CS 7641 Machine Learning Final Report

Christine Olds, Christian Lee, Jeongrok (Michael) Yu,
Tae Eun (Harrison) Kwon, and Alexandra Pfleegor

I. Introduction/Background

TED Talks have been around since 1984, but were not publicly available until June 2006, when they were published online. Now TED Talks are watched or listened to around 3 billion times annually [1]. Thus, we decided to ask: what makes a good TED Talk?

A. Literature Review

Extensive research investigates TED Talks' role in popularizing topics [2] [3] [4]. Nonlinear methods, including Support Vector Regression and Gaussian radial basis functions, enhance the accuracy of predicting a video's popularity [5]. Sentiment Analysis is used to extract a text's emotional tone, classifying it as negative, neutral, or positive [6]. Apriori association rule mining can cluster groups of commonly occurring keywords [7]. Features such as keywords, sentiment, etc. can be used to classify and predict the popularity of TedTalks [8].

Encoder-decoder models, unlike BERT or GPT models, can take in the input sequence and generate embedding using the encoder, then produce an output sequence. This has been pivotal in the advances in numerous fields of Natural Language Processing including text summarization. Most of the state-of-the-art models now use encoder-decoder models like BART and T5 and fine-tune the models with human-annotated summaries [9] [10] [11]. However, there has been a boom in decoder-only models such as the Chat-GPT model, thanks to their scalability in pre-training data and the incorporation of Reinforcement Learning with Human Feedback (RLHF) [12]. We believe that it would be interesting to see how Chat-GPT performs when only incorporating prompt engineering and plan to compare the performance between the two different architectures of models.

B. Data Description and EDA

The dataset comprises data on 5,662 TED Talks from 1972 to 2022 and includes the features shown in Table I for each talk. Exploratory data analysis was performed in a Jupyter Notebook using a Python 3 kernel.

1) Numerical Features: The expected numerical columns are `_id`, `duration`, `likes`, and `views`. A summary of the dataset revealed that the `likes` column was being read as a string rather than an integer. To fix this, we created a function that converts the string into an integer and multiplies it by a factor of 1,000 if there was a "K" at the end of the string or a factor of 1,000,000 if there was an "M" at the end of the string. The four numerical features are now `_id`, `duration`, `likes`, and `views`.

TABLE I
Data set Attributes

Attribute	Description
`_id`	unique dataset identifier for each TED Talk
`duration`	duration in seconds of each TED Talk
`event`	TED event at which the talk was given
`likes`	number of likes of each TED Talk, to a factor as indicated [1]
`page_url`	official link to video on TED website
`published_date`	date that TED Talk was published on official TED website
`recorded_date`	actual date that TED Talk was performed (may differ from published date)
`related_videos`	list of video IDs related to respective TED Talk
`speakers`	list of speakers, including each speaker's name and occupation description
`subtitle_languages`	list of subtitle languages available, including the language as well as a code [2]
`summary`	summary of the TED Talk
`title`	title of the TED Talk
`topics`	a list of topics, including a unique identifier for each topic
`transcript`	a complete transcript of each TED Talk in English
`views`	number of views that each TED Talk has
`youtube_video_code`	link to the YouTube video listing

[1] For example, a "K" indicates thousands, while an "M" indicates millions.
[2] For example, English is listed with its respective code, "en".

The EDA revealed right-skewed histograms for views and likes, which we acknowledge can be attributed to the exponential growth YouTube and the fame TED talks have gained over the years as well. The polarity is evident in that the most viewed TED talk within our dataset has over 73 million views while the least-watched one has only 587 views. However, we inspected what were the two outliers within the 'durations' category, as a TED talk over 4 hours long was irregular. One of the outliers was a collection of 10 or more TED Talks talking about climate change and the other one was a countdown video for a live TED conference. We concluded that these talks while talking about one topic, are not similar to other talks, and thus decided to exclude these two videos. Figures 1, 2, 3 display box and whisker plots of `duration`, `likes`, and `views`, respectively.

Next, we checked for correlation between the numerical columns. As expected, there was a high correlation between likes and views. Figure 4 shows the correlation matrix. Thus, we decided to only use the `likes` variable in our final models as our measure of popularity.

2) Categorical Features: The main categorical features we were interested in exploring were `speakers`, `subtitle_languages`, and `topics`. The data for these

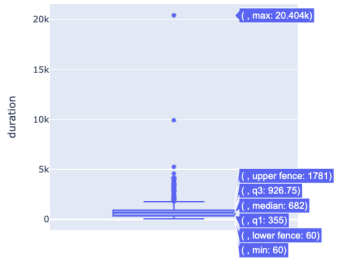

Fig. 1. Box and Whisker Plot for Duration: $\mu = 707.692, \sigma = 521.179$

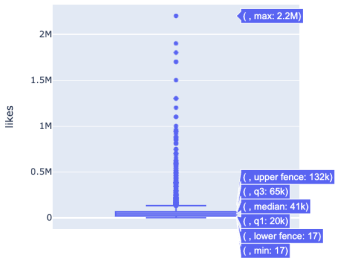

Fig. 2. Box and Whisker Plot for Likes: $\mu = 63,362.765, \sigma = 108,900.280$

columns were originally in dictionary format. To make the information more accessible, we extracted the values from the dictionaries and created three new columns: `speakers_list`, `sl_list` (subtitle languages list), and `topics_list`.

Using this information, we were able to see that speakers oftentimes return, with the top three speakers being Alex Gendler with 48 videos, Iseult Gillespie with 35 videos, and Matt Walker with 17 videos. It will be interesting to see if these speakers tend to have more 'popular' videos. Additionally, we found that the most commonly presented topics are science with 1267 videos, technology with 1231 videos, and TED-Ed with 878 videos.

3) Transcript: Since we plan to use the calculated sentiment of the transcript in the regression model to predict popularity, the transcript is an integral part of the dataset. All text cleaning will be explained in the data preprocessing section. During EDA, we noticed that 648 videos were missing transcripts. Since we cannot recreate or estimate these transcripts, we will be excluding those videos from the dataset.

4) Time Series: Since we have the recorded and published dates from the data, we plotted the number of talks published per year. This graph is shown in Figure 5.

C. Dataset Link

The dataset can be accessed on Kaggle using this link.

II. Problem Definition

A. Problem

- Create a regression model to predict the popularity of TED Talks based on association rule mining and sentiment of the transcript
- Compress long transcripts and compare the summaries with the ground truth

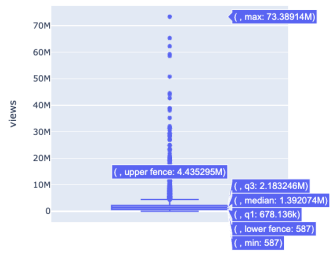

Fig. 3. Box and Whisker Plot for Views: $\mu = 2,130,394.214, \sigma = 3,679,481.018$

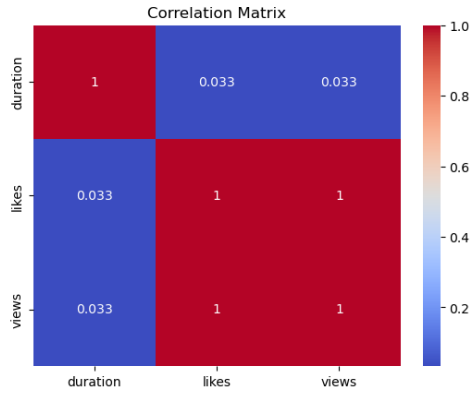

Fig. 4. Correlation Matrix

B. Motivation

- Understand how the commonly used words and sentiments of a TED Talk affect its popularity.
- Challenge state-of-the-art text summarization models to encapsulate a long transcript into short sentences.

III. Methods

A. Data Preprocessing

As seen in Figure 6, we utilized several steps to clean and process our data.

1) Popularity Calculation: After some basic data cleaning, we created our popularity metric using the following equation:

$$\texttt{popularity} = \log(\texttt{likes})$$

This changed from our original equation of

$$\texttt{popularity} = \left(\frac{\texttt{likes}}{\texttt{views}}\right) * 100$$

The reason for this change will be discussed in more detail below. We utilize the log of `likes` as our proxy for popularity, although this is an imperfect measure. The choice of taking the log of `likes` comes from the distribution of the `likes` variable being skewed. After taking the log of `likes`, the distribution becomes closer to a normal distribution.

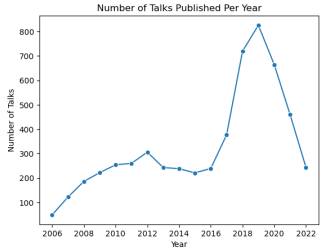

Fig. 5. Number of TED Talks Per Year

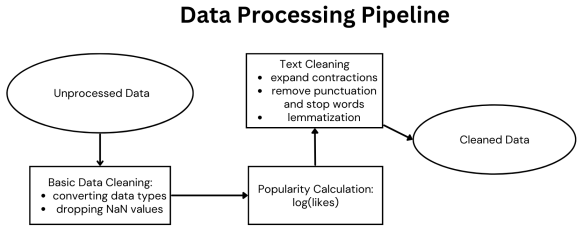

Fig. 6. Data Processing

2) Text Cleaning: We also needed to clean the textual data for our sentiment analysis algorithms. Using the contractions package, we first expand all contractions. For example, we turn 'I'm' into 'I am' and 'can't' into 'can not'. This becomes useful once we remove punctuation and stopwords in the next step. After that, we lemmatize all words in each transcript before putting it all back into one single string.

For the text generation task, we further split the data into sentences because each input is in a sentence form for encoder-decoder models before being tokenized. It was imperative to distinguish acronyms and punctuation so that the model does not confuse periods in the U.S. as the punctuation marker for ending a sentence. Most of the summarization language models have a maximum token limit for the encoder. Due to the exceeding length of our transcripts, we had to further split one transcript to not exceed the limit and made sure the end of each split was at the end of a sentence. The column with the processed transcript is then called `processed_transcript`.

3) Feature Engineering: In our initial stages of data preprocessing, we leveraged prompt engineering via the OpenAI API to extract emotional content from TED Talks, resulting in 'detected emotion' data formatted in JSON. This process was pivotal in mapping the complex emotional responses elicited by these talks. For the final phase of our analysis, we successfully transformed these JSON-formatted emotion data into numerical features through a meticulous encoding process. Each emotion detected has been converted into categorical variables, allowing us to quantitatively assess their impact on the talks' popularity.

With these newly encoded features integrated into our SVR model, we have enhanced our ability to predict the popularity of TED Talks. Our analysis is built on the hypothesis that the emotional resonance of a talk significantly influences its viewer engagement and appeal. This integration marks a significant advancement in our modeling efforts, promising more accurate predictions by capturing the nuanced emotional drivers behind audience reactions.

B. Machine Learning Algorithms/Models

1) Regression: We tinkered with the implementation of several different types of regression models to predict the popularity of TED Talks, including linear regression, lasso regression, and support vector regression (SVR), all from Python's `sklearn` library.

Linear regression was the initial model attempted because of its simplicity and interpretability. The coefficients from the linear regression allow for a quick analysis of significant variables. Due to the large amount of features in the model, the next model attempted was lasso regression. Lasso regression acts as a variable selection technique by forcing coefficients of insignificant predictor variables to 0. Since we found that other models have had success predicting the popularity of success using the Gaussian radial basis function kernel during our research, we implemented this kernel, called 'rbf'. Additionally, we also attempted a linear, polynomial, and sigmoid kernel.

We took a few approaches to building our SVR model. First, we focused on the topics list to add more features to our model by one-hot encoding the topics for each TED Talk, focusing on only the topics that appeared at least 100 times in the dataset to avoid overfitting. Other features included the TextBlob and VADER scores calculated from the sentiment analysis algorithm. The linear regression model produced an MSE of 0.0014 and an R^2 value of 0.0038. Among the lasso regression model as well as the four SVR models, the linear regression model had the best results.

The results prompted further exploration of the response variable, popularity. The distribution is shown in Figure 7. To

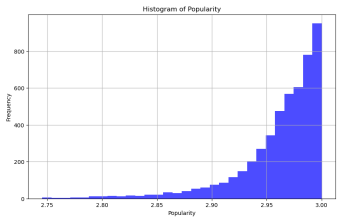

Fig. 7. Histogram of Popularity

find a better metric to predict, we experimented with taking the log of likes. The distribution is shown in Figure 8. Since the log of likes is more normally distributed than popularity, we decided to predict the log of likes instead of the popularity. This produced higher R^2 values, which are discussed in the results section below. The second approach to building

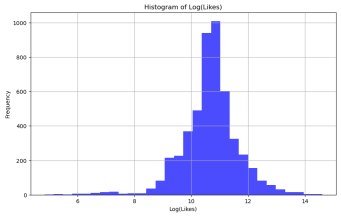

Fig. 8. Distribution of Log(Likes)

the regression model used the detected emotions as features. Other features used included the TextBlob and VADER scores calculated from the sentiment analysis algorithm. See the results section for discussion and analysis of the results.

2) Sentiment Analysis: Sentiment Analysis is utilized to find the perspective, view, or attitude of the writer by using computational methods to quantify and categorize a writer's sentiment [6]. For this project, we utilized two different lexicon-based sentiment analysis tools, TextBlob and VADER, as our unsupervised methods. Both TextBlob and VADER assign a score between -1 and 1, where -1 signifies a negative sentiment, 0 signifies a neutral sentiment, and +1 signifies a positive sentiment [13]. After data preprocessing (see data preprocessing section for more information), the TextBlob and VADER algorithms were applied to the processed transcript. New columns containing the scores for each respective algorithm were added to the dataset. See the results section for a discussion of our analysis of the algorithms.

3) Comprehensive Feature Integration: Our refined SVR model integrated not only the topics and sentiment scores but also the occupations of speakers and the quantified emotions, enhancing the model's predictive capability. This comprehensive integration allowed for a more detailed exploration of how various aspects of a TED Talk—from the content's thematic and emotional depth to the speaker's professional background—impact its popularity.

The integration of these diverse features resulted in improved model performance, underscoring the complex interplay between content quality, emotional engagement, and speaker attributes in determining the success of TED Talks.

4) Association Rule Mining: Association Rule Mining is used to uncover relationships, or "rules", between variables in a large dataset. The model uses metrics such as support, confidence, and lift to measure the strength of these rules [7]. For this project, we utilized association rule mining to extract commonly used words and word clusters from TED Talk titles. The support of a rule represents the fraction of all titles that demonstrate that rule. The rules that generated the highest support included single words such as "life", "world", "us", and "future". The confidence of a rule measures the strength of the association between its items, that is, how likely an item is to appear given the other item's presence. The rules that generated the highest confidence included word clusters such as {"brief", "history"}, {"solve", "riddle"}, and {"climate", "action"}.

The original goal in conducting association rule mining was to use the rules it produced and their associated metrics as features in our regression model to predict the popularity of TED Talks. The idea was that extracting common words or strongly associated word clusters may reveal trends of intentional titling of videos for the purpose of attracting the attention of potential viewers (i.e. "clickbait"). However, a correlation matrix of rule and popularity metrics reveal weak or counterintuitive relationships, as shown in Figure 9. Thus, we concluded not to append the association rule mining metrics to our regression dataset.

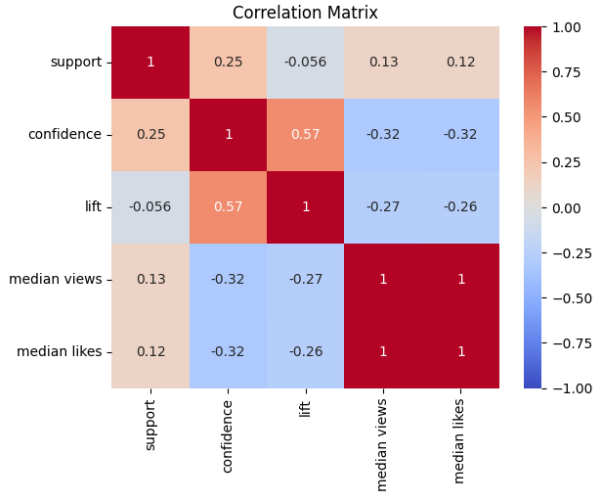

Fig. 9. Correlation Matrix

Upon the completion of extracting detected emotions from TED Talks, we needed to conduct some EDA on the engineered features before using the emotions to enhance our popularity prediction model. We also wanted to explore the possibility of predicting the emotion of a TED Talk based on features such as talk topic, transcript, etc. Thus, another opportunity was introduced to apply association rule mining to explore the extracted emotions set and its relationship with other features. Examining the associations between the topics of a TED Talk and their corresponding emotions, we found that the strongest rules simply map the most prevalent topics to the emotion 'Inspired'. Upon further analysis, we found that nearly 90% of their entire set of TED Talks included 'Inspired' as one of the top three emotions. While this observation aligns with our expectations given the context and nature of TED Talks, the severely imbalanced nature of the dataset may pose an issue when trying to train a classifier model to predict the emotion of a TED Talk. A classifier that maps every TED Talk to 'Inspired' may demonstrate a misleadingly high accuracy, despite completely ignoring all the other emotions. Possible solutions may include oversampling TED Talks that do not exhibit the 'Inspired' emotion, or completely removing 'Inspired' from the set of emotions and instead classifying TED Talks by the remaining emotions.

5) Text Summarization: Another supervised method we utilized was text summarization. For this, we first had to come up with a clever way to split the long transcripts into smaller chunks, as all the encoder-decoder models we plan to use have a maximum token limit. Therefore, we first tokenized the transcripts into contextual tensor embeddings using the respective models (t5 [9] and BART-large model [10]) and determined the splitting points while maintaining the sentence information. Next, we subsampled about 10 videos that we all personally watched, and conducted summarization on this small batch before scaling our experiments to validate if this hierarchical summarization method is valid.

First, we were planning to utilize the BART summarization model to summarize the splits of transcripts to output a compressed transcript that was shorter than the maximum token size, then run the summarization model again on this compressed transcript to shorten the summary like the one-sentence ground truth summary within our TED dataset. In this hierarchical summarization using pre-trained language models to initially truncate the transcript, we often faced hallucination, meaning that the model was outputting the same sentences or repeating words from previous steps. After reading relevant papers on text summarization, we realized that this was a known problem for hierarchical summarization on the pre-trained language models, and hallucination is especially a problem for models fine-tuned on the "XSum" dataset [14]. Consequently, we devised a different approach in initially shrinking the transcript.

The solution to hallucination was utilizing an extractive summarization method using the Sumy module to compress the transcript initially. Within Sumy, we integrated LexRank, an extractive summarization algorithm based on PageRank algorithm [15]. LexRank first computes the similarity between sentences in the text, constructs a graph representation of the text, iteratively ranks the text based on the similarity of the graph representation, and extracts the highly ranked sentences. Given this extraction, we now feed it to the t5 [9] and BART-large [10] summarization models (both fine-tuned on CNN/Daily_Mail dataset) and obtain more relevant and non-hallucinating results.

The final method we tried for the text summarization task is just using a decoder-only model, ChatGPT, to summarize our long transcript. ChatGPT is advantageous when compared to other models in that it has a much larger maximum token length of 16,385 tokens. This means that ChatGPT can take in the transcript at once, without losing any information about the transcript during summarization. All of our experiments regarding text summarization tasks were conducted on Google Colab with a Tesla V100-SXM2-16GB GPU. An example of our summarization result is shown in Table IX within the Appendix.

IV. Results and Discussion

A. Regression Analysis

We calculated the R^2 and mean squared error (MSE) to measure the accuracy of our models. The results for the regressions using topics as predictor variables are shown in Table II.

TABLE II
REGRESSION MODELS - TOPICS

Model	R^2	MSE
Linear	0.0919	0.8875
Lasso	-0.0032	0.9805
SVR - Linear	0.0962	0.8833
SVR - RBF	0.1276	0.8526
SVR - Poly	0.0529	0.9256
SVR - Sigmoid	-3.761	4.6538

The results for the regressions using detected emotions as predictor variables are shown in Table III.

TABLE III
REGRESSION MODELS - EMOTIONS

Model	R^2	MSE
Linear	0.4088	1.3338
Lasso	-0.0040	2.2656
SVR - Linear	0.3889	1.3788
SVR - RBF	0.5246	1.0726
SVR - Poly	0.4461	1.2496
SVR - Sigmoid	-61.1057	140.1363

As evidenced by the results in Tables II and III, the models that utilized detected emotions as a feature tended to have much higher R^2 values, indicating that these models fit the data better.

Following initial regression analyses using topics and detected emotions as predictors, we expanded our model to include speaker occupations alongside a detailed emotional response set. This comprehensive approach hypothesizes that the interplay between speaker background, thematic content, and emotional impact collectively influences TED Talks' popularity.

- **Emotions:** Quantified emotional impacts extracted from the talks.
- **Topics:** Frequently appearing topics, one-hot encoded.
- **Occupations:** Categorical representation of speakers' professional fields.

Extensive fine-tuning was conducted using support vector regression with different kernels. The systematic grid search identified the RBF kernel as the most effective. The results of this comprehensive model are presented below:

TABLE IV
INTEGRATED REGRESSION MODELS

Model	R^2	**MSE**
SVR - RBF (Fine-Tuned)	0.5144	1.0462
SVR - Linear (After LASSO)	0.4965	1.0654

These results signify the SVR model using the RBF kernel, after fine-tuning, achieved an R^2 score of 0.5144 and an MSE of 1.0462. This marks a significant improvement and suggests that the model effectively captures the factors influencing the popularity of TED Talks.

Building on this foundation, we further explore how emotions, topics, and occupations contribute as key predictors in our model, each adding a unique dimension to our understanding of what drives audience engagement.

1) Emotion Coefficients: Inspired (0.5185) is the most influential emotion, with inspiring content tending to receive more likes due to its uplifting nature and deep resonance with viewers. Curious (0.3636) has a significantly impactful effect, as content that sparks curiosity effectively engages viewers. Moved (0.3303) creates a strong connection with the audience, enhancing likeability.

2) Topic Coefficients: Psychology (0.0961), Brain (0.0942), and Personal Growth (0.0741) engage and relate to viewers on self-improvement and understanding human behavior. Culture (0.0562) and Entertainment (0.0451) positively influence likes, highlighting viewer interest in cultural and entertainment content. Conversely, Music (-0.0250) exhibits a slight negative influence, likely due to oversaturation or specific viewer preferences.

3) Occupation Coefficients: Educator (0.0950) indicates that videos featuring educators are more liked, likely due to their informative nature. Singer (-0.0108) and Singer-songwriter (-0.0005) show a neutral to slightly negative impact, reflecting specific viewer preferences or contextual factors in their appearances in videos.

4) Overall Insights: Strong, positive emotions are highly effective at driving likes; content creators should focus on producing inspiring, enlightening, and emotionally engaging content. Focus on engaging topics like psychology and personal growth is advised, while being cautious with music-related content due to potential oversaturation. Videos featuring educators perform well, suggesting a preference for educational content, which could be leveraged in content planning and marketing strategies.

B. Sentiment Analysis

After data preprocessing and text cleaning (see data preprocessing section for more information), we applied two sentiment analysis algorithms, TextBlob and VADER, to the transcripts. The disparity in the scores was quite interesting. TextBlob tended to score the transcripts more conservatively as a 'neutral' sentiment around 0. Table V the summary statistics from the TextBlob algorithm.

TABLE V
TEXTBLOB AND VADER RESULTS

	Min	Mean	Median	Max
TextBlob	-0.259643	0.095959	0.098674	0.594444
VADER	-0.999900	0.656523	0.996300	1.000000

VADER tended to assign more extreme sentiment scores closer to -1 or 1. Table V also shows the summary statistics from the VADER algorithm.

The disparity between the two algorithms is apparent in Figure 10 where blue is the TextBlob algorithm and orange is the VADER algorithm. Since sentiment analysis is an unsupervised learning method, we created a ground truth by manually reviewing 50 videos and assigning a score between -1 and 1. We then compared our scores to the scores assigned by TextBlob and VADER by calculating the MSE.

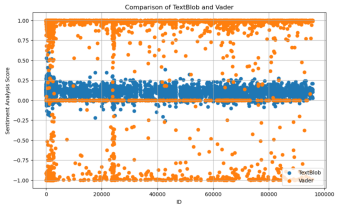

Fig. 10. TextBlob vs. VADER Comparison

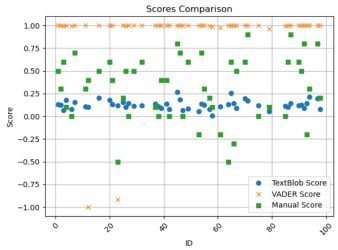

Fig. 11. Comparison Between TextBlob, VADER, and Manual Scores

TABLE VI
SENTIMENT ANALYSIS SUMMARY STATISTICS FROM SUBSET OF DATA

	Min	Mean	Median	Max
TextBlob	0.007780	0.127299	0.122397	0.268511
VADER	-0.999800	0.919600	0.999500	0.999900
Manual	-0.500000	0.320000	0.400000	0.900000

For the 50 videos we manually scored, the summary of statistics is shown in Table VI. The distribution of scores for these 50 videos can be visualized in Figure 11 where blue is the TextBlob score, orange is the VADER score, and green is the manual score. The MSE's of the TextBlob and VADER algorithms when compared to the manual scores were 0.15 and 0.57, respectively. See Figure 12 for a visualization of the MSE scores, where blue is the TextBlob algorithm and orange is the VADER algorithm.

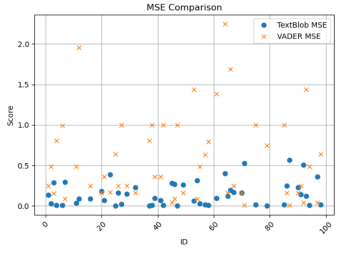

Fig. 12. TextBlob vs. VADER MSE Scores

Therefore, based on these 50 videos, TextBlob is a more accurate algorithm compared to VADER. We think this is because VADER is intended for analyzing social media text and reviews of movies or products [13]. According to the

mission statement of TedTalks, their goal is "discovering and spreading ideas that spark conversation, deepen understanding and drive meaningful change" [1]. Thus, the fact that the scores of our sentiment analysis algorithms tend to be more positive is expected with the general sentiment of TedTalks.

C. Sentence Embedding Similarity Scores

We initially proposed to evaluate our summaries using automatic summarization or generation metrics such as BLEU or ROUGE score [16] [17]. However, we realized that our goal was not to replicate the exact summary but to retain and explain the transcript in essence. Therefore, we modified our metrics for the text generation task to measure the similarity of the summaries generated by different models.

For each output summary using different models, t5, BART, and ChatGPT, we calculated the sentence contextual embedding of the summaries and evaluated the cosine similarity score with the ground-truth summary provided by the dataset. We did this by first tokenizing the generated summaries with the respective model, feeding them into an encoder to obtain contextual embedding, and computing the dot product of embedding divided by the product of the length of their length. Integrating contextual embedding that has different tensors for the same word when given different contexts was crucial in calculating similarity scores. The similarity score is in the range of 0 to 1, where 0 means that the two tensors are orthogonal and dissimilar and 1 means that the angle between the two tensors is smaller, implying similarity. The results of the similarity metrics are shown in Table VII.

TABLE VII
COSINE SIMILARITY OF TEXT SUMMARIZATION WITH GROUND-TRUTH

Model	Similarity Metrics
Lex-Bart	0.79468
Lex-t5	0.61622
ChatGPT	0.99851

We observed that the hierarchical pre-trained models like BART and t5 did not outperform the state-of-the-art language model which is ChatGPT. We acknowledge that the cosine similarity metric is limited in that it ignores the order and semantic meaning of the summaries, which leads to longer summaries with more information having higher scores. This is attributable to the extremely high score of the text summarization results from ChatGPT as the average length of the summaries almost triples from the summarization of encoder-decoder models. We tried to make a fair comparison by limiting the length of the output for ChatGPT models, but this led to ChatGPT stopping short in the middle of the summaries most of the time and not producing good summaries.

V. CONCLUSION

In conclusion, our study has highlighted the challenges encountered in predicting the popularity of TED Talks videos. Through innovative approaches in feature engineering, such as various sentiment analysis algorithms, association rule mining, and monotonic transformation of a feature, we significantly enhanced the performance of our popularity prediction models.

Building upon these promising results, our future work aims to extend the applicability of our regression models to predict the popularity of other YouTube videos with similar characteristics. Moreover, our research agenda includes addressing hallucination issues in hierarchical summarization models for encoder-decoder-based models. By leveraging advanced natural language processing techniques and pre-trained language models, we aspire to generate concise and informative summaries of other contents as well. In essence, our study not only leveraged data from TED talks but also demonstrated the effectiveness of innovative feature engineering techniques in predicting video popularity. Furthermore, our exploration of NLP methodologies underscores our commitment to advancing summarization techniques for diverse multimedia content.

VI. CONTRIBUTION TABLE

See each team member's contributions in table VIII.

TABLE VIII
CONTRIBUTION TABLE

Name	Contributions
Harrison	Feature Engineering to extract 'emotion' from transcripts, several SVR models Implementation and Optimization
Michael	Text Summarization Code with corresponding report sections, Conclusion
Christian	Association Rule Mining Code with corresponding report sections
Christine	EDA, Sentiment Analysis, Regression Code with corresponding report sections
Alexandra	Data Processing Code with corresponding report sections, GitHub Page, SVR implementation, Topics and Emotions Classifier
All	Manually scoring sentiments of videos, recording presentation portion

REFERENCES

[1] [Online]. Available: https://www.ted.com/about/our-organization

[2] G. S. di Carlo, "The role of proximity in online popularizations: The case of ted talks," *Discourse Studies*, vol. 16, no. 5, pp. 591–606, 2014. [Online]. Available: https://doi.org/10.1177/1461445614538565

[3] K. MacKrill, C. Silvester, J. W. Pennebaker, and K. J. Petrie, "What makes an idea worth spreading? language markers of popularity in ted talks by academics and other speakers," *Journal of the Association for Information Science and Technology*, vol. 72, no. 8, pp. 1028–1038, 2021. [Online]. Available: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.24471

[4] C. R. Sugimoto, M. Thelwall, V. Larivière, A. Tsou, P. Mongeon, and B. Macaluso, "Scientists popularizing science: characteristics and impact of ted talk presenters," *PloS one*, vol. 8, no. 4, 2013. [Online]. Available: https://doi.org/10.1371/journal.pone.0062403

[5] T. Trzciński and P. Rokita, "Predicting popularity of online videos using support vector regression," *IEEE Transactions on Multimedia*, vol. 19, no. 11, pp. 2561–2570, 2017.

[6] N. Rathee, N. Joshi, and J. Kaur, "Sentiment analysis using machine learning techniques on python," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, pp. 779–785. [Online]. Available: https://doi.org/10.1109/ICCONS.2018.8663224

[7] C. M. Rahman, F. A. Sohel, P. Naushad, and S. M. Kamruzzaman, "Text classification using the concept of association rule of data mining," *CoRR*, vol. abs/1009.4582, 2010. [Online]. Available: http://arxiv.org/abs/1009.4582

[8] R. Obiedat, "Predicting the popularity of online news using classification methods with feature filtering techniques," *Journal of Theoretical and Applied Information Technology*, vol. 98, p. 8, 04 2020. [Online]. Available: https://www.researchgate.net/publication/359214966_PREDICTING_THE_POPULARITY_OF_ONLINE_NEWS_USING_CLASSIFICATION_METHODS_WITH_FEATURE_FILTERING_TECHNIQUES

[9] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *CoRR*, vol. abs/1910.13461, 2019. [Online]. Available: http://arxiv.org/abs/1910.13461

[11] J. Phang, Y. Zhao, and P. J. Liu, "Investigating efficiently extending transformers for long input summarization," *ArXiv*, vol. abs/2208.04347, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251442728

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[13] V. Bonta, N. Kumaresh, and J. Naulegari, "A comprehensive study on lexicon based approaches for sentiment analysis," *Asian Journal of Computer Science and Technology*, vol. 8, pp. 1–6, 03 2019.

[14] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 1906–1919. [Online]. Available: https://aclanthology.org/2020.acl-main.173

[15] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking : Bringing order to the web," in *The Web Conference*, 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID:1508503

[16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Association for Computational Linguistics, 2002, p. 311–318. [Online]. Available: https://doi.org/10.3115/1073083.1073135

[17] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

APPENDIX

TABLE IX
EXAMPLE OF TEXT SUMMARIZATION

Ground Truth: Sir Ken Robinson makes an entertaining and profoundly moving case for creating an education system that nurtures (rather than undermines) creativity.
BART → BART: "I have a big interest in education, and I think we all do" There isn't an education system on the planet that teaches dance every day to children. Art and music are normally given a higher status in schools than drama and dance.
SUMY → BART: One is the extraordinary evidence of human creativity in all of the presentations that we've had. If you're not prepared to be wrong, you'll never come up with anything original. You'd think it would be otherwise, but it isn't.
ChatGPT: The speaker highlighted the significance of creativity in education, asserting its equal importance to literacy. He critiqued the conventional hierarchy of subjects, urging a reevaluation of how we nurture diverse forms of intelligence and creativity in children to prepare them for an uncertain future.