YouTube Content-Based Stock Predictor

Team 2:

Dapo Adegbile
Sutianyi Wen
Jiaman Betty Wu
Yiwen Wang
Christopher Oblak

Abstract:

In this project, we built predictive models using YouTube to predict "buy", "hold", and "sell" on Nvidia stock. We explored many avenues for feature building and concluded on the use of basic video metrics, sentiment analysis on captions and video titles. Using this information, we achieved a validation accuracy of 80.4% for buy, a hold accuracy of 84.6%, and a sell accuracy of 62.9%. In the past, research was conducted using text based social media to predict the stock price but we believed YouTube could perform better as videos have more information than tweets. As a result, we found YouTube has promising predictive power over stock prices.

Introduction:

As techniques for predictive modeling improve, the dream of financial freedom through predicting stock fluctuations has grown. One of these avenues for gathering and synthesizing data on certain markets has been through the use of news and Twitter scraping. However, YouTube might contain more resourceful information from experts in fields that relate directly to products and company performance.

The difference between YouTube and other textual data such as Twitter is the scope associated with those platforms. A five-minute video is able to convey, in a more descriptive nature than say a tweet, the sentiment for a product/company. There are many "YouTubers" that are able to provide candid feedback that are being digested by millions of followers each day. That amount of influence is substantial, and if harnessed, can be a powerful tool to predict stocks[1].

Nvidia has been one of the top stocks over the last two years. In the last year, the stock price ranged from $257 to $614, and currently sits at $575. Being able to confidently predict even one sell or buy a month could have huge portfolio impacts. For these reasons we sought out Nvidia as the stock to investigate and YouTube as the platform for scrapping content meaning.

YouTube content and stock prediction can correlate through the huge influential power of some influencers on the way products and services are received by the population[2,3]. To this end, we were interested in YouTube's top searched content. We hoped to show that the top videos on a given day had potential predictive properties associated. To do this, we looked at standard metrics and caption data. The influential power of YouTubers, along with caption mining, we hoped to find predictive power to anticipate market shifts, ultimately highlighting the influential power of YouTube content.

Background:

Stock price prediction is challenging due to the volatility and non-linearity of stock markets. A variety of machine learning models have been applied to predict stock price changes with efficient performances [4,5,6,7]. For example, Nikou et al. utilized four machine learning models to predict the daily closing price of an exchange-traded fund that represents nearly 85% of the market capitalization of the UK[8]. The best prediction in this study was achieved by Long Short-Term Memory with an MSE of 0.094. Likewise, machine learning methods were applied to India stock market prediction[10] with inputs from India's National Stock Exchange and New York Stock Exchange.

In addition to using financial indices, stock prices are affected by factors such as human behaviors, corporate branding, and product performances [11,12,13]. A study [14] used Twitter text as public sentiment indicators and generated a Fuzzy Neural Network model for stock price prediction. This model incorporated the effects of emotions on individual decision making. In this paper, the researchers measured 6 moods by sentiment scores and predicted the changes of Dow Jones Industrial Average closing values. The results showed a prediction accuracy as 87.6%.

Generally, research on stock prediction by machine learning has been focused on predicting the market, instead of a single stock. In our study, we aimed at modeling the impacts of non-financial features on one specific stock. We obtained features from YouTube, which is an important information source for individual investors. Similar to Twitter, YouTube videos reflect users' emotions, for example, by the number of likes and dislikes. YouTube videos also imply information about the corporate branding and product performances from titles and captions. For this reason, we do not think our data would perfectly fit one technique, or approach, rather we sought to utilize general machine learning models in an effort to explore the data predictive potential.

Data:

A. Data Description:

Building out the dataset occurred in two phases. By scrapping YouTube daily for the top 25 videos related to Nvidia, we obtained raw data on videos for that snapshot in time-related to their basic video metrics (view count, likes, dislikes, publication date, ect.), as well as all caption data by using Amazon Web Service (AWS). From this data, we were able to build and aggregate features over every hour in which data was collected.

The second step is to create the response variable based on 96-hour later percentage change of Nvidia stock price. Because we were interested in predicting stable stock price shifts instead of the stock volatility, we found that using future 96-hour price change yields more stable and significant

price change compared to using shorter future periods such as future 24-hour, 48-hour and 72-hour price change. The rules of generating the response variable is:

i) future 96-hour stock price percentage change is < -3%, the response variable is "sell";
ii) future 96-hour stock price percentage change is between +/- 3% , the response variable is "hold";
iii) future 96-hour stock price percentage change is greater than +3%, the response variable is "buy".

The 3% change threshold was chosen because it is a safe trading strategy for short-term traders in general. For example, assuming Nvidia stock price at 2021-04-12-14:30:00 is $160 and the stock price rises to $172 at 2021-04-16-14:30:00. In this case, the stock price increases 7.5% in 96-hour so we will set action at 2021-04-12-14:30:00 as "buy". In the end, we constructed a dataset with 798 rows and each row contains the aggregated information for a time stamp. The starting time of our data set is from 2020-10-02-16:30:00.

B. Data Visualization and EDA Implication:

For exploratory data analysis, we first examined the response variable distribution. According to the countplot (Figure 1), the dataset has label distribution where 56% are "hold", 23% are "buy" and 21% are "sell". Then, we examined the correlations between variables in order to identify what features should be excluded in our models in order to address the multicollinearity problem before fitting a regression model. From the heatmap (Figure 2), we excluded highly correlated variables pairs from our model. Lastly, we examined the class distribution in two dimensional features by running PCA. There appears to have some clusterings but it is difficult to separate the three classes.(Figure 3)

C. Challenges and Drawbacks of Data:

First, we were unable to scrape from YouTube in a consistent and fixed time interval because the AWS data pipeline was not triggered properly every hour. As a result, our dataset size became small and the time stamp for each observation was not continuous so we treated each row as an independent instance instead of a time series object. Besides, we faced challenges when creating the dependent variable for the after-hour trading period. Yahoo Finance only provides hourly data and we imputed stock price for after-hour, weekend and national holidays. Therefore, our models suffered from a small size and discontinuous time stamp.

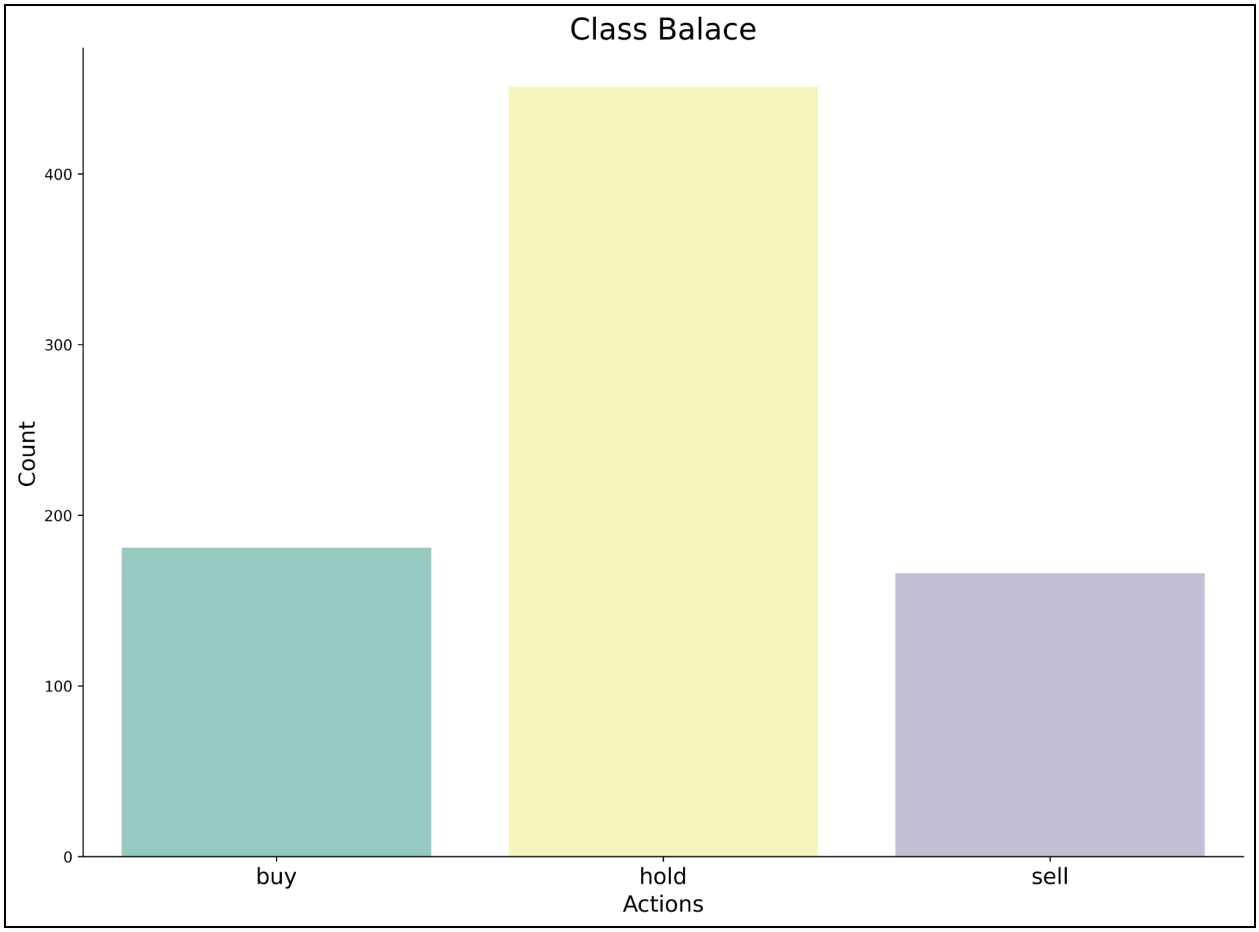

Figure 1. Response Variable Distribution

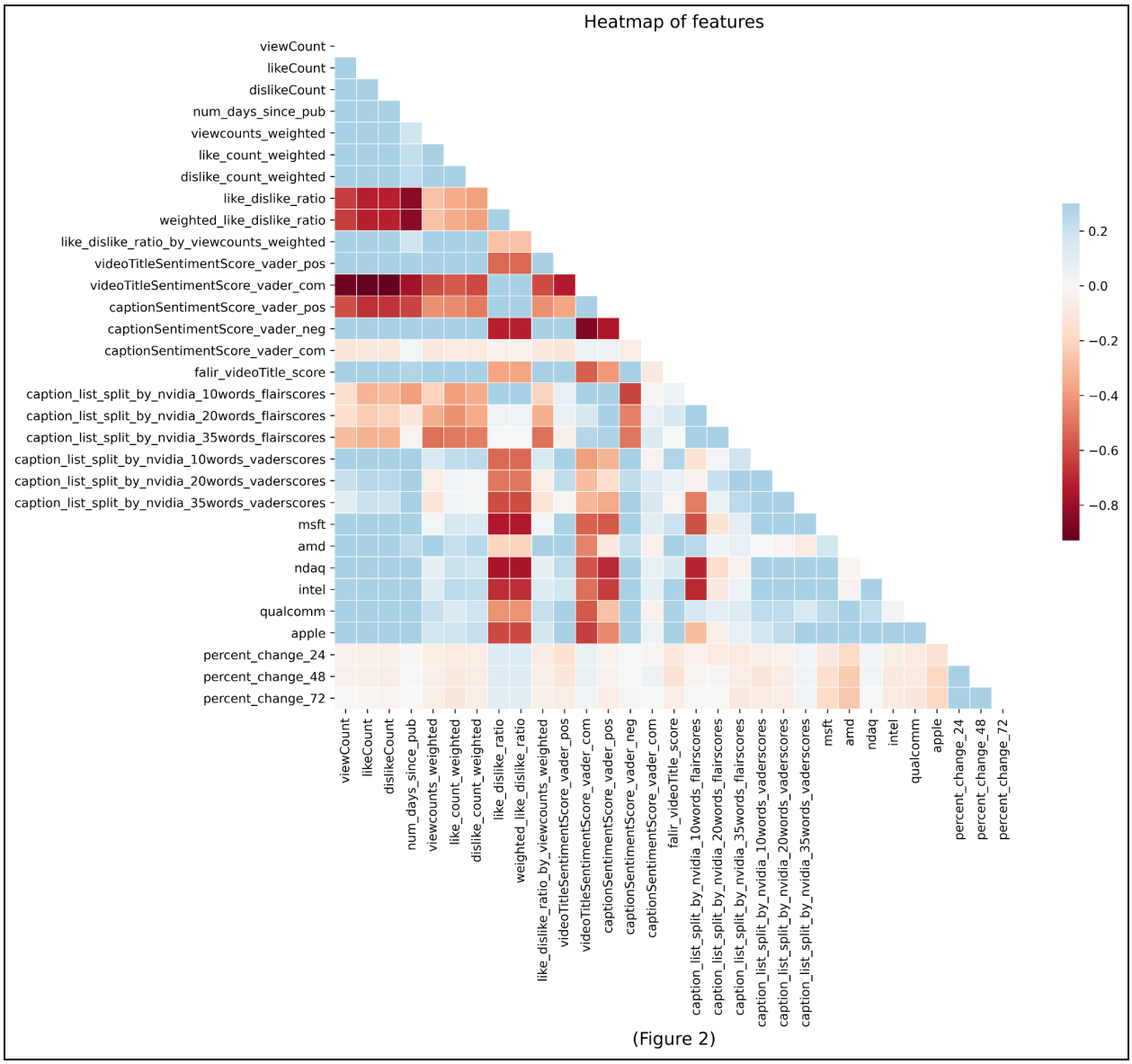

Figure 2. Heatmap of Correlation Between Features Pairs

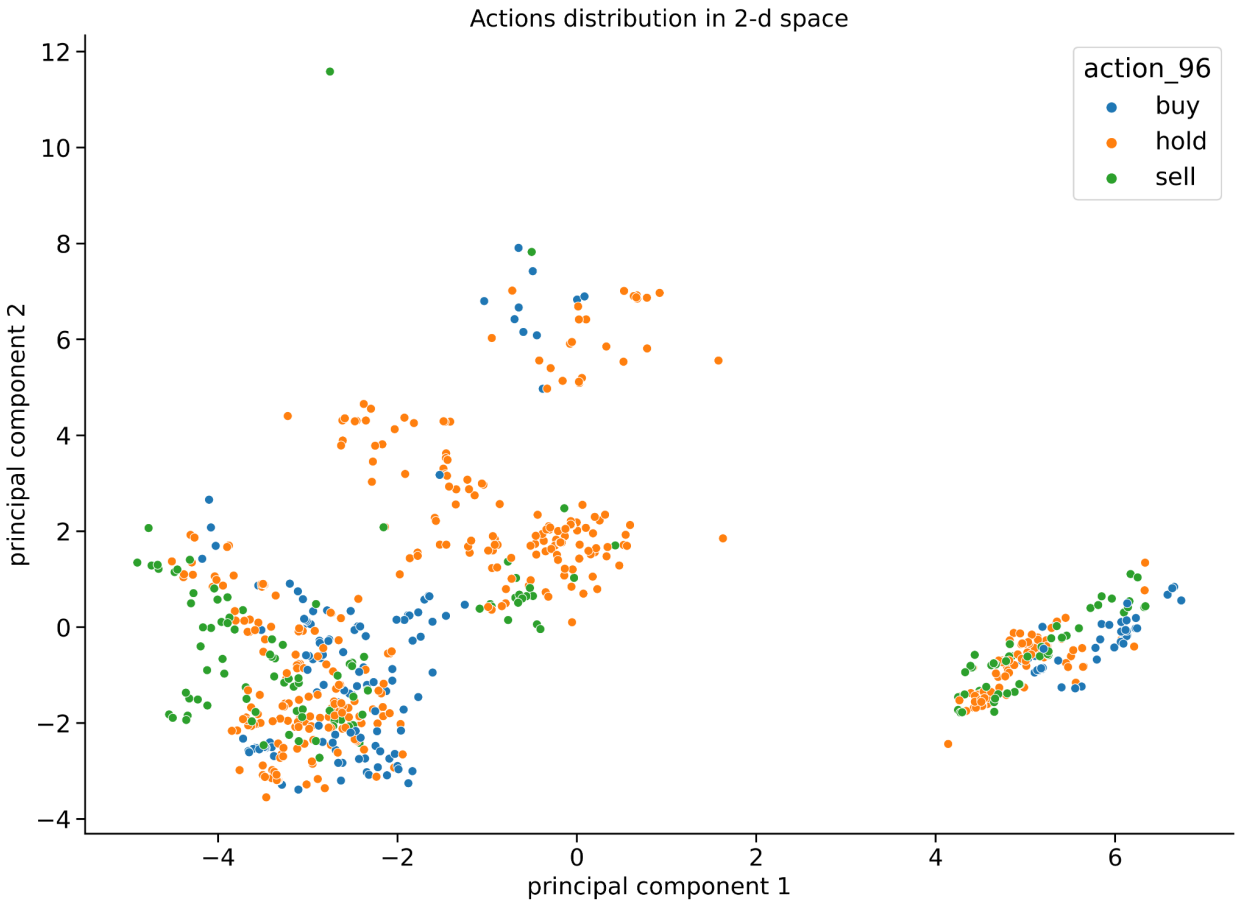

Figure 3. PCA Results

Methods:

A. Feature Engineering

a. Sentiment Scores

Finding a way to quantify the sentiment of video titles and caption text was the key for the project. We were able to rely on two popular pre-training sentiment analysis tools, NLTK's Vader [15] and the Flair's Pytorch trained model [16]. Vader measures the polarity of the words from a dictionary of positive and negative words within a string [17]. Flair uses the state-of-the-art word embedding techniques to produce a sentiment score [18]. We found that Vader and Flair returned different scores, leaving us to include both sentiment analysis in the model to see if they both potentially provided important information in different manners.

We also had to address the noise associated with large stings that were not formatted in an appropriate sentence structure. A long winded speech captured from auto closed caption

generators posed a problem when trying to capture sentiment. To do this, we selected words surrounding our key youtube search word "nvidia" and created arbitrary sentences that could more effectively be categorized as negative or positive. We chose differing lengths of sentences to try and find an optimal amount of word surround "nvidia" to capture an accurate sentiment score.

b. Weighting and Grouping

For other numeric features such as the number of views, likes, and dislikes, we would like to adjust for the fact that older videos naturally get more views, likes, and dislikes. To adjust for that, we divided the raw counts by the number of days a given video has been published. The resulting features are the number of views, likes, and dislikes per day since the videos were published.

After all features were generated, we grouped the videos by hours based on the time of the scrap, and then aggregated the data by taking averages. There are two reasons that we decided to aggregate by hours. First, YahooFinance, which is the API that we used to generate the labels, only returns stock prices by hours. Second, we could maintain the data size by assigning the same labels for all videos in the same group. However, having the same labels for many observations could potentially negatively affect the model performance. Therefore, aggregating the data by hours is reasonable in this case. The final features were then scaled to have mean zero and standard deviation one.

B. Models

Predicting buy, hold, sell actions based on stock prices is a classification problem. There are two main categories of machine learning classifiers that can be applied to solve this problem. The first is parametric models, such as logistic regression. Logistic regression fits a linear decision boundary to separate different classes. It has the highest predictive power when the relationship between features and the label is linear. The other category is the non-parametric approach. There are a myriad of variations of non-parametric machine learning models, such as bagging, random forest, and gradient boosting. In addition, deep learning based algorithms also have high predictive powers for classification problems.

Among many models we explored, we focused on logistic regression and LightGBM[19] for this project. As mentioned previously, logistic regression separates classes based on linear decision boundaries. This model tends to have high bias and low variance. Because of these properties, it is less likely to overfit to the training data. From exploratory data analysis, we

found that the relationship between stock prices and sentiments is complex and non-linear. However, logistic regression is still a reasonable choice to be the baseline model. Due to the more complex nature of stock prices, we explored tree-based ensemble models such as LightGBM. LightGMB is a variation of the tree-based gradient boosting algorithm that is computationally optimized.

Both logistic regression and LightGBM models are trained with the training set, and perform hyperparameter tuning using the grid search on a validation set.

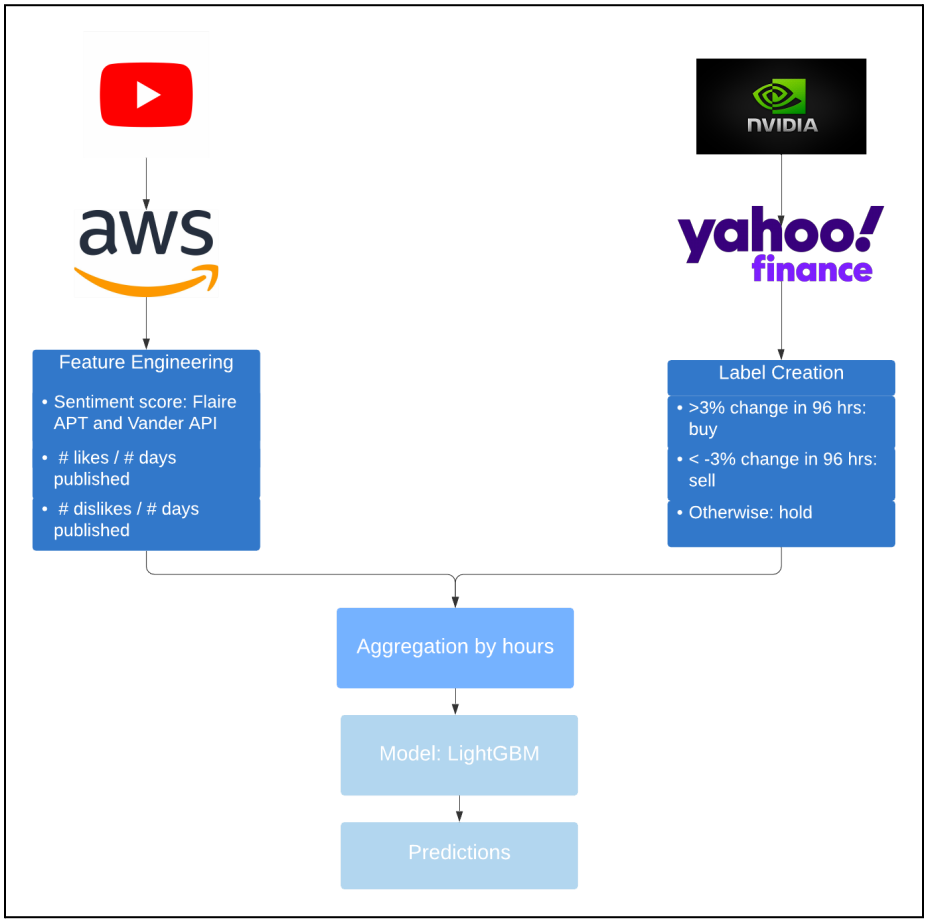

Figure 4. Data and Model Flow Chart

Results:

Of the models we ran on our data, we found the Light Gradient Boosting Machine to be the most effective, giving us a buy accuracy of 80.4%, a hold accuracy of 84.6%, and a sell accuracy of 62.9% on our validation data. Conversely, our baseline model (Logistic Regression) generated accuracies of 33.9% for buy, 83.85% for hold, and 44.4% for sell (Figure 8). Other classification methods we made use of were Random Forest Classification, Naive Bayes, and Stochastic Gradient Descent, but none of these models performed as well as our LightGBM model. We then applied the LightGBM model to a test dataset, generated by scraping Nvidia stock data from April 1st to April 11th. In applying our model to the test data, we were able to generate a buy accuracy of 60%, a hold accuracy of 68%, and a sell accuracy of N/A. There were no true or predicted values of the sell indicator in our test data because there was no 3% decrease in any given hour, giving us a non existent sell accuracy. With a more complete test dataset we should be able to generate more reliable accuracy metrics for when to sell the stock. Our model's PR Curve (Figure 5) , ROC Curve (Figure 6), and confusion matrix (Figure 7) can be found below:

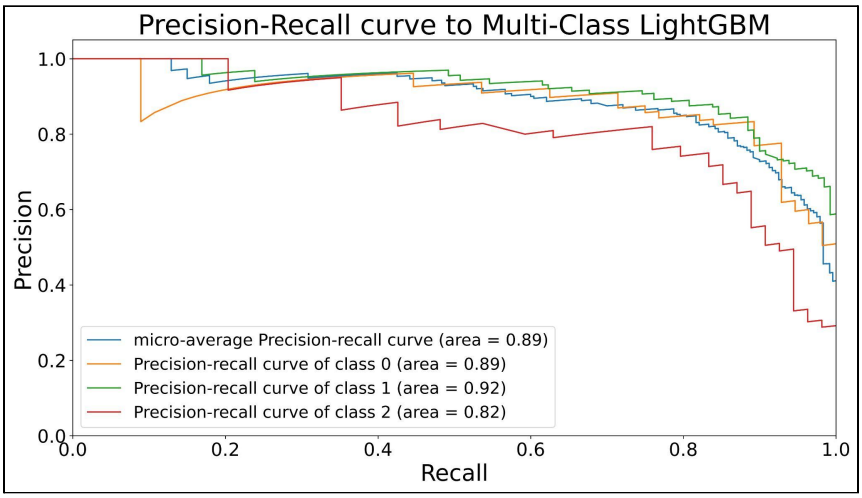

Figure 5. PR Curve of LightGBM Results

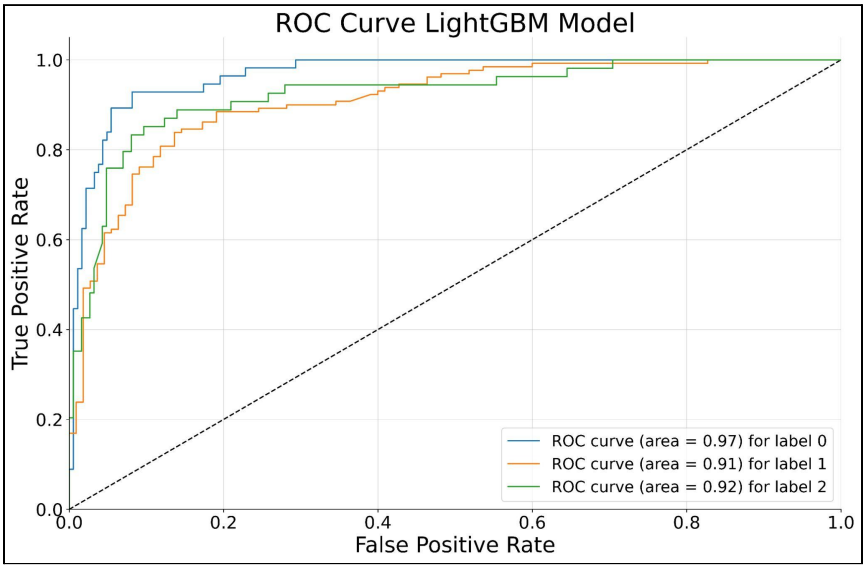

Figure 6. ROC Curve of LightGBM Results

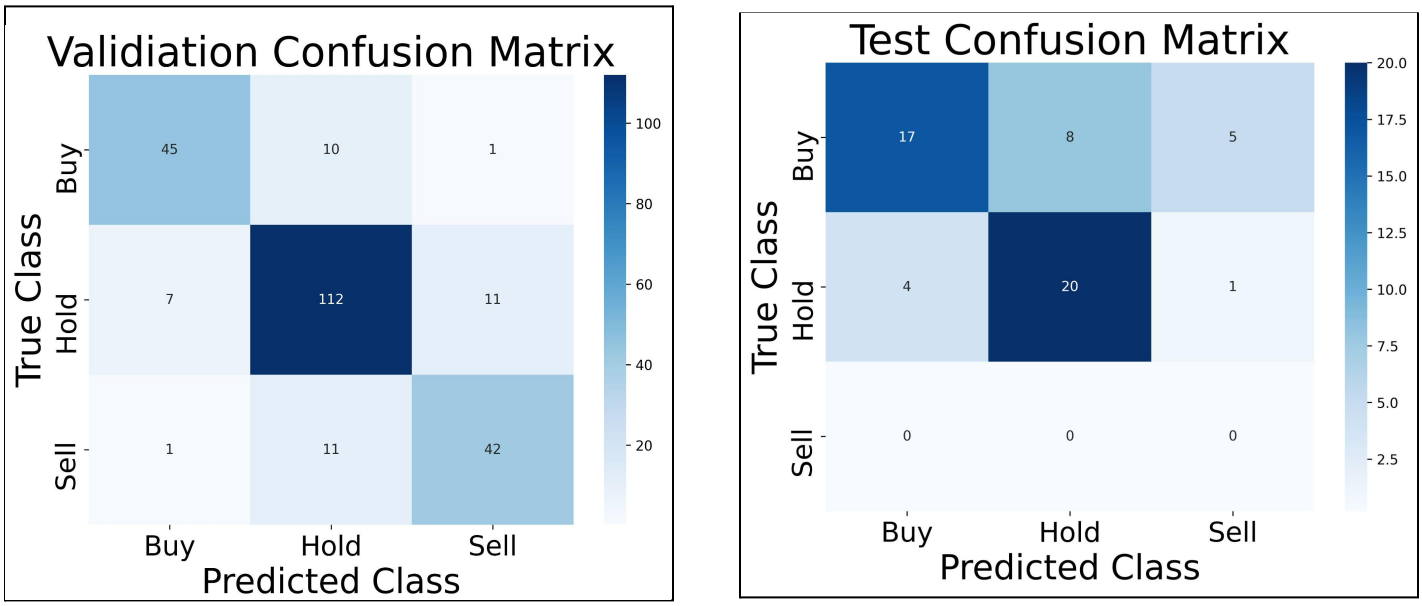

Figure 7. Confusion Matrix on Validation and Test Data

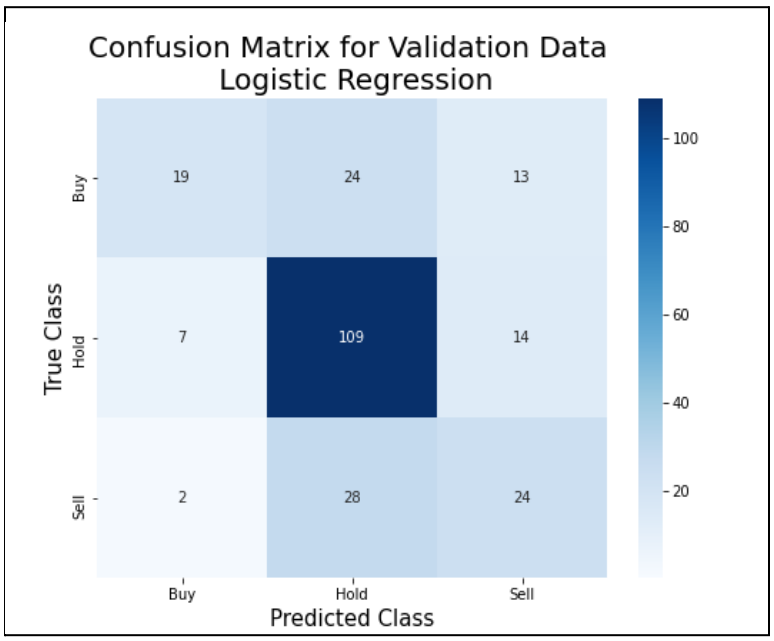

Figure 8. Confusion Matrix on Validation Data of Logistic Regression Model

Conclusion:

To understand how individual investors can leverage information from YouTube to make investment decisions, we used YouTube videos about Nvidia to predict the investment action for Nvidia's stock. We created our own data pipeline to scrape Nvidia related YouTube videos' information such as captions, view counts, and like counts as our features to predict investment action.

Our analysis highlighted that information from YouTube videos were useful to facilitate investment decisions in the stock market because our models showed promising results in confusion matrices and accuracy scores. From the feature importance plots, the sentiments of video titles and captions are the top three important features in our model. However, our model was only trained on a small dataset and did not account for the time series property of stock price so the model may not perform well for unseen data. For instance, Nvidia's stock price rocketed from $552 on April 1st to $623 USD on April 14 but our model still predicted "sell" actions.

Despite the limitations discussed previously, our project has one important takeaway: YouTube videos contain useful information such as video captions and titles to predict the trend of stock price. Our model suggested there exists a correlation between video sentiment and stock prices. Future work includes obtaining information from other social platforms such as Twitter to increase predictive power on stock prices. As a result, individuals can leverage massive information on social media to make investment decisions.

Roles:

Christopher Oblak

- Initiated and oversaw the gathering of YouTube search data from Oct 2020 to Present Day
- Built key metric features from the YouTube individual videos
- Coded captions to rough sentence structure for sentiment analysis
- Coded and built sentiment analysis using both NLTK Vader and Flair pretrained models
- Managed and finalized the github repo
- Contributed to the report by writing the Abstract and Introduction

Dapo Adegbile

- Ran different classification models and tuned hyperparameters to evaluate our final model's prediction capability relative to other classification methods.
- Performed general EDA to better understand initial features

Yiwen Wang

- Performed literature review to understand background and try out models based on former research work.
- Built key metric features from the YouTube individual videos

Betty Wu (Jiaman)

- Generated sentence embeddings for all unique videos, created addition features based the embedding, cleaned the resulting dataframe
- Ran regressions based on the final cleaned data frame with and without the embeddings

Sutianyi Wen

- Used Yahoo Finance API to pull stock price data; imputed missing data and generated response variables
- Merged features and responses into dataframe, and performed EDA on final dataset

Timeline of activity:

Date	Task/Progress
21 FEB 2021	Proposal Completed
23 FEB 2021	Start GitHub Repo Data Cleaning
10 MAR 2021	Feature Creation / Expanding the Model
16 MAR 2021	Model building
23 Mar 2021	Progress Report Completed
30 MAR 2021	Feature Restructuring Model Refinement
06 APR 2021	EDA Analysis Completed
10 APR 2021	Compare Models Start Final Report
13 APR 2012	Final Report 1st Draft
19 APR 2021	Video Completed
21 2100 April 2021	Class Presentation Q&A
23 2100 April 2021	Final Report 2nd Draft

References:

1. Otani, A. (2021, March 21). *The New Stock Influencers Have Huge—and Devoted—Followings*. The Wall Street Journal. https://www.wsj.com/articles/the-new-stock-influencers-have-hugeand-devotedfollowings-11616319001
2. McKenna, B. (2020). *3 Cool New Products That Should Help Power NVIDIA Stock Higher*. https://www.fool.com/investing/2020/11/23/3-cool-new-products-that-should-help-power-nvidia/
3. Narayanan, A. (2021). *Is Nvidia Stock A Buy As New Chip To Take On Intel, AMD?* Investor's Business Daily. https://www.investors.com/research/nvidia-stock-buy-now/
4. Nabipour, M., Nayyeri, P., Jabani, H., & Mosavi, A. (2020). Deep learning for Stock Market Prediction. 10.3390/e22080840
5. Mehtab, S., Sen, J., & Dutta, A. (2020). Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models. *arXiv:2009.10819*.
6. Hu, Z., Zhao, Y., & Khushi, M. (2021). A Survey of Forex and Stock Price Prediction Using Deep Learning. *Appl. Syst. Innov.*, *4*(9). https://doi.org/10.3390/asi4010009
7. Kalyvas, E. (2001). Using neural networks and genetic algorithms to predict stock market returns. *MSc thesis*. http://dl.fxf1.com/files/books/english/Using%20Neural%20Networks%20and%20Genetic%20Algorithms%20to%20Predict%20Stock%20Market%20Returns.pdf

8. BlackStone. (2021). *iShares MSCI United Kingdom*. https://www.ishares.com/us/literature/fact-sheet/ewu-ishares-msci-united-kingdom-etf-fund-fact-sheet-en-us.pdf
9. Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science*, (132), 1351–1362.
10. BankBazaar. (2011). *Non-financial factors and stock prices – where to invest?* https://blog.bankbazaar.com/non-financial-factors-affecting-stock-prices-where-should-you-invest/
11. Rautiainen, S., & Marcial, J. (2017). *The Effects of Social Behaviour on the Stock Market*. https://core.ac.uk/download/pdf/161417762.pdf
12. Huang, J. (2018). The Customer Knows Best: The Investment Value of Consumer Opinions. *Journal of Financial Economics*, (128), 164–182. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2758807
13. Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. *Intell Sys Acc Fin Mgmt.*, (26), 164-174. https://doi.org/10.1002/isaf.1459
14. Bollen, J., & Mao, H. (2010). Twitter mood as a stock market predictor. *IEEE Computer*, *44*(10), 91-94.
15. Hutto, C. J., & Gilbert, E. E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*.

16. Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual String Embeddings for Sequence Labeling. *27th International Conference on Computational Linguistics*, 1638--1649. https://www.aclweb.org/anthology/C18-1139/

17. Beri, A. (2020). *SENTIMENTAL ANALYSIS USING VADER*. towards data science. https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

18. Magajna, T. (2018). *Text Classification with State of the Art NLP Library — Flair*. towards data science. https://towardsdatascience.com/text-classification-with-state-of-the-art-nlp-library-flair-b541d7add21f

19. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: a highly efficient gradient boosting decision tree. *In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, 3149–3157.

20. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. https://arxiv.org/abs/1908.10084

Appendix:

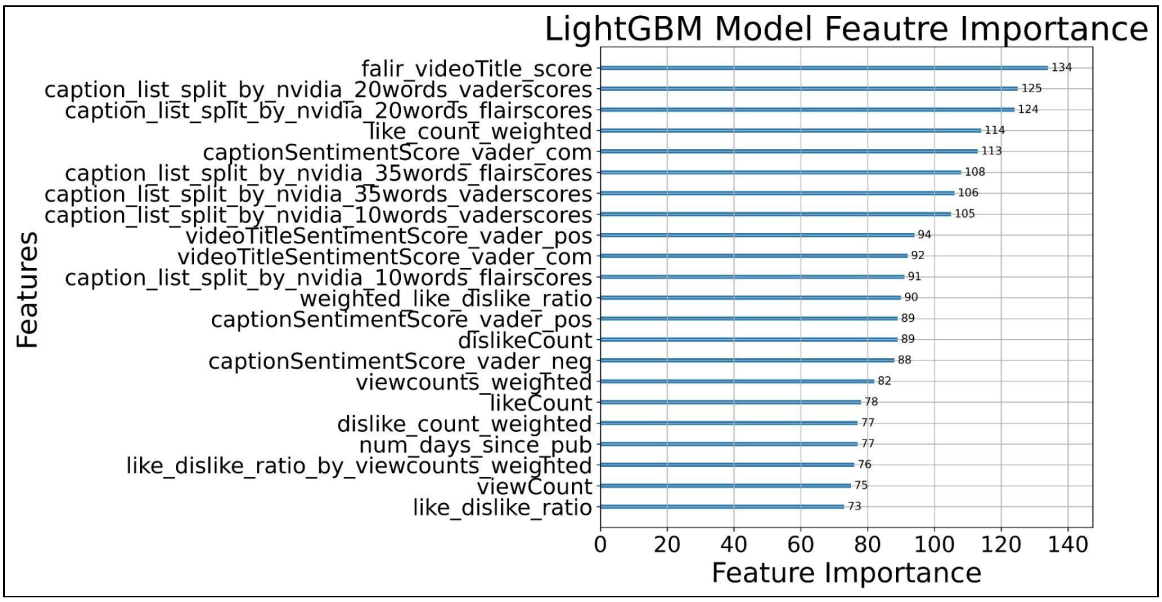

Figure 9. Plot of Variable Importance to the Model

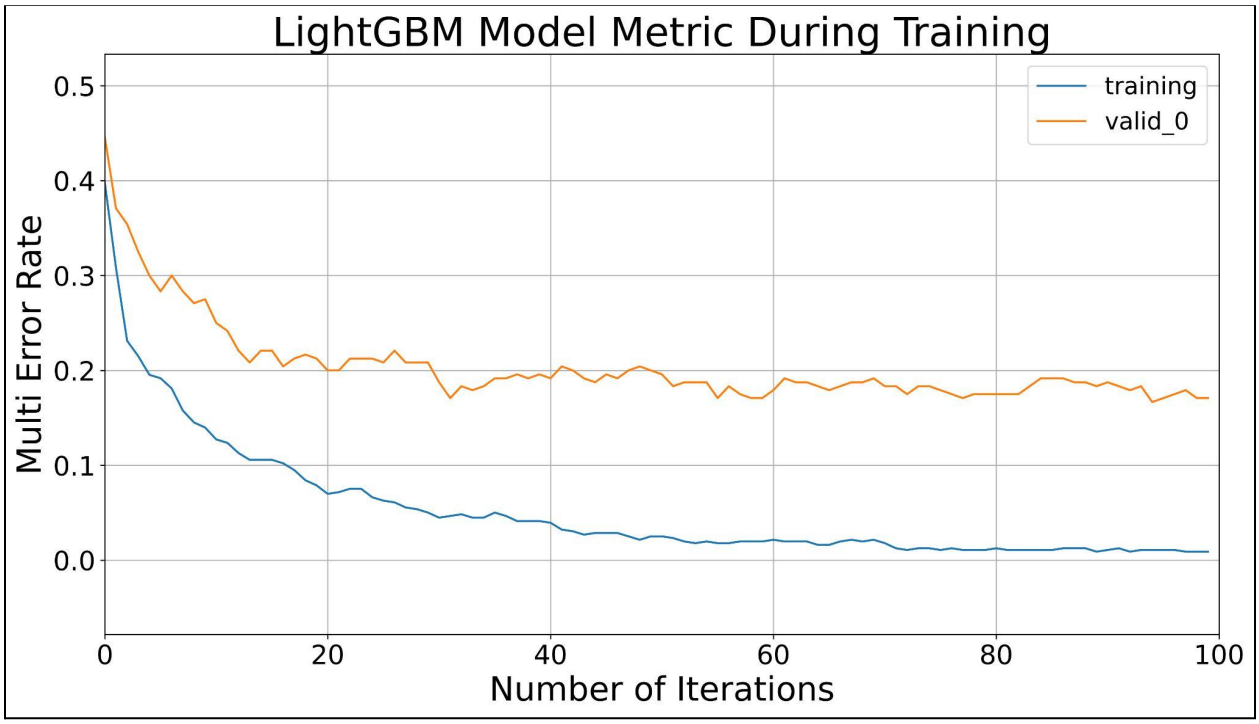

Figure 10. Plot of Error over Number of Iterations