and some engineered. Once missing values, outliers, and unappropriated instances had been removed, 822 instances remained, each with 30 predictors and the target variable ('times liked'). A selection of these had moderate to high correlation ($r > 0.25$), but many did not. Models were selected with these constraints in mind:

- **Linear regression (LR)** – makes four assumptions: target variable can be modelled linearly, errors are normally distributed, no heteroscedasticity, and observation independence. (Only variables above a correlation threshold of $r > 0.25$ were used).
- **Bayesian-Ridge regression (BRR)** – creates a linear model that uses probabilities in place of exact values and adds a penalty based on any multicollinearity.
- **Support-vector machine regression (SVR)**– creates a hyperplane from training data to map the target variable. Works well on small datasets and depends more on multidimensional position rather than Pearson correlation values.
- **Random forest regression (RFR)** – an average of decision trees built from instances with replacement which provides better real-world application and prevents overfitting. These models are not excellent at extrapolating values beyond those observed in the training set.
- **Neural Network regression (NN)**– finds more complex relationships in the data, not just the linear.

Materials and Methods (10 marks)

All the analytical work was produced using Python and several libraries through Jupyter Notebook (Ari & Ustazhanov 2014; Bird 2006; Harris et al. 2020; Kim & Wurster 2019; Kluyver et al. 2016; McKinney 2011; Pedregosa et al. 2011; Virtanen et al. 2020; Waskom 2021). The data was sourced directly from YouTube using the YouTube API (for which there is no available citation or documentation), meaning there were no missing or incorrect values unless creators made errors in the video uploading process. The data pipeline was engineered to remain relevant and enable the dataset to grow in real time (restrained only by the APIs daily 10,000 request capacity). From the metadata available, the following features were constructed:

- Days since upload
- Video duration
- No. tags in title
- Title length
- Description length
- No. of tags
- Average comment count per day
- Average views per day
- No. of emojis in title
- No. of emojis in description
- Description sentiment
- Weekday of uploaded
- Month of uploaded

These features all influence a viewer's decision to watch a video, and more viewers increases the likelihood of more likes. Some features predict whether or not a video can elicit an emotional connection (emoji's, tags in title, sentiment, etc.) that may result in an emotional response in the form of a like or subscribe (Gregurec & Grd 2012; Santamaría-Bonfil & López 2019). And many are indicators of past occurrences (the daily averages). When viewed holistically, these features are very telling of whether a video will receive likes; the complex relationships aren't necessarily detected through simple Pearson correlation coefficients with the target variable (See Notebook for individual reasoning behind each variable). From here, an Exploratory data analysis was undertaken. A few right skewed numerical variables were identified and transformed by $\log_{10}(x+1)$ to better distribute the data which consequently improved Pearson correlation with the target variable. All numerical variables except the sentiment score were then converted to z-scores (the sentiment scores were provided as scaled scores by the 'nltk' library). Z-scoring is a simple method of normalizing data. It communicated the data in terms of standard deviations from the mean, made outlier identification simpler, showed improved correlation with the target variable, and made scales comparable to prevent innate weighting issues that can occur when variable ranges differ (Patro & Sahu 2015). Some leniency was allowed in terms of outliers to maintain a good number of instances, removing only those with a z-score of three or more; in hindsight the problematic variables perhaps could have been excluded all together. It is also worth mentioning that in a real-world context, there aren't necessarily any outliers present as the true range is much wider than what is represented in the dataset. An ANOVA found there was no significance in the month uploaded (likely because holiday periods are not uniform globally), but there was significance in the day uploaded (p-values of 0.33 and $8\text{x}10^{-27}$ respectively). Both were kept as interaction terms may be informative in the more complex models. A principal component analysis (with whitening to reduce multicollinearity and give features the same variance) on the predictors found that 95% of the variance could be explained with just 20 components, which demonstrated the potential for dimensionality reduction.

For the sake of reproducibility, all random states were set to 42. This is relevant in the data splitting process where 5 folds were randomly generated, placing 80% of the data in the training set and 20% in the testing set. The accuracy of each model was then assessed through cross validation within the training set and by measuring the mean squared error (MSE), root mean squared error (RMSE), and r-squared (R^2) values of the performance on the training set and on the remaining 1/5th of the data. Good accuracy in the training predictions and poor accuracy in the testing predictions would flag over fitting issues and overall model accuracy and relevance. These techniques proved useful in preventing overfitting. The accuracy markers were useful in identifying improvement between each untuned model (with arbitrary hyper parameters) and their tuned model.

Results

(10 marks)

All models showed some degree of improvement after tunning, the largest improvement was observed in the neural network model which initially had an R^2 value of 0.1. The less complex the model, the lower the improvement seemed to be. This is a consequence of there being fewer hyper parameters in those models. In all cases except RF, tuning hyperparameters improved the accuracy. However, for RF it was a matter of reducing overfitting. Ideally this would be done by adding more data, but improvements can be made through pruning. In any case, the RF model is not well suited for long term use because it cannot extrapolate beyond the ranges provided to it during training. As subscribership and likes will hopefully increase, the RF could not accurately interpret the most current data which presumably has more viewers and likes. However, it can still be used to assess variable importance to assist in future analysis.

Table 1: The tuned model metrics compare to the naive baseline; the mean of the target variables in the training set. If PCA was observed to improve the model, the number of components used is outlined.

Model	MSE	RMSE	R^2	PCA Components
Naïve Baseline	0.02372	0.15403	-0.00029	N/A
LR	0.00385	0.06208	0.83750	10
BRR	0.00362	0.06017	0.84736	-
SVR	0.00365	0.06038	0.84629	-
RF	0.00261	0.05105	0.89013	-
NN	0.00364	0.06029	0.84673	29

Discussion

The number of likes a video will receive is never certain and the same video could perform entirely differently if it were uploaded a week or even a day later; as the ANOVA clearly indicated. There will always be a degree of uncertainty associated with these predictions due to the concepts of chaos theory. Considering these factors and the fact that these channels are still growing and the subscribership is relatively low, the models have performed quite well. On a small dataset full of context dependent factors and complex relationships, it would be unreasonable to expect much better.

The R^2 values can be interpreted as the percentage of variance that can be explained by the model; all the models can therefore explain more than 80% of the variance. The RMSE can be interpreted as the standard deviation of unexplained variance; so, the standard deviation of unexplained error in all the models was less than 0.07. Even with the overfitting issues identified in the RF model, it was still the best performing model based on these metrics. However, the RF model would be ineffective at extrapolating beyond the range in the dataset and is computationally more expensive than other models. For a creator's purposes, the trade-off between accuracy and time may make the LR model more attractive. As the data set grows the computational costs of the other methods will also continue to rise. In time, other models may become more advantageous, but currently creators don't need a precise number, just an indicator. That is why the LR model has been saved for future use. With better outlier treatment, and more nuanced/imaginative predictive features, the accuracy may even be enhanced.

Conclusion (5 marks)

A model was created that enables YouTube creators to make mock predictions to determine how many likes a video will get. Five models were trained, and their accuracy was tested with MSE, RMSE, and R^2 performance metrics. The RF was most accurate, but it was decided that the time costs associated with this model made the LR model more attractive. Time constraints have resulted in limited optimization of hyperparameters, however with the hardware available this was unavoidable. In terms of data usage, perhaps some tuning could have been done better regarding the number of k-folds used to split the data. Some features that might benefit future attempts include:

- An indicator of creator gender, as content regarding such personal matters may receive more viewers if it is a female's voice.
- An indicator describing rhythm associated with emotional response (gentle voice, excited voice, music, background noise, etc.)
- A feature that generalises tags into category (making the most of tag data without the computational cost of one hot-encoding every individual value).

The success of this analysis showcases the benefits that data science can have in this space, as well as its applications in marketing, brand management, and forecasting.

References

(5 marks)

Ari, N & Ustazhanov, M 2014, 'Matplotlib in python', in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO),* IEEE.

Bird, S 2006, 'NLTK: the natural language toolkit', in *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions,* pp. 69-72.

Gregurec, I & Grd, P 2012, 'Search Engine Optimization (SEO): Website analysis of selected faculties in Croatia', in *Proceedings of Central European Conference on Information and Intelligent Systems,* pp. 211-218.

Harris, CR, Millman, KJ, Van Der Walt, SJ, Gommers, R, Virtanen, P, Cournapeau, D, Wieser, E, Taylor, J, Berg, S, Smith, NJ, Kern, R, Picus, M, Hoyer, S, Van Kerkwijk, MH, Brett, M, Haldane, A, Del Río, JF, Wiebe, M, Peterson, P, Gérard-Marchant, P, Sheppard, K, Reddy, T, Weckesser, W, Abbasi, H, Gohlke, C & Oliphant, TE 2020, 'Array programming with NumPy', *Nature*, vol. 585, no. 7825, 2020-09-17, pp. 357-362.

Kim, T & Wurster, K 2019, 'Emoji for python'.

Kluyver, T, Ragan-Kelley, B, Pérez, F, Granger, BE, Bussonnier, M, Frederic, J, Kelley, K, Hamrick, JB, Grout, J & Corlay, S 2016, *Jupyter Notebooks-a publishing format for reproducible computational workflows*, vol. 2016.

McKinney, W 2011, 'pandas: a foundational Python library for data analysis and statistics', *Python for high performance and scientific computing*, vol. 14, no. 9, pp. 1-9.

Patro, S & Sahu, KK 2015, 'Normalization: A preprocessing stage', *arXiv preprint arXiv:1503.06462*.

Pedregosa, F, Varoquaux, G, Gramfort, A, Michel, V, Thirion, B, Grisel, O, Blondel, M, Prettenhofer, P, Weiss, R & Dubourg, V 2011, 'Scikit-learn: Machine learning in Python', *The Journal of Machine Learning Research*, vol. 12, pp. 2825-2830.

Santamaría-Bonfil, G & López, OGT 2019, 'Emoji as a proxy of emotional communication', in *Becoming Human with Humanoid-From Physical Interaction to Social Intelligence*, IntechOpen.

Virtanen, P, Gommers, R, Oliphant, TE, Haberland, M, Reddy, T, Cournapeau, D, Burovski, E, Peterson, P, Weckesser, W & Bright, J 2020, 'SciPy 1.0: fundamental algorithms for scientific computing in Python', *Nature methods*, vol. 17, no. 3, pp. 261-272.

Waskom, ML 2021, 'Seaborn: statistical data visualization', *Journal of Open Source Software*, vol. 6, no. 60, p. 3021.