ECE225 Report: Popularity Prediction on YouTube Videos

Xinyi Liang
xil242@ucsd.edu
UC San Diego
USA

Nan Xiao
naxiao@ucsd.edu
UC San Diego
USA

Abstract

This study explores predicting user preferences for trending YouTube videos characterized by net likes (the difference between likes and dislikes). We embarked on a comprehensive analysis involving exploratory data analysis, feature engineering, and the application of various predictive models. Our feature set spans fundamental video features including video titles, comments enabled or disabled, and descriptions, as well as those with rigorous pre-processing on text-based data, including word-level and sentence-level sentiment analysis using TF-IDF. To solve the video popularity regression problem, we employed linear regression, Random Forest, Neural Networks, and Primary Component Analysis, each evaluated for the Mean Squared Error (MSE) in predicting the net likes. The analysis reveals the nuances of user preferences as well as the significant factors that make a YouTube video trending. The findings underscore the efficacy of diverse modeling approaches in understanding user preferences, providing valuable insights for enhancing recommendation systems and user experience in online culinary domains. The codes are currently open-source: Video Popularity Prediction Codes

Keywords: Predictive Modeling, Data Analysis, Linear Regression, Random Forest, Sentiment Analysis, Multi-Layer Perception

1 Introduction

YouTube videos are becoming increasingly popular for people busy at work seeking entertainment. Generating those with high popularity helps content creators earn much for their livings. Because of this, We aim to develop a machine learning model to predict the popularity of YouTube videos, measured by the "net likes" (difference between likes and dislikes). We will incorporate word-level and sentence-level sentiment analysis on the video title/channel title to explore the impact of the texts' emotional tone on viewer engagement.

2 Literature Review

We reviewed the prevailing works on preference analysis as well as preference-based recipe recommendations:

In terms of preference analysis, Zhang et. al [6]. incorporated the feature Collaborative Filtering (CF) and travel intents analysis into the analysis of user preference. Such methods, though restricted to domain-specific analysis, inspired us to apply feature filterings on YouTube videos to represent the consumed videos' features. Chin-Hui et. al. [2] proposed Aspect-Based Rating Prediction methods like ARPM/ARPM-Social integrating aspect detection and sentimental analysis information. Zhao et. al. [7] applied feature-specific metrics, including the preference topical-region preference and category-aware topical-aspect preference, for supervising a unified probabilistic model. Those methods provide a powerful paradigm for mining more significant features from the existing ones to build a model. However, the complexity of predicting different types of user preferences can make relying only on a single model inaccurate. Simplifying user preferences into an easier-to-learn one is necessary.

In terms of preference-based recipe recommendations, one other research that Food.com dataset is *Generating Personalized Recipes from Historical User Preferences* [3]. They combine two important tasks from natural language processing and recommendation systems: data-to-text generation [1] and personalized recommendation [4]. Personalized recipe generation involves expanding upon a recipe name and incomplete ingredient information to produce comprehensive, natural-text instructions aligned with the user's historical preferences. Their approach incorporates both technique- and recipe-level representations derived from the user's previously consumed recipes. They employ an attention fusion layer to merge these 'user-aware' representations, guiding the generation of recipe text. From their research report, we identified the significance of ingredients and historical preferences for users. Moreover, an attention fusion layer can be employed to comprehend some text related to users and recipes.

Instead of investigating general recipes, our works shed light on understanding user preferences. We aim at inferring their degree of liking towards ratings using the dominant features of the YouTube videos they consumed.

3 Exploratory Data Analysis

In this project, we focus on US trending YouTube videos data in Trending YouTube Video Statistics. The dataset provides a comprehensive overview of trending YouTube videos, capturing key metrics that reflect their popularity and audience

engagement, including the video title, channel title, category id, trending data, publish time, tags, views, likes and dislikes, description, and comment count. To measure the popularity of YouTube videos, we introduce **"net likes"** as our prediction target, which is difference between likes and dislikes. To explore gain insights and a deeper understanding of the data at hand, we do exploratory data analysis as the following. By thoroughly understanding the data, its characteristics, and its intricacies, EDA ensures that the predictive models built are well-informed, robust, and reliable.

3.1 Missing Values

Only description has 570 missing values. However, when we dig inside every feature, we also find some tags is labeled as "[none]" which is also missing values. We will deal with these features then.

3.2 Feature Engineering

3.2.1 Net Likes. The net likes is the difference between likes and dislikes. This measures the popularity of the video. This project, we focus on predicting this value based on video's information.

3.2.2 Time. Here, we transform "trending data" and "publish time" to the integer. We use their difference ("trending time") to record the length of popularity.

3.2.3 Binary Features. "comments disabled", "ratings disabled", "video error or removed" all record whether the video allows comments and ratings and whether the video goes wrong. The original datatype is bool, so we transform them into 0 or 1 for better regression.

3.2.4 Title and Channel Title. We calculate the tile and channel title's length to explore the impact of titles on the popularity.

3.2.5 Description. As is mentioned in 2.1, we know that description has many missing values. Therefore, we take the length of description into account instead of text. The figure confirms the fact that videos with no description tend to get low popularity (i.e. net likes). When we calculate the length of description, the NA one would be 0.

3.2.6 Tags. For this feature, the tag is in the formula "tag1 | tag2 | .. | tag n" or "[none]". First, we extract each tag from the string into a list. Second, we calculate the number of tags as "tags length". For "[none]" tag, the tag number will be 0. The figure confirms the fact that videos with no tags tend to get low popularity (i.e. net likes).

3.2.7 Category ID. Here, we would explore the relationship between net likes and video's category class. We can see that the videos in category 10 and 29 tend to have high net likes.

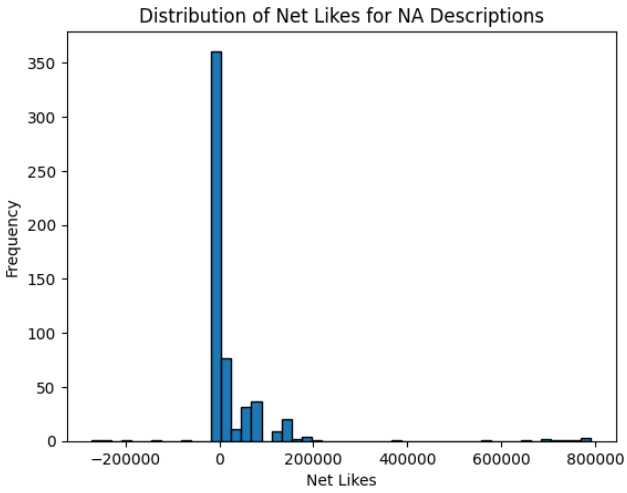

Figure 1. Distribution of Net Likes for NA Descriptions

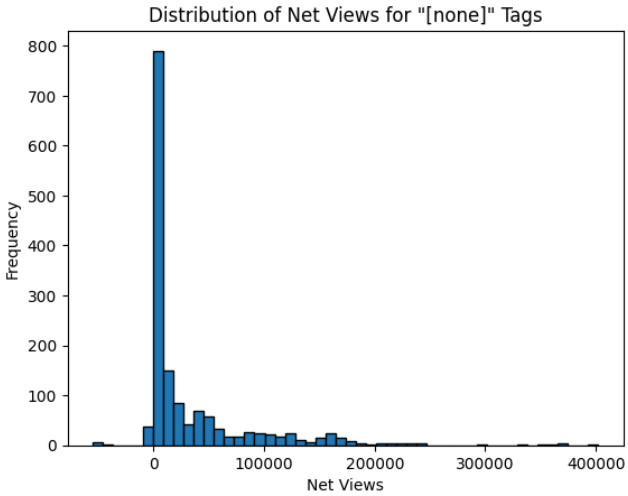

Figure 2. Distribution of Net Likes for NA Tags

3.3 Feature Selection

From the correlation plot, we can see that the net likes is highly correlated with views and comment counts, and is correlated with category id. Therefore, we can tentatively conclude that views, comment counts and category class would affect our prediction of popularity.

4 Predictive Task

4.1 Task Description

Our task is to predict the number of "net likes" of a given Youtube video, which characterizes the popularity of that video. To optimize the performance, we plan to reduce feature dimensions to improve the dataset's Signal-Noise Ratio and to reduce the training time.

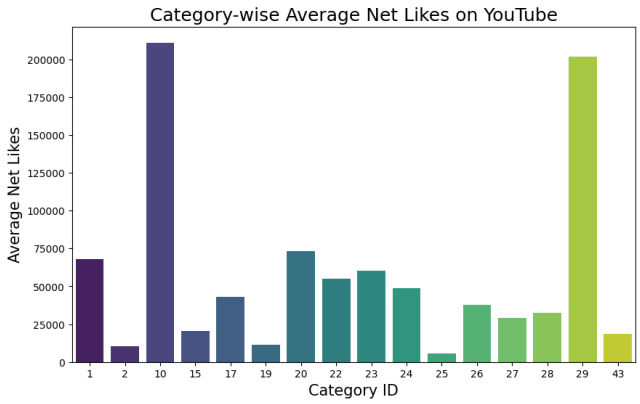

Figure 3. Category-wise Average Net Likes on YouTube

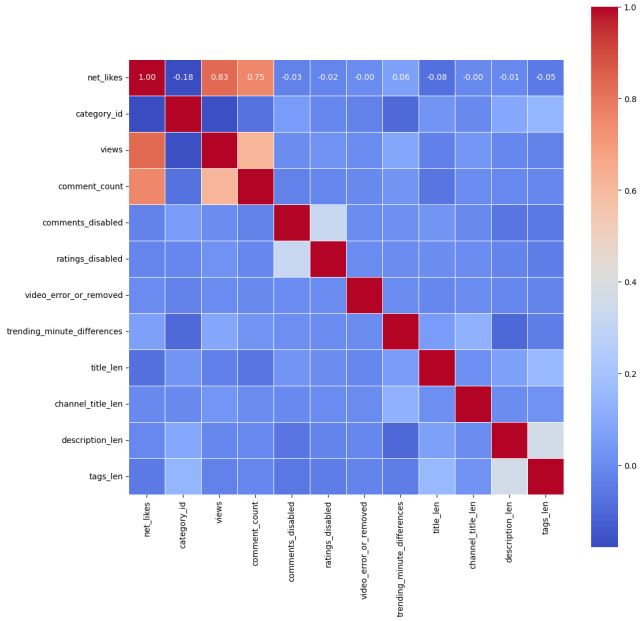

Figure 4. Correlation Plot between Features

4.2 Evaluation Metric

We plan to use Mean-Squared Error (MSE) to supervise our training. It evaluates how the predicted "net likes" deviates from the ground truth.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

5 Model

5.1 Linear Regression

5.1.1 Model Description. We selected the linear regression model, as the primary method for predicting video's net likes,which is popularity of that video. This decision was based on the following considerations: Firstly, linear regression is simple and easy to understand, providing intuitive explanations for the impact of features. Secondly, we assume a linear relationship between video's net likes and certain features. From correlation plot, we select ['category id', 'views', 'comment count', 'comments disabled', 'ratings disabled', 'video error or removed', 'trending minute differences', 'title length', 'channel title length', 'description length', 'tags length'] as explanatory variables.

The optimization of linear regression is as following:

$$\text{Minimize}\left(\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right)$$

where:

- y_i is the response variable for the i^{th} observation.
- β_0 is the intercept of the model.
- β_j is the coefficient for the j^{th} explanatory variable.
- x_{ij} is the j^{th} explanatory variable for the i^{th} observation.
- p is the number of explanatory variables.
- n is the number of observations.

5.1.2 Model Results. The fitted linear model coefficients and performance on test set is as following:

- Model Coefficients: [-1.408e+03 1.721e-02 2.2667e+00 -1.894e+04 -7.9798e+04 -2.4094e+04 2.774e+00 -2.963e+02 -7.3378e+02 3.928e+00 -4.928e+02]
- Model intercept: 67687.61737724596
- Mean Squared Error (MSE): 9.327086034e+09
- Coefficient of Determination (R^2): 0.785

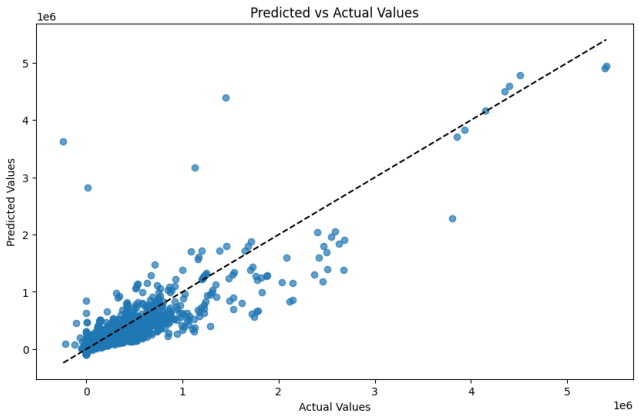

Figure 5. Performance of Linear Regression on Test Dataset

5.2 Random Forest

5.2.1 Model Description. Random Forest Regressor is a powerful and flexible model that is well-suited for predicting complex and nuanced outcomes like video 'net likes'. First, it can capture complex, non-linear relationships between features and the target variable. Second, This method is effective at handling interactions between different features (e.g., how 'views', 'comments', and 'tags' collectively influence 'net

likes') without requiring explicit feature engineering. Third, Due to its ensemble nature, where it builds multiple decision trees and averages their results, RF is generally more robust against overfitting.

$$Y_{RF}(X) = \frac{1}{n}\sum_{i=1}^{n} Y_{tree_i}(X)$$

where $Y_{tree_i}(X)$ is the prediction of the ith decision tree.

5.2.2 Model Optimization Strategy.

1. **Feature engineering**: We apply **one-hot encoding** on category id for categorical analysis instead of treating them as integer.
2. **Hyperparameter Tuning**: We employed grid search to determine the optimal hyperparameter (estimator numbers, max depth of decision tree) for Random Forest. Using cross-validation, we evaluated the model's performance under different hyperpameters' value and select best ones.
3. **Cross Validation**: We also implemented cross-validation to ensure a more reliable assessment of the model's performance, safeguarding against overfitting and confirming that the model generalizes well to unseen data.

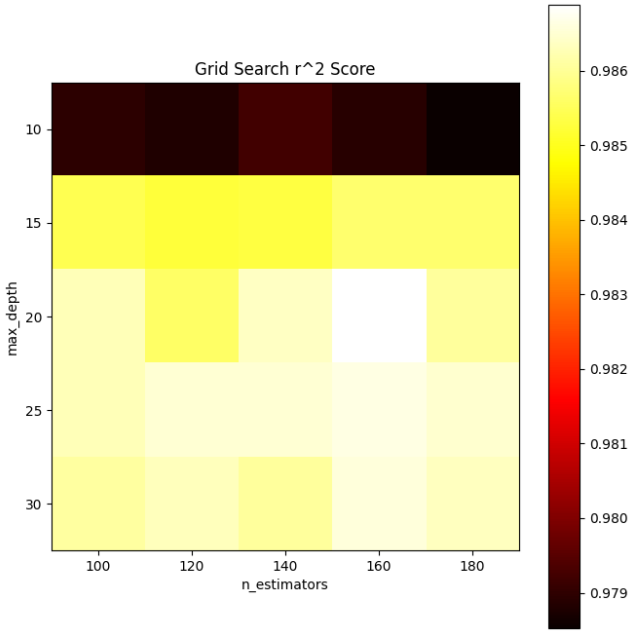

Figure 6. Grid Search for best max_depth and n_estimator of Random Forest

5.2.3 Model Results.

After hyper-parameter tuning, we find the best max depth is 25 and best number of estimators is 100. We train Random Forest again using these best parameters and test it. The results are:

- Mean Squared Error (MSE): 4.56656936e+08
- Coefficient of Determination (R^2): 0.9895

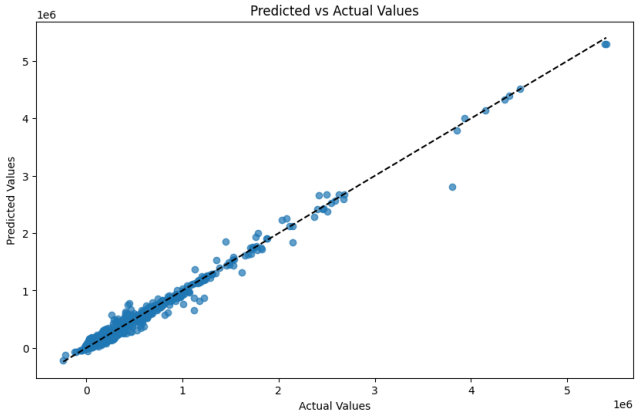

Figure 7. Performance of Random Forest on Test Dataset

Random Forest can also provide insights into the importance of each feature in predicting the 'net likes', helping in understanding which aspects of a video contribute most to its popularity. We can see that comment count is the most important feature, then views, tags length, and trending duration. Whether the video allows comments, ratings, or fails doesn't influence the video's popularity.

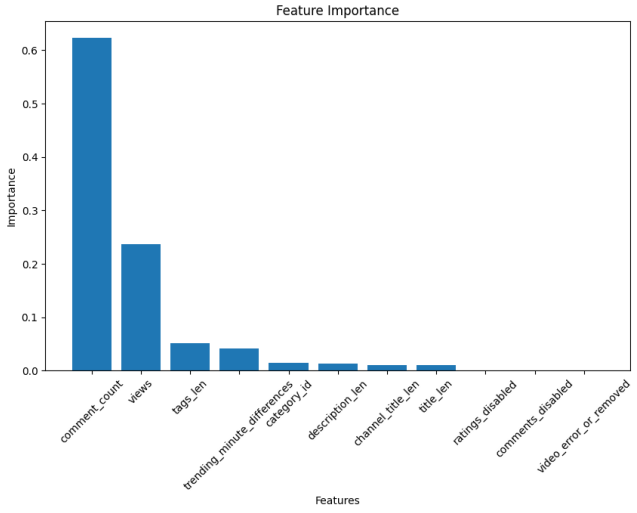

Figure 8. Feature Importance to Net Likes

5.3 Word-level Sentiment Analysis

5.3.1 Model Description.

From Feature Importance plot, we can see that tags are more influential than title, channel title, description on the prediction. **TF-IDF** (Term Frequency-Inverse Document Frequency) of tags is important for predicting 'net likes' on YouTube videos because it effectively captures the relevance and uniqueness of the tags in relation to the video content. **This method emphasizes tags that are distinctive and more specific to a particular**

video, rather than common and broadly used tags. Such a weighting mechanism helps in distinguishing videos that cover niche topics or specific interests, which often have a dedicated and engaged audience, potentially leading to higher 'net likes'. By converting tags into a quantifiable measure reflecting their significance, TF-IDF allows for a more nuanced and informative feature set in predictive modeling, thereby improving the accuracy of predictions regarding viewer engagement and preferences.

1. **Data cleaning**: We clean the tags using a simple texthero call, so that our feature representations (bag of words -> tf-idf) is not filled with bad vectors. The new review text would exclude punctuation, numbers, single chars, and multiple spaces.
2. **Text Embeddings**: We also tokenize the tags using a bag of words -> **TF-IDF** objects. Then, this object will be used to create text embeddings of tags, and feed them into Random Forest.

Each word's tf-idf value can be calculated as the following formula, then the tag is a tf-idf vector.

$$\text{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

$$\text{IDF}(t,D) = \log\left(\frac{\text{Total number of documents } D}{\text{Number of documents with term } t}\right)$$

$$\text{TF-IDF}(t,d,D) = \text{TF}(t,d) \times \text{IDF}(t,D)$$

where t is term, d is document, D is a set of documents.

5.3.2 Model Results. We add new tag tokens into our best Random Forest, and retrain them on the training set. The results of improved RF using TF-IDF is:

- Mean Squared Error (MSE): 3.70520608e+08
- Coefficient of Determination (R^2): 0.9915

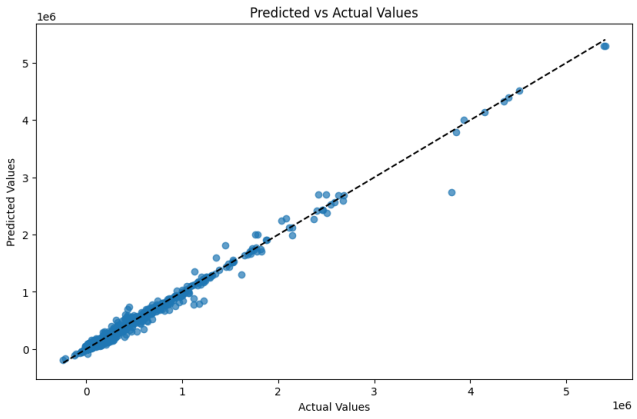

Figure 9. Performance of TF-IDF Random Forest on Test Dataset

5.4 Neural Network

5.4.1 Model Descriptions. We used a neural network to predict the number of net likes given the features of the YouTube videos as inputs. The network architecture is shown in Figure 10, where we first pass the input to a group of convolutional layers ($Conv(64, 128, 256, 512, 1024)$) to transform the video features to higher dimensions where they can be more distinguishable. We use Batch Normalization layers ($BN(128, 256, 512, 1024)$ in between the convolutional layers to stabilize feature values and to avoid the gradient vanishing/exploding problems. After that, for inference, We use a Multi-Layer Perceptron ($Dense(1024, 512, 256, 128, 64)$) with an output layer of dimension 1 to regress the log net likes. Since we use log net likes for model training (to be discussed in the next section), we exponentially transformed the outputs in the end to get our net likes' predictions.

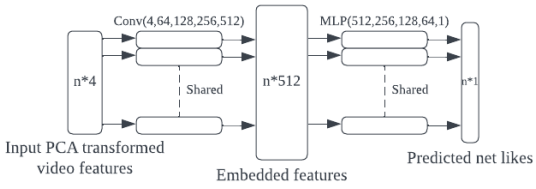

Figure 10. Neural Network architecture

5.4.2 Data pre-processing and feature transformations. To filter less important features, We adopted Primary Component Analysis to pre-process the cleaned data to reduce the feature dimensions. Specifically, We tried different numbers of components p and finally concluded $p = 4$ as our choice. In addition, We performed text sentiment classification on the channel title and the title using the DistilBERT [5] model implemented in the pipeline package. We computed the sentiment score for each of the texts (ranging [-1,1], positive for positive moods, and negative for the opposite) and used them as input features to perform the PCA and to feed the neural network. Since the net like values are having a wide range, we apply the following log transform on the values:

$$logTransform(x) = \begin{cases} -log(-x) & \text{x<0} \\ 0 & \text{x=0} \\ log(x) & \text{x>0} \end{cases} \quad (1)$$

So that we can re-scale the net likes to a range that can be smaller and more distinguishable.

5.4.3 Sentence-level Sentiments of Title and Channel Title. For better net likes' predictions, We feed the title and the channel title to the fine-tuned DistilBERT [**?**] model for classifications. It gives a sentiment score ranging [-1, 1] representing whether the title/channel title has positive or

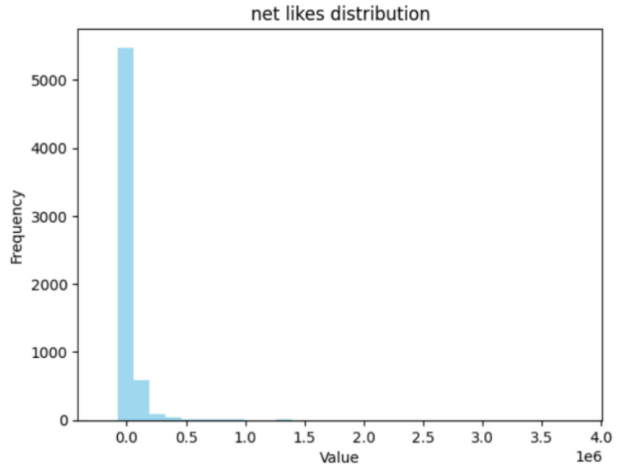

Figure 11. Raw net like values

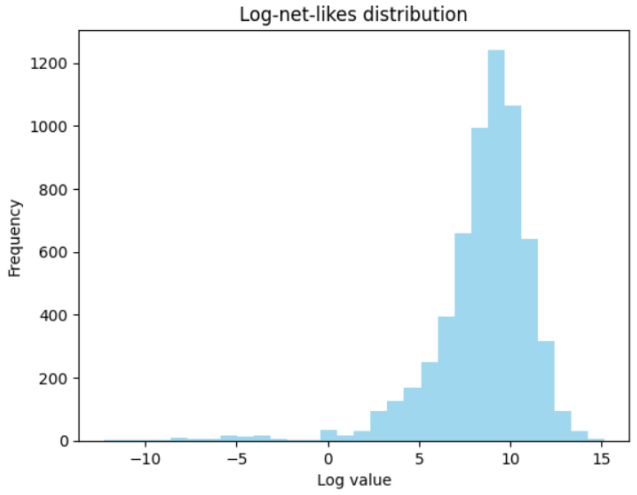

Figure 12. Log transformed net like values

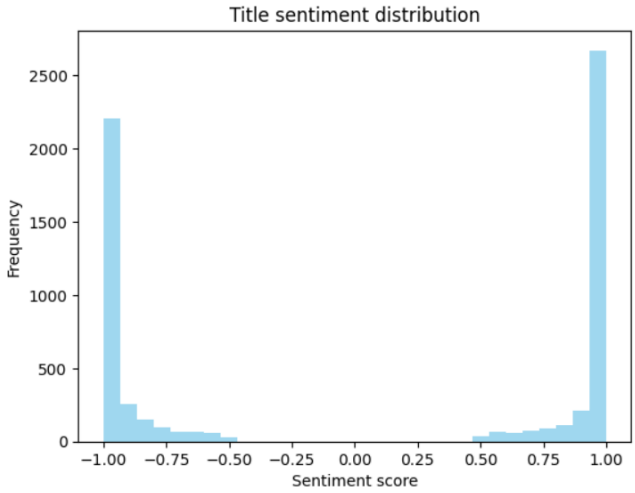

Figure 13. Video title sentiment distribution

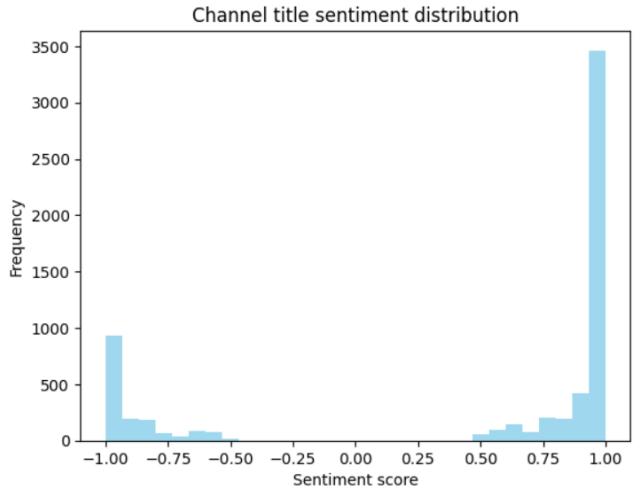

Figure 14. Channel title sentiment distribution

negative sentiments (with its sign), as well as how positive or negative that can be.

The distributions of the video titles' sentiments and the channel title's sentiments are shown in figure 14. We can see that they're polemic and confident enough to characterize whether the video's title or the channel's title has a positive or negative sentiment.

5.4.4 Results. To compare the performance under different PCA decompositions, We re-formulated the task into a video popularity binary classification one: We threshold the net likes with 10000, where the data above are marked popular while the others are marked unpopular. In this way, the efficacy of the PCA decompositions can be demonstrated by the prediction accuracy more intuitively.

The accuracy results for performing classification without PCA and with PCA components (p) to be 2, 3, 4 are shown in figure 16. We can see that the neural network itself can give satisfying predictions even without having any degree of PCAs (i. e. with p=2 or 3), indicating that the neural network can effectively learn the features. The model reaches the optimal performance when p=4, which can be the hyper-parameter that we choose for the remaining explorations using sentimental analysis.

To demonstrate the effectiveness of adding sentimental analysis, We chose $p = 4$ according to the experiment above, regressed on the net likes, and used Mean Squared Error (MSE) to characterize the model performances. The results are shown in the table below, where "No PCA+MLP" means not applying PCA and using only dense layers Dense(64, 128, 256, 512, 256, 128, 64) for the network, "Conv" meaning using convolution layers $Conv(64, 128, 256, 512)$ to replace the first 3 dense layers, and "PCA" meaning using PCA $p = 4$ to reduce

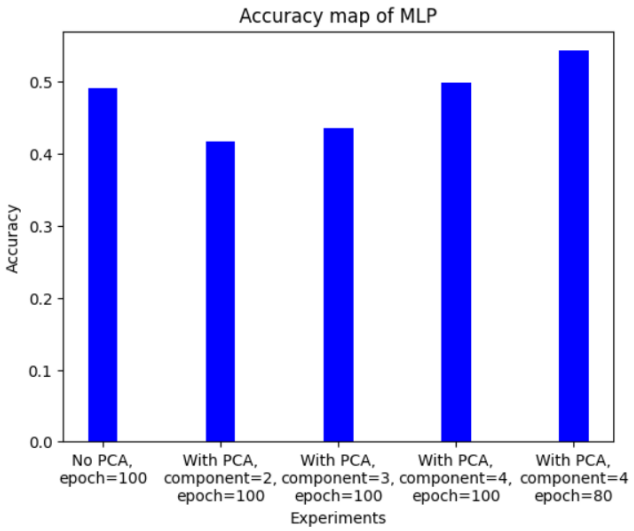

Figure 15. Accuracy map with PCA

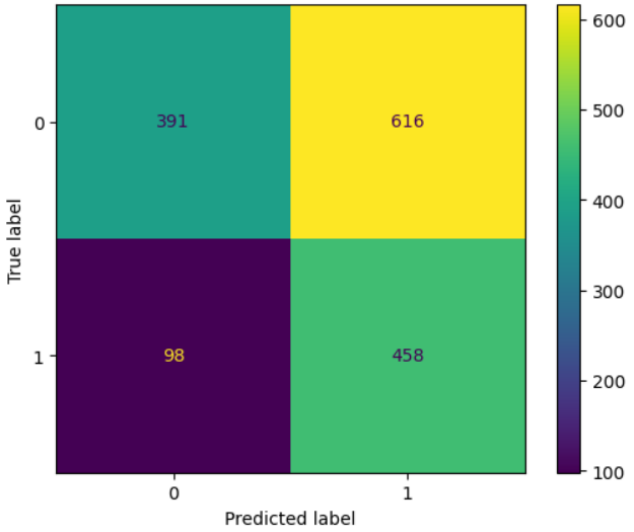

Figure 16. Confusion matrix under p=4, without sentiment analysis

feature dimensions. We can see that adding convolutional layers helps improve the performance by extracting features more effectively. More importantly, we can observe that adding sentimental features to train the model improves its prediction of net likes, demonstrating a potentially strong significance that such features can have on the net likes.

6 Results in summary

The table for results, in summary, can be written here: We can see that simple neural networks, though augmented with PCA, show the worst performance because of their limited model complexity (simply the concatenation of a set of convolution layers and a set of dense layers) in comparison to larger and more complex deep neural networks. Despite this, adding sentence-level sentiment features helps improve the network performance. In addition, after introducing randomness to models and the data sampling process, Random Forest performs much better than the simple Linear Regression methods. The TF-IDF-based word-level sentimental analysis is fine-grained enough to mine significant sentimental features from data, which helps improve the performance of Random Forest when trained together.

Table 1. Neural Networks Performance

Model	MSE
No PCA+MLP	3.289e10
No PCA+Conv+MLP+No senti	2.822e10
PCA+Conv+MLP+No senti	2.443e10
PCA+Conv+MLP+Senti	2.143e10

Table 2. Model Performance Comparison

Model	MSE
PCA+Conv+MLP	2.443e10
PCA+Conv+MLP+SentiCls	2.143e10
Linear Regression	9.327e09
Random Forest	4.567e08
Random Forest + TF-IDF	3.705e08

7 Conclusions

In conclusion, during this project, we discovered the effectiveness of word-level sentimental features, the TF-IDF values characterizing the tags' relevance and uniqueness related to the video content, in net likes' predictive task. We also explored the efficacy of the sentence-level sentiment features, the text sentimental scores indicating the positiveness or negativeness of the overall sentence, on building a simple Neural Network. We concluded that such sentimental features, either coarse-grained in sentences or fine-grained in words, are distinctive and predominant enough to improve the model's learning capability on the data. Such discovery follows our intuition that humans favor more eye-drawing titles/descriptions, too.

References

[1] Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (2018), 65–170.

[2] Chin-Hui Lai and Chia-Yu Hsu. 2021. Rating prediction based on combination of review mining and user preference analysis. *Information Systems* 99 (2021), 101742.

[3] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105* (2019).

[4] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. 2002. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*. 127–134.

[5] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[6] Kai Zhang, Keqiang Wang, Xiaoling Wang, Cheqing Jin, and Aoying Zhou. 2015. Hotel recommendation based on user preference analysis. In *2015 31st IEEE International Conference on Data Engineering Workshops*. IEEE, 134–138.

[7] Kaiqi Zhao, Gao Cong, Quan Yuan, and Kenny Q Zhu. 2015. SAR: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews. In *2015 IEEE 31st international conference on data engineering*. IEEE, 675–686.

Received 19 December 2023