Optimum Model For Video Memorability Prediction

Maureen Maguire
1921399

Maureen.maguire47@mail.dcu.ie

ABSTRACT

The aim of this study is to predict media memorability. This task is part of the MediaEval 2020 Benchmarking Initiative for Multimedia Evaluation. The purpose of this study is to create a model which efficiently predicts how memorable videos are. The performance of the models is based upon the Spearman Rank Correlation Coefficient score of the predictions and ground truth. This paper details multiple approaches using machine learning algorithms to complete this task and compares the performance of each approach in order to find the optimum one.

1 INTRODUCTION AND RELATED WORK

With the exponential growth of E-commerce in recent years, businesses are turning to social media platforms like Tik Tok and Instagram for marketing. They are required to be more tactical with the content of their advertisements because adverts on these platforms are very time restricted. Businesses only have a few seconds to make an impact and so they must ensure their ads maximize memorability and stand out of the crowd. But what makes a video memorable ? This paper looks at how we can predict the memorability of videos based on their features. The dataset used in this study was provided by MediaEval. It consists of 6000 videos, their features and their memorability score. This score is divided into short term and long-term scores. There are a broad range of features related to the videos. The features used in this study are the video captions, C3D spatial-temporal visual features and the histogram of motion patterns(HMP). The captions are the video titles and, in most cases, provide an informative description of the video contents. The data provided was 6000 instances which was split into 4800 for model development and 1200 for model validation. The goal is to implement several individual regression models to efficiently predict the numerical short term and long-term scores. Then employ ensemble techniques, combining these models to create a more powerful model. The models implemented are Linear Regression, Ridge Regression, Decision Tree, Support Vector Machine and ensemble methods including Random Forest, Voting, Bagging and Stacking.

2 KEYWORDS

Support Vector Machine, Ridge Regression, Decision Tree Regression, Linear Regression, Random Forest, Ensemble, Voting, Bagging, Stacking

3 MY APPROACH

The goal of this study is to produce a model which can predict long and short-term memorability scores as efficiently as possible. While training and validating models, the efficiency of the predictions is based on the Spearman's rank correlation coefficient score with the true labels. Several different approaches were implemented and compared based on this score. The approaches are Ridge Regression, Decision Tree, Support Vector Machine(Regression) and finally, Ensemble with Random Forest, Voting, Bagging and Stacking. The goal is to determine the optimum model for memorability prediction. Each approach is trained on several pre-computed individual video features each with unique transformations applied. An approach which doesn't seem to have been experimented on previously is to combine the different transformations of captions and use them together as the feature matrix. In this study, I combined the TF-IDF, One Hot encoded and Sequences together and applied them to each mode.

4 PRE-PROCESSING

In order to prepare the dataset for the models some pre-processing techniques were carried out. This included transforming the feature vectors into a representation that can be used more easily by estimators further on in the study. One of the main features used for prediction in this study were the video captions. In previous studies on video memorability, captions have been found to be the best features to use for memorability prediction (Azcona, 2019). Standard text cleaning was applied to the captions which included converting text to lowercase, replacing punctuation with spaces and removing stop words. In the study (Uysal, 2014), this was found to greatly improve model's accuracy. When working with Machine Learning algorithms it is advised to work with numbers as opposed to text. Therefore, the captions text was encoded into numerical values. The following pre-processing techniques were then applied to the captions; One Hot Encoding, Sequencing and TF-IDF (Scikit-Learn, 2019). TF-IDF was used in a previous study and was found to be very successful (Azcona, 2019). A test was carried out using Linear Regression and the R-squared score of predictions and true values to confirm the data in this study was non-linear. The features that returned a negative R squared score could be considered non-linear (Mahapatra, 2019). This was implemented for each model and all data was found to be non-linear. The non-linear data was normalised, rather than standardised. The transformations of the text greatly increased the numbers of features, and so Principal Component Analysis (PCA) was implemented. It was important to ensure that very little information was lost when reducing dimensionality because information loss could result in poorer performance. PCA allows the preservation of 95% variance, thus minimising information loss. This is a common requirement for many machine learning algorithms (Scikit-Learn, 2019). Finally, all captions were combined with the C3D and HMP features and PCA was subsequently applied with 95% variance retained. This has been found to improve results (Cohendet, 2018).

4. METHODOLOGIES

Numerous models were implemented. Each model was tested on all transformed features mentioned above. 2-fold cross validation was also applied when there was a risk the model was overfitting. For the better performing models, I used Grid Search to find the optimum hyperparameter values. I defined my own scorer for Grid Search, the scorer returned the Spearman coefficient. This ensured Grid Search was validating the values in terms of Spearman score. Once each model was optimised, they were used in three ensemble methods in the hope of combining their strengths and creating an even more powerful model.

4.1 Decision Trees Regression

The target variables in this study are the float values of short and long-term memorability. As the target vectors are numerical it is appropriate to utilise Decision Tree Regression. It is recommended to apply dimensionality reduction before training a decision tree, so PCA was applied. This ensures that the tree can find discriminative features. The DecisionTreeRegressor from sklearn.tree uses the classification and Regression Tree (CART) algorithm and allows us to easily control a lot of hyperparameters of the model to ensure that the model is not overfitting or underfitting. For example, restricting the tree depth can ensure overfitting does not occur.

4.2 Ridge Regression

Ridge regression has been seen to give good results for memorability prediction (Azcona, 2019). I wanted to build on this by using Ridge regression with different input features. I wanted to avoid creating a model with too much flexibility. Using regularisation, the model can fit the data while keeping the models weights small. It is advised to always have some sort of regularisation present (Géron, 2019). The higher alpha is set here the more regularised the model. Different values of alpha were tested. Though this is a linear model, it was implemented so its strengths could be utilised later in the ensemble approach.

4.2 Support Vector Regressor

This study builds on the work carried out for the competition in preceding years in which SVR was implemented (Azcona, 2019). In the study (Phillip Isola, 2014), SVR was implemented to predict memorability. I chose SVM because it is very versatile and can support nonlinear regression. The Scikit-Learn SVR class was used. As the data we are dealing with is non-linear, kernelized SVM models were used. Different kernels, regularisation and epsilon margin values were tested through grid search. It is advised to scale the training data for SVM before applying it to the model as SVR class assumes this has been done. If this is not done, then the results of the model can be affected. This was carried out as part of a pipeline in the pre-processing phase.

4.3 Ensemble Methods

Ensemble methods are very commonly used in competitions, like the Netflix Prize (Netflix, 2009). Last year, DCU's MediaEval task submission found ensembling models performed well (Azcona, 2019). The ensemble methods used are Random Forest, Majority Voting, Bagging and finally Stacking. Up to this point some good predictor models have been created. The goal of using ensembles is to produce an even more powerful model by combining the strengths of the weak learners. It is common practise to incorporate ensemble techniques at the end of the study and it would be expected that the ensemble model outperforms other models, benefiting from "*wisdom of the crowd*" (Géron, 2019). For the voting regressor, multiple tests were carried out to determine which combination of models would perform best as base estimators. The two models that chosen were the Ridge regression and SVR models. I then used the voting regressor to fit each of the bases on the data. Three models were trained, each on the top three performing features which were TF-IDF, One Hot Encoded data and the combination of Captions, C3D and HMP. I then implemented the stacking regressor combining the Ridge, SVR and voting model. Finally, as a last test, I implemented the bagging method, combining this with the stacking method.

5 RESULTS

As discussed, multiple approaches were implemented in order to predict the memorability scores of videos and to determine which approach is optimum. As you can see in Figure 1, several models achieved high Spearman scores. Overall the short-term memorability scores were much higher than the long-term scores. Captions were the leading feature matrix for predictions, specifically the TF-IDF and One Hot Encoding transformations. Out of all feature's, sequences performed the worst with a highest score of only 0.232 for short-term and 0.069 for long-term. The most notable models' performance was that of the Stacking ensemble method for short-term predictions achieving 0.447 and, the voting regressor for long-term predictions with a score of 0.201.

Individual Models

The poorest performer out of this group was Decision Tree in both short- and long-term scores. Its highest scores only reached 0.293 for short-term and 0.107 for long-term. The second-best individual model was SVR with best scores of 0.436 and 0.180 respectively. Bootstrapping was tested in place of randomly sampling the data on SVR using Bagging Regressor from sklearn. However, this negatively affected the spearman score. Linear, RBF and Polynomial kernels were tested with Grid search for SVM. It was found that Polynomial kernel performed better with TF-IDF and RBF with other features like HMP and C3D. The best individual model was found to be Ridge regression with TF-IDF as the feature matrix, scoring 0.446 Spearman score. Ridge regression is a linear model and therefore it was expected that TF-IDF would perform well here. TF-IDF outperformed all other features for both short- and long-term memorability on this model. Decreasing the alpha negatively impacted performance. When the alpha is set low enough the model essentially becomes linear regression as there is no regularisation. It was found that alpha set to 10, making it a regularised model, greatly improved the Spearman score. The combination of the text transformations did not perform as well as was hoped, mostly scoring less than the individual text transformations.

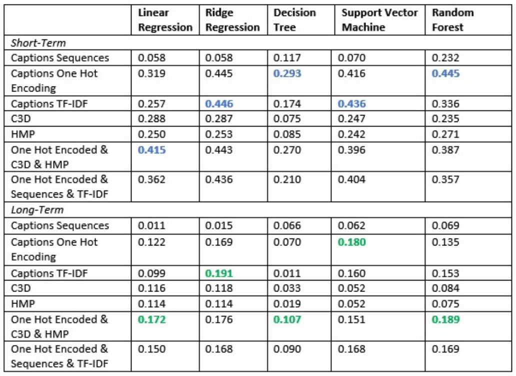
Ensemble Models

The results of the ensemble methods can be found in Figure 2 below. The ensemble method which performed the best for short-term predictions was the stacking regressor. However, this improvement was very slight, achieving only 0.002 higher than the Ridge model mentioned above. I performed 10-fold cross validation on the best model, the stacking model. This gave me an idea as to the actual performance of the model. Once again, used Random search to zone in on a range for the hyperparameters. This assisted in optimizing the model even further. The voting regressor managed to perform better than the best regressor in the ensemble, which was the Ridge model. SVR and Ridge estimators were used with the Voting Regressor and achieved a score of 0.455 which was on the highest scorer. The best performing ensemble method for long-term was the voting regressor using SVR and Ridge as base estimators. This model attained 0.201 Spearman score. As expected, when using an ensemble, we get slightly better predictions than using individual models.

6 CONCLUSION AND FUTURE WORK

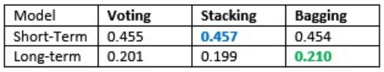
In this study, we achieved the task of predicting the memorability scores of videos. It was found that a stacking ensemble model is optimum when predicting the short-term memorability of videos. Moreover, it was found that a bagging ensemble method was found to perform best for predicting long-term memorability of videos. The final scores for short-term was 0.457 and long-term was 0.201. Both models achieve high results when the feature matrix is the TF-IDF transformation of the video captions. If the text data could be explored further, perhaps classifying the groups of words that provide the most information for prediction assigning these more weights for predictions. For, example captions containing references to people through words like "his", "her", "man", "girl". It has been proven that image memorability increases with the presence of people in the image (Isola, 2011). Perhaps this could be applied to videos too.

Figure 1: Individual Models Spearman Correlation



	Linear Regression	**Ridge Regression**	**Decision Tree**	**Support Vector Machine**	**Random Forest**
Short-Term					
Captions Sequences	0.058	0.058	0.117	0.070	0.232
Captions One Hot Encoding	0.319	0.445	**0.293**	0.416	**0.445**
Captions TF-IDF	0.257	**0.446**	0.174	**0.436**	0.336
C3D	0.288	0.287	0.075	0.247	0.235
HMP	0.250	0.253	0.085	0.242	0.271
One Hot Encoded & C3D & HMP	**0.415**	0.443	0.270	0.396	0.387
One Hot Encoded & Sequences & TF-IDF	0.362	0.436	0.210	0.404	0.357
Long-Term					
Captions Sequences	0.011	0.015	0.066	0.062	0.069
Captions One Hot Encoding	0.122	0.169	0.070	**0.180**	0.135
Captions TF-IDF	0.099	**0.191**	0.011	0.160	0.153
C3D	0.116	0.118	0.033	0.052	0.084
HMP	0.114	0.114	0.019	0.052	0.075
One Hot Encoded & C3D & HMP	**0.172**	0.176	**0.107**	0.151	**0.189**
One Hot Encoded & Sequences & TF-IDF	0.150	0.168	0.090	0.168	0.169

Figure 2: Ensemble Methods Spearman Correlation



Model	**Voting**	**Stacking**	**Bagging**
Short-Term	0.455	**0.457**	0.454
Long-term	0.201	0.199	**0.210**

Bibliography

Azcona, D. M. E. H. F. W. T. a. S. A., 2019. Predicting media memorability using ensemble models.. *CEUR Workshop Proceedings.*

Cohendet, R. Y. K. D. N. a. D. C., 2018. Annotating, understanding, and predicting long-term video memorability.. *In Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval,* pp. (pp. 178-186)..

Géron, A., 2019. *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow.* Third Edition ed. s.l.:O'REILLY.

Isola, P. P. D. T. A. a. O. A., 2011. Understanding the intrinsic memorability of images. *Advances in neural information processing systems.*

Mahapatra, A., 2019. *How to tell if a dataset is linear or not?.* [Online]
Available at: https://medium.com/@abhinav.mahapatra10/ml-basics-regression-how-to-tell-if-a-dataset-is-linear-or-not-594a4f1e8aaf
[Accessed April 2020].

Netflix, 2009. *Netflix Prize.* [Online]
Available at: https://www.netflixprize.com/leaderboard.html
[Accessed 2020].

Phillip Isola, J. X. D. P. A. T. a. A. O., 2014. What Makes a Photograph Memorable?. *IEEE Trans. Pattern Anal. Mach.Intell,* p. 1469–1482.

Scikit-Learn, 2019. *Preprocessing.* [Online]
Available at: https://scikit-learn.org/stable/modules/preprocessing.html
[Accessed 2020].

SciKit-Learn, 2019. *VotingRegressor.* [Online]
Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingRegressor.html
[Accessed April 2020].

Scikit-Learn, 2019. *Working With Text Data.* [Online]
Available at: https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
[Accessed April 2020].

Uysal, A. a. G. S., 2014. The impact of preprocessing on text classification. I. *Information Processing & Management,* pp. pp.104-112..